U.PORTO

PROGRAMA DOUTORAL

BIOLOGIA BÁSICA E APLICADA

A workflow for developing biochemical pathway models using ordinary differential equations

João Pedro Gomes dos Santos

D

2024

U.PORTO

João Pedro Gomes dos Santos.

A workflow for developing biochemical pathway models using ordinary differential equations

D.ICBAS **2024**

A workflow for developing biochemical pathway models using ordinary differential equations

João Pedro Gomes dos Santos

SEDE ADMINISTRATIVA

INSTITUTO DE CIÊNCIAS BIOMÉDICAS ABEL SALAZAR

FACULDADE DE CIÊNCIAS

FACULDADE DE MEDICINA

João Pedro Gomes dos Santos

A workflow for developing biochemical pathway models using ordinary differential equations

Dissertation for applying to a Doctoral degree in Basic and Applied Biology;
Doctoral Program of the University of Porto (Institute of Biomedical Sciences Abel Salazar, Faculty of Sciences of the University of Porto, Faculty of Medicine of the University of Porto)

Supervisor: Professor Jeanette Hellgren Kotaleski
Position: Professor
Affiliation: KTH Royal Institute of technology, Karolinska Institute
Co-Supervisor: Doctor Teresa Summavielle
Position: Principal Investigator
Affiliation: i3S - Institute for Research and Innovation in Health

Acknowledgements

These acknowledgments are written in both Portuguese and English to best reflect the diverse group of people who have supported me. Each section is in the language most familiar to the individuals being thanked, to honor their contributions in a meaningful way.

Estas palavras de agradecimento estão escritas em português e em inglês para refletir melhor o grupo diverso de pessoas que me apoiaram. Cada secção está no idioma mais familiar para as pessoas que estão a ser agradecidas, para homenagear as suas contribuições de uma forma significativa.

Quero expressar a minha mais profunda gratidão à minha família—os meus pais, a minha irmã, os meus padrinhos e os meus primos—pelo apoio incondicional durante os meus momentos mais difíceis. Um agradecimento especial ao meu primo Gonçalo, que foi quase como um co-supervisor, fornecendo feedback técnico inestimável e orientação intelectual ao longo da minha tese. Gustavo, obrigado por estares sempre atento e por garantires que eu estava bem. Também sou muito grato aos meus tios Matias e Isabel pela calorosa hospitalidade, pela comida deliciosa e por toda a ajuda que me deram quando precisei. O apoio deles durante as minhas visitas ao Tasco fez uma grande diferença. À minha prima Sara, obrigado pelo apoio atencioso, especialmente por me ajudar a enfrentar os desafios relacionados com o ADHD e a depressão, e por me ajudar a compreender melhor os meus pensamentos e sentimentos. À Filomena, agradeço também a calorosa hospitalidade e toda a comida deliciosa durante a minha estadia na Póvoa de Varzim. Agradeço também ao resto da família da Filipa—à Sofia, ao André, ao Gabriel e ao António—pelos momentos maravilhosos que passámos juntos, e um agradecimento especial ao António e ao Gabriel por todas as brincadeiras divertidas que tivemos.

Beyond my family, I am immensely grateful to my friends for their companionship and unwavering support. Jessica, Rickard, Luís, João Manuel, Mafalda, John, and Ricardo—our hangouts, conversations, and shared moments helped me maintain balance and made my time in Stockholm/Copenhagen truly unforgettable. Carla, your friendship and our long talks always provided the space I needed to relax and shift my focus away from my thesis and life's challenges. Your support made a real difference.

Quero também agradecer aos meus amigos online do nosso chat do Teamspeak "Vacagalo"—Gustavo (Fanekaz), Gonçalo (Dolando), João Janes (Naja), Miguel Viola (Telheiras), José Soares (Super), João Coelho (Samir), André Corda (Bamba), Rui Poeiras (Murmilos), Rui Pereira (Gaga), João Santos (Rajantes), Pedro Miguel Caldeira (Caldeira), Luís

Filipe (IURDE). Vocês foram uma fonte constante de entretenimento e apoio durante o meu doutoramento. Quer fosse a jogar jogos de computador, a contar piadas ou a discutir, ajudaram-me a descontrair e a manter o ânimo. Um agradecimento especial ao Telheiras por perguntar sempre sobre o progresso da minha tese, motivando-me a avançar e terminar o mais rapidamente possível.

Estou extremamente grato aos meus amigos do ensino secundário—João Miguel, Gonçalo e João Queiroz, e aos meus amigos da universidade—Filipe, Miguel, Tiago Torres, Vera, Tiago Silva e João Pedro. Os bons momentos que partilhámos durante os verões e o Natal ajudaram-me sempre a recarregar energias e deram-me a força necessária para continuar ao longo do ano.

Estou também grato aos meus colegas do GABBA pelo apoio durante as fases iniciais do meu doutoramento. O vosso encorajamento foi fundamental para me manter focado e motivado.

In my academic journey, Jeanette, your guidance and valuable feedback have been truly invaluable. Olivia, thank you for helping me get started with MATLAB, teaching me the fundamentals of sensitivity analysis, and contributing significantly to our co-authored paper. Parul, your camaraderie as we tackled early challenges with MATLAB was greatly appreciated. Daniel, collaborating with you on our paper and having those stimulating scientific discussions was truly enriching. Kadri, your efforts in implementing our MATLAB model and co-authoring the paper were instrumental, and your fresh perspective greatly helped me refine the code. Andrei, your support with the mathematical analysis, coding assistance, and insightful conversations had a profound impact. Johanna, Alex, Johannes, and Robert—thank you for your support and camaraderie, which made my time in the lab more enjoyable and motivating.

Um agradecimento especial à Teresa Summavielle por aceitar ser minha co-supervisora, por me guiar na burocracia portuguesa e por me permitir trabalhar no seu laboratório sempre que necessário.

Finalmente, a minha mais profunda gratidão vai para a minha namorada Filipa. O seu amor, paciência e apoio emocional ao longo desta jornada foram absolutamente essenciais. Este trabalho não teria sido possível sem ela, e agradeço profundamente a sua ajuda na revisão de texto e as suas valiosas sugestões.

João Santos was supported by a PhD fellowship (PD/BD/114180/2016) awarded by Fundação para a Ciência e Tecnologia (FCT).

Contributions

This section lists the contributions made by each author in the publication Santos *et al.* [1].

João P. G. Santos created the SBtab files for the models by Nair *et al.* (2016) [2, 3], Viswan *et al.* (2018) [4, 5], and Fujita *et al.* (2010) [6, 7]. These SBtab files were then converted in the different softwares used in this project. He also developed the following tools in MATLAB and wrote the corresponding sections of the article: Diagnostics tool, Parameter estimation, global sensitivity analysis (GSA), and SBtab to SBML + m + tsv conversion tool. He also created the MATLAB images and edited all the images from the other software.

Anu G. Nair is the author of the original model on which this paper is based. He also collaborated with Kadri Pajo to obtain the data from the original model and prepare it for the new study.

Kadri Pajo collaborated in writing the article in addition to the above.

Daniel Trpevski developed the work on NEURON and wrote the corresponding sections of the article .

Andrey Stepaniuk and Daniel Keller developed the work on STEPS and BioNetGen, including the Simulation Setup Webapp, and wrote the corresponding sections of the article.

Olivia Eriksson collaborated with João Santos on the GSA and assisted in writing the corresponding section of the article.

Jeanette Hellgren Kotaleski and Andrei Kramer coordinated the project and and contributed to writing sections of the article. Additionally, Andrei Kramer also developed the get_thermodynamic_constraints.md, and sbtab_to_vfgen.Re tools.

Joao, Kadri, Andrei and Jeanette reviewed the final article and prepared it for submission.

Resumo

Esta dissertação apresenta um software inovador e integrado para a análise de modelos de vias bioquímicas utilizando equações diferenciais ordinárias (EDOs), combinando de forma única ferramentas para inicialização de modelos, simulação, estimativa de parâmetros e análise numa única plataforma. Seguindo os princípios FAIR (Findable – encontrável, Accessible – acessível, Interoperable – interoperável, Reusable – reutilizável), o quadro facilita o desenvolvimento e análise de modelos que vão desde interações moleculares simples até cascatas de sinalização neurobiológica complexas, com foco particular na sinalização dos gânglios basais.

As capacidades do software são demonstradas através da validação sistemática usando quatro modelos de complexidade crescente. A validação inicial emprega dois modelos personalizados: LIRMO, que simula interações básicas ligando-recetor, e LIREM, que estende estas interações para incluir resposta de um efetor. O software, então, demonstra a sua capacidade de melhorar modelos existentes através do TM-M, onde mecanismos complexos de auto-fosforilação são traduzidos com sucesso em EDOs mantendo a precisão biológica. Finalmente, o modelo G-PIPK destaca a capacidade do software para lidar com cascatas de sinalização sofisticadas, integrando múltiplas vias de neurotransmissores e revelando novas perceções sobre dinâmicas de sinalização específicas de compartimentos entre sinais glutamatérgicos e colinérgicos convergentes em neurónios do estriado.

O software demonstra um desempenho robusto em modelos de diferentes complexidades, lidando com sistemas de 7 a 35 parâmetros livres. Através de análises de sensibilidade abrangentes, identifica parâmetros-chave que influenciam o comportamento do modelo, enquanto análises de perfil de verosimilhança revelam limitações de identificabilidade de parâmetros. As capacidades preditivas do software são validadas por simulações precisas de dinâmicas de sinalização. Especificamente, o modelo G-PIPK demonstra o impacto do volume de cada compartimento nos processos de sinalização, nomeadamente nas espinhas dendríticas em comparação com células inteiras.

Este trabalho avança a neurociência computacional ao proporcionar aos investigadores uma ferramenta versátil para o desenvolvimento e análise de modelos bioquímicos. O software aproxima dados experimentais e modelação computacional, permitindo uma análise rigorosa de vias de sinalização celular. A sua aplicação a mecanismos de sinalização neuronal fornece uma base para investigações futuras sobre plasticidade sináptica e neuromodulação, com potenciais implicações para a compreensão tanto da função cerebral normal

como de condições patológicas.

Abstract

This dissertation introduces an innovative, integrated software for analysing biochemical pathway models using ordinary differential equations (ODEs), uniquely combining tools for model initialization, simulation, parameter estimation, and analysis in a single platform. Following FAIR (Findable, Accessible, Interoperable, Reusable) principles, the framework facilitates the development and analysis of models ranging from simple molecular interactions to complex neurobiological signalling cascades, with particular focus on basal ganglia signalling.

The framework's capabilities are demonstrated through systematic validation using four models of increasing complexity. Initial validation employs two custom-designed models: LIRMO, simulating basic ligand-receptor interactions, and LIREM, extending these interactions to include effector responses. The framework then demonstrates its ability to enhance existing models through TM-M, where complex auto-phosphorylation mechanisms are successfully translated into ODEs while maintaining biological accuracy. Finally, the G-PIPK model showcases the framework's capacity to handle sophisticated signalling cascades, integrating multiple neurotransmitter pathways and revealing novel insights into compartment-specific signalling dynamics between converging glutamatergic and cholinergic signals in striatal neurons.

The software demonstrates robust performance across varying model complexities, handling systems from 7 to 35 free parameters. Through comprehensive sensitivity analyses, it identifies key parameters influencing model behaviour, while profile likelihood analyses reveal parameter identifiability constraints. The framework's predictive capabilities are validated through accurate simulation of signalling dynamics. Specifically, the G-PIPK model demonstrates how compartment volume impacts signalling processes, particularly in dendritic spines versus whole cells.

This work advances computational neuroscience by providing researchers with a versatile tool-kit for developing and analysing biochemical models. The framework bridges the gap between experimental data and computational modelling, enabling rigorous analysis of cellular signalling pathways. Its successful application to neuronal signalling mechanisms provides a foundation for future investigations into synaptic plasticity and neuromodulation, with potential implications for understanding both normal brain function and pathological conditions.

Contents

Acknowledgements i

Contributions iv

Resumo v

Abstract vii

List of Publications xiv

List of Figures xv

List of Tables xviii

List of algorithms xix

Acronyms xx

1 Introduction 1

1.1 Basal ganglia and related structures 1

1.1.1 Striatum 2

1.1.2 Globus pallidus 3

1.1.3 Subthalamic nucleus 4

1.1.4 Substantia nigra 5

1.1.5 Thalamus 6

1.2 Neurons of the striatum 6

1.2.1 Spiny projection neurons 7

1.2.2 Interneurons 7

1.3 Dendritic synapses of SPNs 10

1.3.1 Neurotransmitters and their receptors 11

1.3.1.1 Glutamate 11

1.3.1.2 GABA 14

1.3.1.3 Dopamine 16

1.3.1.4 Acetylcholine 17

1.3.2 Striatal plasticity in corticostriatal synapses 18

1.3.2.1 Long-term depression . . . 19
1.3.2.2 Long-term potentiation . . . 23
1.3.2.3 Spike timing-dependent plasticity . . . 28
1.3.3 Intracellular pathways important for striatal plasticity . . . 28
1.3.3.1 G_q-coupled receptor cascades . . . 29
1.3.3.2 Endocannabinoid signalling . . . 30
1.3.3.3 cAMP dynamics . . . 31
1.3.3.4 Ca^{2+}-dependent signalling . . . 31
1.3.3.5 Phosphatase-mediated control . . . 32
1.3.3.6 Kinase-mediated regulation . . . 33
1.3.3.7 Other integrator proteins . . . 34
1.3.3.8 Synaptic effectors . . . 35
1.4 Modelling in neuroscience . . . 36
1.4.1 Multicellular and network models . . . 36
1.4.1.1 Population models . . . 37
1.4.1.2 Spiking neural networks . . . 37
1.4.1.3 Connectome-based models . . . 38
1.4.2 Cellular and electrical models . . . 39
1.4.2.1 Single-compartment models . . . 39
1.4.2.2 Multi-compartment models . . . 39
1.4.3 Sub-cellular and biochemical models . . . 40
1.4.3.1 Reaction-diffusion models . . . 40
1.4.3.2 Kinetic models . . . 41
1.4.3.3 Stochastic models . . . 42
1.4.4 Cross-scale integration . . . 42
1.5 FAIR Principles in Computational Neuroscience . . . 43
1.5.1 Findability . . . 44
1.5.2 Accessibility . . . 44
1.5.3 Interoperability . . . 45
1.5.4 Reusability . . . 46
1.5.5 Importance and implementation of FAIR in neuroscience research . . . 46
1.5.6 FAIR principles in the context of this thesis . . . 47
1.5.6.1 Findability . . . 47
1.5.6.2 Accessibility . . . 48

1.5.6.3 Interoperability . . . 49
1.5.6.4 Reusability . . . 49

2 Aims **51**

3 Methods **53**
3.1 Biochemical pathway models . . . 53
3.1.1 Model components . . . 54
3.2 Model Data . . . 55
3.2.1 SBtab . . . 55
3.3 Simulation . . . 56
3.3.1 Reaction kinetics . . . 58
3.3.2 Ordinary differential equation systems . . . 58
3.4 Scoring functions . . . 59
3.5 Parameter estimation . . . 60
3.5.1 Interior point algorithm . . . 61
3.5.2 Simulated annealing . . . 62
3.5.3 Pattern search . . . 64
3.5.4 Genetic algorithm . . . 64
3.5.5 Particle swarm . . . 68
3.5.6 Surrogate optimization . . . 70
3.6 Sensitivity analysis . . . 72
3.6.1 Local sensitivity analysis . . . 72
3.6.1.1 Parameter sampling for local sensitivity analysis . . . 73
3.6.2 Global sensitivity analysis . . . 74
3.6.2.1 First and total order sensitivities calculation . . . 74
3.6.2.2 Parameter sampling for construction of sensitivity matrices . . . 76
3.6.2.3 Bootstrapping . . . 77
3.7 Profile likelihood analysis . . . 77

4 Results and Discussion **81**
4.1 Example models . . . 81
4.1.1 LiRMo . . . 81
4.1.2 LiREM . . . 83
4.1.3 Discussion . . . 84
4.2 The modelling system . . . 86

4.2.1 Model initialization into MATLAB . . . 86
4.2.2 Model simulation . . . 87
4.2.2.1 MATLAB simulation environment . . . 87
4.2.2.2 Model equilibration . . . 90
4.2.2.3 Model inputs and outputs . . . 90
4.2.2.4 Model parameter substitution . . . 91
4.2.3 Model parameter set scoring . . . 93
4.2.4 Discussion . . . 94
4.3 The model analysis software . . . 96
4.3.1 Creation of dummy data . . . 96
4.3.2 Diagnostics . . . 101
4.3.3 MATLAB Optimization Toolbox . . . 101
4.3.3.1 fmincon . . . 102
4.3.3.2 Simulated annealing . . . 103
4.3.3.3 Pattern search . . . 104
4.3.3.4 Genetic algorithm . . . 104
4.3.3.5 Particle swarm . . . 105
4.3.3.6 Surrogate optimization . . . 105
4.3.4 Parameter estimation . . . 105
4.3.5 Sensitivity analysis . . . 107
4.3.5.1 Local sensitivity analysis . . . 108
4.3.5.2 Global sensitivity analysis . . . 109
4.3.6 Profile likelihood analysis . . . 111
4.3.7 Discussion . . . 112
4.4 Timing model . . . 115
4.4.1 Model overview . . . 115
4.4.1.1 Calcium cascade . . . 115
4.4.1.2 Autophosphorylation of CaMKII . . . 116
4.4.1.3 Dopamine cascade . . . 116
4.4.1.4 Hooks to original model . . . 117
4.4.2 Model analysis . . . 117
4.4.3 Parameter estimation . . . 119
4.4.4 Sensitivity analysis . . . 122
4.4.4.1 Local sensitivity analysis . . . 122

4.4.4.2 Global sensitivity analysis . . . 123
4.4.5 Profile likelihood analysis . . . 127
4.4.6 Discussion . . . 129
4.5 G-PIPK model . . . 130
4.5.1 Model overview . . . 131
4.5.1.1 Ligand-GPCR binding . . . 132
4.5.1.2 GPCR-G-protein coupling . . . 134
4.5.1.3 G_α interaction with $G_{\beta\gamma}$ . . . 134
4.5.1.4 G_αGTP to G_αGDP conversion . . . 135
4.5.1.5 G_α interaction with PLC . . . 135
4.5.1.6 Hydrolysis of PIP_2 by PLC . . . 136
4.5.1.7 PI cycle . . . 136
4.5.1.8 PIP_2-dependent Kv7.2/3 membrane current . . . 136
4.5.1.9 Real-time monitoring: PH probes for IP_3 and PIP_2 . . . 137
4.5.2 Experimental data for model calibration and validation . . . 137
4.5.2.1 Ligand-GPCR binding dynamics . . . 139
4.5.2.2 G-protein coupling and interactions . . . 140
4.5.2.3 Downstream signalling and membrane response . . . 142
4.5.3 Model analysis . . . 144
4.5.4 Parameter estimation . . . 146
4.5.5 Sensitivity analysis . . . 148
4.5.5.1 Local sensitivity analysis . . . 148
4.5.5.2 Global sensitivity analysis . . . 150
4.5.6 Profile likelihood analysis . . . 154
4.5.7 Volume considerations . . . 156
4.5.8 Model extrapolations for cell and spines . . . 159
4.5.9 Discussion . . . 161

5 Conclusion **166**

Bibliography **169**

Appendices **205**

A SBtab **205**

B Example models **210**

C TM-M **228**

D G-PIPK **232**

List of Publications

The work described in this dissertation has been the basis for the following publication, code repositories and models:

Research articles:

[1] João PG Santos et al., 2022. "A modular workflow for model building, analysis, and parameter estimation in systems biology and neuroscience". *Neuroinformatics* 20(1), pp. 241–259

Software:

[8] João PG Santos, 2024. "Subcellular workflow". *GitHub repository*, URL: `https://github.com/jpgsantos/Subcellular_workflow`, Release tag: 2.0

[9] João PG Santos et al., 2022. "Subcellular workflow". *GitHub repository*, URL: `https://github.com/jpgsantos/Subcellular_workflow`, Release tag: 1.0

Original models:

[10] João PG Santos and Jeanette Hellgren Kotaleski, 2024. "Model GPIPK". *GitHub repository*, URL: `https://github.com/jpgsantos/GPIPK`, Release tag: 1.0

Modified models:

[11] João PG Santos and Jeanette Hellgren Kotaleski, 2024. "Timing Model Modified". *GitHub repository*, URL: `https://github.com/jpgsantos/Timing_Model_Modified`, Release tag: 1.0

[3] João PG Santos et al., 2022. "Model Nair 2016". *GitHub repository*, URL: `https://github.com/jpgsantos/Model_Nair_2016`, Release tag: 1.0

[5] João PG Santos et al., 2022. "Model Viswan 2018". *GitHub repository*, URL: `https://github.com/jpgsantos/Model_Viswan_2018`, Release tag: 1.0

[7] João PG Santos et al., 2022. "Model Fujita 2010". *GitHub repository*, URL: `https://github.com/jpgsantos/Model_Fujita_2010`, Release tag: 1.0

List of Figures

1.1 Basal ganglia connections and neurotransmitters . . . 2
1.2 Proportions and characteristics of striatal interneurons . . . 8
1.3 Molecular representation of mGluR5 . . . 14
1.4 Corticostriatal LTD . . . 24
1.5 Corticostriatal LTP . . . 26

3.1 Example polling vectors for pattern search . . . 65
3.2 Distributions . . . 76
3.3 Parameter profile likelihood examples . . . 80

4.1 LiRMo and LiREM schematics . . . 82
4.2 LiRMo MATLAB import code . . . 87
4.3 LiRMo species after MATLAB import . . . 88
4.4 LiRMo equilibration demonstration . . . 90
4.5 LiRMo inputs and outputs . . . 92
4.6 LiRMo inputs and outputs with focus on E_3 . . . 92
4.7 LiRMo dummy data . . . 99
4.8 LiRMo E_0 diagnostics plot . . . 102
4.9 LiRMo diagnostic scores plot . . . 103
4.10 LiRMo optimized θ . . . 107
4.11 LiRMo optimized θ comparative scores . . . 107
4.12 LiRMo LSA . . . 108
4.13 LiREM LSA . . . 109
4.14 LiRMo and LiREM GSA S_{Fi} and S_{Ti} for all Es . . . 111
4.15 LiRMo and LiREM PLA for all Es . . . 113
4.16 TM-O and TM-M schematic overview and experimental timing configurations 119
4.17 TM-M inputs $E_{0–9}$ . . . 120
4.18 TM-M diagnostic scores plot . . . 121
4.19 TM-M outputs $E_{0–2}$ . . . 122
4.20 TM-M LSA $E_{0–3}$ (DA then Ca^{2+}) . . . 123
4.21 TM-M LSA $E_{5–8}$ (Ca^{2+} then DA) . . . 124
4.22 TM-M GSA S_{Fi} for each Eo . . . 125

4.23 TM-M GSA S_{Ti} for each Eo 126
4.24 TM-M PLA for each E 128
4.25 G-PIPK network diagram of the GPCR signalling pathway 133
4.26 G-PIPK diagnostic scores plot 145
4.27 G-PIPK first optimized θ 147
4.28 G-PIPK LSA 148
4.29 G-PIPK GSA S_{Fi} for each Eo 151
4.30 G-PIPK GSA S_{Ti} for each Eo 152
4.31 G-PIPK GSA S_{Fi} and S_{Ti} for all Es 153
4.32 G-PIPK PLA summary 155
4.33 G-PIPK prediction with basal ACh 161

B.1 LiRMo reactions after MATLAB import 211
B.2 LiRMo parameters after MATLAB import 213
B.3 LiRMo inputs experiment $(E)_{0–3}$ 213
B.4 LiRMo outputs $E_{0–3}$ 214
B.5 LiRMo $E_{1–3}$ diagnostics plot 215
B.6 LiREM dummy data 216
B.7 LiREM inputs $E_{0–3}$ 218
B.8 LiREM outputs $E_{0–2}$ 218
B.9 LiREM outputs E_3 219
B.10 LiREM optimized θ 219
B.11 LiREM optimized θ comparative scores 220
B.12 LiRMo and LiREM GSA S_{Fi} for each Eo 221
B.13 LiRMo and LiREM GSA S_{Ti} for each Eo 222
B.14 LiRMo and LiREM GSA S_{Fi} for each E 223
B.15 LiRMo and LiREM GSA S_{Ti} for each E 224
B.16 LiRMo and LiREM GSA S_{Fi} for final values of each Eo 225
B.17 LiRMo and LiREM GSA S_{Ti} for final values of each Eo 226
B.18 LiRMo and LiREM GSA S_{Fi} and S_{Ti} for all Es 227

C.1 TM-M outputs $E_{3–5}$ 228
C.2 TM-M outputs $E_{6–8}$ 229
C.3 TM-M output E_9 229
C.4 TM-M optimized parameterization (θ) 230

C.5 TM-M optimized θ comparative scores 230
C.6 TM-M GSA S_{Fi} and S_{Ti} for all E 231

D.1 G-PIPK data 235
D.2 G-PIPK $E_{0,1}$ diagnostics plot 238
D.3 G-PIPK $E_{2,3}$ diagnostics plot 239
D.4 G-PIPK $E_{4,5}$ diagnostics plot 240
D.5 G-PIPK $E_{6,7}$ diagnostics plot 241
D.6 G-PIPK $E_{8–11}$ diagnostics plot 242
D.7 G-PIPK $E_{12,13}$ diagnostics plot 243
D.8 G-PIPK $E_{14,15}$ diagnostics plot 244
D.9 G-PIPK $E_{16–19}$ diagnostics plot 245
D.10 G-PIPK $E_{20,21}$ diagnostics plot 246
D.11 G-PIPK E_{22} diagnostics plot 247
D.12 G-PIPK second optimized θ 248
D.13 G-PIPK third optimized θ 249
D.14 G-PIPK PLA for all Es $P_{1–12}$ 250
D.15 G-PIPK PLA for all Es $P_{13–24}$ 251
D.16 G-PIPK PLA for all Es $P_{25–35}$ 252
D.17 G-PIPK prediction with ACh peak 253
D.18 G-PIPK prediction with ACh dip 254
D.19 G-PIPK prediction with basal ACh plus glutamate peak 255
D.20 G-PIPK prediction with ACh peak plus glutamate peak 256
D.21 G-PIPK prediction with ACh dip plus glutamate peak 257

List of Tables

1.1 Glutamate receptors 13

3.1 Fields utilized in the ‘Compound’ sheet of the SBtab 57

4.1 LiRMo and LiREM parameters 85
4.2 LiRMo and LiREM species as present in the SBtab 88
4.3 LiRMo θ used in figure 4.6 93
4.4 LiRMo scores in figure 4.6 given θ from table 4.3 94
4.5 LiRMo dummy data generating parameters 100
4.6 TM-M parameters 118
4.7 G-PIPK experimental design used for calibration and validation 138
4.8 Cell and spine typical dimensions 157
4.9 Cell and spine species concentrations 159

A.1 Fields utilized in the ‘Reaction’ sheet of the SBtabs 205
A.2 Fields utilized in the ‘Parameter’ sheet of the SBtabs 206
A.3 Fields utilized in the ‘Output’ sheet of the SBtabs 207
A.4 Fields utilized in the ‘Experiments’ sheet of the SBtabs 208
A.5 Fields utilized in the individual experiments sheets of the SBtabs 208
A.6 Fields utilized in the individual experiments input sheets of the SBtabs 209

B.1 LiRMo and LiREM kinetic laws and reactions as present in the SBtab 210
B.2 LiRMo and LiREM parameters as present in the SBtab 212
B.3 LiREM dummy data generating parameters 217

D.1 G-PIPK free parameters 233
D.2 G-PIPK constrained parameters 234
D.3 G-PIPK data generating parameters 1 236
D.4 G-PIPK data generating parameters 2 237

List of Algorithms

1 Interior point method pseudocode 62
2 Simulated annealing pseudocode 63
3 Pattern search pseudocode 66
4 Genetic algorithm pseudocode 67
5 Particle swarm pseudocode 69
6 Surrogate optimization pseudocode 71

Acronyms

Term	Description
O	output
S_{Fi}	first order sensitivity index
S_{Ti}	total order sensitivity index
μ	mean
$\mu^*_{d_p}$	mean absolute effect
σ	standard deviation
θ	parameterization
d_p	elementary effect
f	function
s_{Eo}	score per experimental output
s_E	score per experiment
s_f	scoring function
s_t	total score
2-AG	2-arachidonoylglycerol
5HT3R	5-hydroxytryptamine (serotonin) type-3 receptor
7TMD	seven-transmembrane domain
A	adenosine
A21	cAMP-regulated phosphoprotein 21 kDa (usually ARPP-21)
A2AR	adenosine A2A receptor
AC	adenylyl cyclase
AC5	adenylyl cyclase type 5
ACh	acetylcholine
AChE	acetylcholinesterase
ADH	alcohol dehydrogenase
ADHD	attention deficit hyperactivity disorder
ADP	adenosine diphosphate
aE	active effector
AEA	anandamide
aET	active effector total amount

Term	Description
ALDH	aldehyde dehydrogenase
AMPAR	α-amino-3-hydroxy-5-methyl-4-isoxazole propionic acid receptor
API	Application Programming Interface
aR	active receptor
aRL	active receptor bound to ligand
aRT	active receptor total amount
ATP	adenosine triphosphate
BDNF	brain-derived neurotrophic factor
BIDS	Brain Imaging Data Structure
Ca^{2+}	calcium ion
CaM	calcium-calmodulin-modulated protein
CaMKII	calcium-calmodulin-dependent protein kinase II
cAMP	cyclic adenosine monophosphate
$Cav_{1.3}$	calcium channel, voltage-dependent, L type, alpha 1D subunit
CB1	cannabinoid receptor type 1
Cdk5	cyclin-dependent kinase 5
CDPDAG	cytidine diphosphate-diacylglycerol
cGMP	cyclic guanosine monophosphate
ChAT	choline acetyltransferase
ChIN	cholinergic interneuron
Ci	confidence intervals
CK1	casein kinase 1
Cl^-	chlorine ion
CNS	central nervous system
COMT	catechol-O-methyl transferase
CR	calretinin
Cr	confidence regions
CREB	cAMP response element-binding protein
D	distribution
D1R	D1 dopamine receptor
D2R	D2 dopamine receptor

Term	Description
D32	dopamine and cAMP regulated phosphoprotein 32 kDa (usually DARPP-32)
D5R	D5 dopamine receptor
DA	dopamine
DAG	diacylglycerol
DAT	dopamine transporter
DGL	DAG lipase
DHPG	dihydroxyphenylglycine
DOPA	Dihydroxyphenylalanine
dSPN	direct spiny projection neuron
E	experiment
eCB	endocannabinoid
Eo	experimental output
ER	endoplasmic reticulum
ERK	extracellular signal-regulated kinase
FA	fast-adapting
FAIR	findability, accessibility, interoperability, and reusability
FGF	fibroblast growth factor
FRET	fluorescence resonance energy transfer
FS	fast-spiking
G_{α}	G-protein α domain
$G_{\beta\gamma}$	G-protein β and γ domain
G-PIPK	GPCR and G-protein (G), PLC (P), PI/PIP/PIP2 (PIP), IP3 (IP), and Kv7.2/3 with its associated current (K)
G-protein	guanine nucleotide-binding protein
G_i	G-protein type i
G_{olf}	G-protein type olf
G_o	G-protein type o
G_q	G-protein type q
G_s	G-protein type s
Ga_PLC_T	the total amount of G_{α} bound to PLC
Ga_PLC_T_Norm	the normalized version of Ga_PLC_T
GABA	γ-aminobutyric acid

Term	Description
Gabc_T	the total amount of G-protein trimer concentration
Gabc_T_Norm	the normalized version of Gabc_T
GAD	glutamic acid decarboxylase
GAP	GTPase-activating protein
GAT	GABA transporter
GDP	guanosine diphosphate
Glu	glutamate
GPCR	G-protein coupled receptor
GPe	globus pallidus externus
GPi	globus pallidus internus
GPLv3	GNU General Public License version 3
GSA	global sensitivity analysis
GTP	guanosine-5'-triphosphate
HFS	high-frequency stimulation
HVA	homovanillic acid
icdf	inverse cumulative distribution function
ID	identification
iE	inactive effector
iET	inactive effector total amount
iGluR	ionotropic glutamate receptor
Ino	inositol
IP3R	inositol trisphosphate receptor
IP_3	inositol 1,4,5-trisphosphate
IP_3_PH_T	the total amount of PH probes bound to IP_3
IP_3_T	the total amount of IP_3
iR	inactive receptor
iRL	inactive receptor bound to ligand
iRT	inactive receptor total amount
iSPN	indirect spiny projection neuron
K^+	potassium ion
KCNQ2/3	potassium channel NQ 2 or 3
Kv7.2/3	potassium voltage-gated 7 channel
L	Ligand

Term	Description
L-DOPA	L-3,4-dihydroxyphenylalanine
lb	lower bound
Lhx6	LIM homeobox protein 6
LiREM	Ligand-Receptor-Effector Model
LiRMo	Ligand-Receptor Model
LL	log likelihood
LSA	local sensitivity analysis
LTD	long-term depression
LTP	long-term potentiation
LTS	low-threshold spiking
M1R	muscarinic acetylcholine receptor M1
M1R_ACh_T	the total amount of M1R bound to ACh
M1R_ACh_T_Norm	the normalized version of M1R_ACh_T
M1R_Gbc_T	the total amount of M1R bound to $G_{\beta\gamma}$
M1R_Gbc_T_Norm	the normalized version of M1R_Gbc_T
M1R_OxoM_T	the total amount of OxoM bound to M1R
M1R_OxoM_T_Norm	the normalized version of M1R_OxoM_T
M4R	muscarinic acetylcholine receptor M4
M_current	the PIP_2-dependent Kv7.2/3 membrane current
M_current_Norm	the normalized version of M_current
mAChR	muscarinic ACh receptor
MAO	monoamine oxidase
MAPK	mitogen-activated protein kinase
MEK	MAPK/ERK Kinase
Mg^{2+}	magnesium ion
mGluR	metabotropic glutamate receptor
MGluR_DHPG_T	the total amount of MGluR bound to DHPG
MGluR_DHPG_T_Norm	the normalized version of MGluR_DHPG_T
MGluR_Gbc_T	the total amount of MGluR bound to $G_{\beta\gamma}$
MGluR_Gbc_T_Norm	the normalized version of MGluR_Gbc_T
MGluR_Glu_T	the total amount of MGluR bound to Glu
MGluR_Glu_T_Norm	the normalized version of MGluR_Glu_T
ML	maximum likelihood

Term	Description
MLE	maximum likelihood estimate
Na^+	sodium ion
nAChR	nicotinic ACh receptor
Nb43	nanobody 43
NGF	neurogliaform
NMDAR	N-methyl-D-aspartate receptor
NO	nitric oxide
NOS	nitric oxide synthase
NPY	neuropeptide Y
NWB	Neurodata Without Borders
ODE	ordinary differential equation
OxoM	oxotremorine M
P	parameter
PA	phosphatidic acid
PDB	Protein Data Bank
PH	pleckstrin homology
PH_T	the total amount of PH probes
PI	phosphatidylinositol
PIP	phosphatidylinositol 4-phosphate
PIP2_PH_T	the total amount of PH probes bound to PIP_2
PIP2_PH_T_Norm	the normalized version of PIP2_PH_T
PIP2_T	the total amount of PIP_2
PIP_2	phosphatidylinositol 4,5-bisphosphate
PIS	phosphatidylinositol synthase
PKA	protein kinase A
PKC	protein kinase C
PKG	protein kinase G
PL	profile likelihood
PLA	profile likelihood analysis
PLC	phospholipase C
PLD	phospholipase D
PP1	protein-phosphatase 1
PP2A	protein-phosphatase 2A

Term	Description
PP2B	protein-phosphatase 2B, also know as calcineurin or PPP3
PPN	pedunculopontine nucleus
PSD	post-synaptic density
pSubstrate	phosphorylated Substrate
PV	parvalbumin
R	reaction
r	radius
RAF	rapidly accelerated fibrosarcoma
RaRTR	ratio of active receptor to total receptor amount
RAS	rat sarcoma virus
RGS4	regulator of G-protein signalling 4
S	species
Sa	surface area
SABI	spontaneously active bursty interneurons
SBML	Systems Biology Markup Language
SBtab	Systems Biology table format
Ser	serine
sGC	soluble guanylyl cyclase
SNARE	Soluble N-ethylmaleimide sensitive factor Attachment protein REceptor
SNc	substantia nigra pars compacta
SNr	substantia nigra pars reticulata
SOM	somatostatin
SPN	spiny projection neuron
STDP	spike-timing-dependent plasticity
STEP	striatal-enriched protein tyrosine phosphatase
STN	subthalamic nucleus
TAN	tonically active neurons
TCA	tricarboxylic acid
TH	tyrosine hydroxylase
Thr	threonine
TM-M	Timing Model-Modified
TM-O	Timing Model-Original

Term	Description
TRPV1	transient receptor potential vanilloid 1
ub	upper bound
V	volume
VA	ventral anterior
VAChT	vesicular ACh transporter
VFTD	Venus flytrap domain
VGCC	voltage-gated calcium channel
VL	ventral lateral
VMAT	vesicular monoamine transporter
WSSR	weighted sum of square residuals

1 Introduction

Mathematical modelling and computational approaches have become indispensable tools for understanding the complex dynamics of neural systems across multiple scales [12–14]. From molecular interactions within individual neurons to the emergent behaviours of large-scale brain networks, computational neuroscience provides a framework to integrate diverse experimental data, generate testable hypotheses, and elucidate the underlying mechanisms of brain function and dysfunction [15–17]. These approaches are particularly valuable for studying intricate sub-cortical structures like the basal ganglia, where the interplay between cellular, circuit, and systems-level processes gives rise to critical functions in motor control, learning, and decision-making [18–20]. However, there is a significant gap in accessible, user-friendly computational tools that adhere to findability, accessibility, interoperability, and reusability (FAIR) principles, particularly for modelling biochemical pathways in neuroscience. By providing a robust and adaptable framework, this thesis makes a significant contribution to the computational neuroscience tool-kit. It facilitates current research and lays the groundwork for future advancements, with the potential to accelerate discoveries and deepen our understanding of complex neurobiological systems.

In this chapter, the brain anatomical and functional regions relevant for this dissertation will be presented, as well as the main cell types, molecules and signalling pathways. Additionally, the different levels of neuorbioogical models will be summarized and the FAIR principles will be introduced.

1.1 Basal ganglia and related structures

The basal ganglia is a group of nuclei that resides deep inside both brain hemispheres [18]. It is functionally diverse and consists of the striatum (including the caudate and the putamen), the globus pallidus, the subthalamic nucleus (STN), and the substantia nigra pars reticulata (SNr) [21]. It primarily receives glutamatergic input from the cortex and the thalamus and dopaminergic input from the substantia nigra pars compacta (SNc) [22–25] and itoutputs to the ventral anterior (VA) and ventral lateral (VL) nuclei of the thalamus [26]. It is the reinforcement learning centre of the brain [27, 28] and it is important for the initiation of movements [29]. A summary of all its elements and connections described in this thesis can be found in figure 1.1

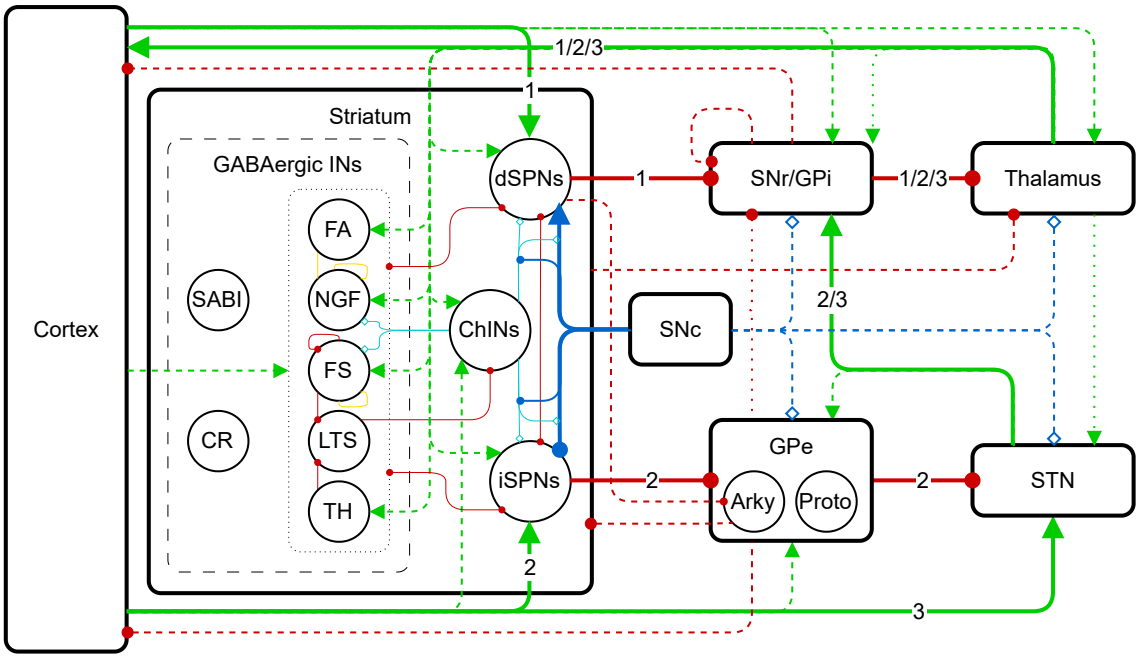


Figure 1.1: Basal ganglia connections and neurotransmitters. Lines represent established connections between cortex, thalamus and basal ganglia regions and neurons. The shape of the lines has been used to describe if the connections are intrastriatal (thin lines), classical described (thick lines) or recently described (dashed lines) connections between basal ganglia elements cortex and thalamus. Line termination represent synapse effect on the receiving neuron, triangle: excitatory, circle: inhibitory, diamond: modulatory. Colours represent the type of synapses present, green: glutamatergic, red: GABAergic, blue: dopaminergic, cyan: cholinergic, gold: electrotonic (gap junctions). Numbers 1 to 3 represent the classically described direct, indirect, and hyper-direct pathways, respectively. Abbreviations: cholinergic interneurons (ChINs), direct spiny projection neurons (dSPNs), fast-adapting (FA), fast-spiking (FS), globus pallidus externus (GPe), globus pallidus internus (GPi), indirect spiny projection neurons (iSPNs), low-threshold spiking (LTS), neurogliaform (NGF), substantia nigra pars compacta (SNc), substantia nigra pars reticulata (SNr), subthalamic nucleus (STN). Original diagram with information from multiple sources [18, 19, 21–26, 30–63].

1.1.1 Striatum

The dorsal striatum, consisting of the caudate and putamen and referred hereafter as just striatum, is the main input nucleus of the basal ganglia, mostly receiving signals from the cortex [24], thalamus [30], and the SNc [25], as delineated by the functional anatomy outlined by Parent and Hazrati [31]. In rodents, its dorsolateral part engages in the control of innate movements and habits [64–66], while the dorsomedial region participates in goal-directed movements [29].

The striatum consists mostly of γ-aminobutyric acid (GABA)ergic projection neurons and interneurons [67], which will be described in greater detail in section 1.2. The projection neurons are, as the name indicates, of special importance for the connections of the striatum with other brain regions. They receive the inputs to the striatum and project mostly to the

globus pallidus [32].

These projection neurons are subdivided into two distinct populations, direct spiny projection neurons (dSPNs) and indirect spiny projection neurons (iSPNs) [19, 33, 34]. The name is due to their relative direct and indirect connection to the globus pallidus internus (GPi)/SNr (also known as the output nucleus of the basal ganglia). The dSPNs directly inhibit the GPi/SNr while the iSPNs inhibit the globus pallidus externus (GPe). The GPe subsequently also has inhibitory projections to the STN which, in turn, stimulates the GPi/SNr. Since the GPi/SNr acts by inhibiting the thalamus, the result is that dSPNs facilitate actions by disinhibiting the thalamus, and the iSPNs suppress actions by increasing thalamic inhibition [32]. The neurons in these two pathways work together to regulate action initiation and suppression [24, 35, 36, 68].

Dopamine (DA), released from the SNc, plays a significant role in regulating the interneurons of the striatum [19, 33, 34, 37, 38]. Both types of spiny projection neurons (SPNs) have receptors for DA, but while the dSPNs express D1 dopamine receptor (D1R), the iSPNs express D2 dopamine receptor (D2R) [19, 33, 34, 37, 38]. This biochemical difference alters the characteristics of their glutamatergic corticostriatal synapses, making the former susceptible to having their intrinsic excitability increased, promoting long-term potentiation (LTP), and the latter prone to having their intrinsic excitability decreased, promoting long-term depression (LTD) [19, 38, 69]. These complementary influences lead to the same outcome due to the antagonistic role of the direct and indirect pathway neurons. The result is the promotion of action initiation when DA is released, so that actions previously associated with positive outcomes are preferentially selected in a form of reward learning [30, 38, 69–71].

Movement disorders can be caused by dysfunctions of the striatal connections [35, 36]. Disorders characterized by excessive and abnormal movements, known as hyperkinetic disorders, are believed to stem from specific damage to the striatal neurons that connect to the GPe. On the other hand, disorders associated with reduced movement, hypokinetic disorders, are thought to be caused by a multifaceted series of alterations in the activity of SPNs, leading to heightened output from the basal ganglia [36]. Problems with the release of DA from the SNc, which reduce the activity of SPNs, can notably lead to such a hypokinetic disorder in the form of Parkinson's disease [38, 72, 73].

1.1.2 Globus pallidus

As previously mentioned, the globus pallidus, a key structure within the basal ganglia, is divided into two distinct segments: the GPi and GPe. These components are distinguished

by their roles, contributions, and connections within the broader structure of the basal ganglia.

Globus pallidus externus

The GPe is constituted by three cell populations: arkypallidal GPe neurons, LIM homeobox protein 6 (Lhx6)-positive, and parvalbumin (PV)-positive [74]. The latter two are also referred to as prototypical GPe neurons [39, 75]. It mainly receives inhibitory GABAergic input from the iSPNs within the striatum, and projects to the STN with inhibitory GABAergic input of its own, in what has been described as the indirect pathway of the basal ganglia [24, 32, 35, 36]. In recent years, new connections have been described that connect the GPe with other brain and basal ganglia regions. Examples are its afferent [40, 41] and efferent [42, 43] connections to the cortex and its projections back to the striatum (mostly by arkypallidal cells) [44, 75].

Globus pallidus internus

The GPi is, in conjunction with the SNr, the 'output' nucleus of the basal ganglia. Classically it was described as being innervated by inhibitory GABAergic projections, primarily from the dSPNs of the striatum (basal ganglia direct pathway) and glutamatergic projections from the STN (basal ganglia indirect and hyperdirect pathways) [24, 32, 35, 36, 45], although in recent years it has been established that it receives inputs from almost all the basal ganglia and even the cortex [40, 41]. Its GABAergic projections where first described as innervating the pedunculopontine nucleus (PPN) of the thalamus [32, 35, 36], more specifically its VA and VL regions, resulting in its constant inhibition. More recently it has also been described that it projects to other basal ganglia regions as well, and even to the cortex itself [41].

1.1.3 Subthalamic nucleus

The STN, another crucial structure within the basal ganglia circuitry, functions as an excitatory node in this intricate network. It primarily receives inhibitory GABAergic inputs from the GPe and, in turn, dispatches excitatory glutamatergic signals to the GPi and the SNr. This role makes the STN a major contributor to the indirect pathway of the basal ganglia [24, 32, 35, 36]. Apart from its classical role in the indirect pathway, the STN has also been found to be directly stimulated by the cortex in the hyperdirect pathway, which serves as another way for the cortex to control movement selection and initiation [45].

Given the tonically active nature of the GPe and its continuous inhibitory GABAergic

output, under normal conditions, the STN typically maintains a state of reduced activity. This balance is disrupted when the inhibitory input from the GPe diminishes during the activation of the indirect pathway of the basal ganglia. Following this, the now excited STN increases its glutamatergic output to the GPi and the SNr. This increased excitation results in heightened inhibitory activity in the GPi and SNr, culminating in the suppression of thalamic activity [24, 32, 35, 36]. This sequence of events contributes to the inhibition of unnecessary or competing motor activities, thus closing the loop of the alternate thalamic regulation.

1.1.4 Substantia nigra

The substantia nigra, an integral component of the basal ganglia's complex network, is anatomically and functionally divided into two primary sectors: the SNc and the SNr. It significantly contributes to the basal ganglia's essential functions in the regulation of motor control, the initiation of actions, and the process of reinforcement learning. Moreover, any alterations or pathologies affecting the substantia nigra, such as Parkinson's disease, can lead to severe motor and cognitive deficits, underscoring the importance of this structure in maintaining optimal brain function.

Substantia nigra pars compacta

The SNc receives inputs from the cortex [46] and is distinctively populated with dopaminergic neurons, making it an important part of the basal ganglia circuitry. These neurons project abundantly to the caudate and the putamen, and less extensively to the rest of the basal ganglia, establishing a vital dopaminergic pathway that plays an indispensable role in modulating the direct and indirect pathways of the basal ganglia circuitry [37, 38]. As referenced in the striatum section 1.1.1, DA released from SNc neurons interacts with D1R and D2R on dSPNs and iSPNs, respectively [19, 33, 34, 37, 38]. It acts in a complementary fashion by activating the former and suppressing the latter, thus aiding in the balance of motor activity initiation and suppression [19, 33, 34, 37, 38, 69].

The SNc's dopaminergic neurons also participate in reward learning through the release of DA associated with the promotion of actions that have led to positive outcomes in the past. This association of DA release with reward mechanisms underlines the SNc's significant contribution to the basal ganglia's role as the brain's reinforcement learning centre [38, 69, 70].

As referred previously in section 1.1.1, problems with the release of dopamine from the SNr can lead to movement disorders, the most well-known of which is Parkinson's disease

[38, 72, 73].

Substantia nigra pars reticulata

Together with the GPi, the SNr constitutes the output nucleus of the basal ganglia. Like the GPi, it primarily receives inputs from inhibitory GABAergic projections coming from striatal dSPNs (direct pathway) and glutamatergic projections from the STN (basal ganglia indirect and hyperdirect pathways) [24, 32, 35, 36, 45]. In addition to these, it can also receive direct glutamatergic input from the cortex [40, 46], GABAergic input from itself, cholinergic input from the PPN, and dopaminergic input from the SNc [47]. In turn, the SNr projects principally to the thalamus [24, 32, 35, 36], more specifically its ventromedial and intralaminar regions in rodents [48] as well as the VL and VA regions in primates [47], and to brain stem regions such as the PPN and the midbrain reticular formation [48].

1.1.5 Thalamus

The thalamus, particularly through its VA and VL in primates [47], and its ventromedial and intralaminar regions in rodents [48], plays a vital role in the basal ganglia-thalamocortical circuitry [24, 32, 35, 36]. As previously mentioned, it is the primary recipient of inhibitory GABAergic projections from the GPi and the SNr. It relays information back to the cortex, thereby positioning itself as the 'gateway' in transmitting the processed signals from the basal ganglia, and completing the cortical-basal-thalamic loop. It is tonically inhibited but some of its regions can be selectively disinhibited when the direct pathway is activated, which is relevant for certain actions. This, in conjunction with suppression of nearby regions by the indirect and hyperdirect pathways (which could lead to competing movements), leads to initiation of specific actions while suppressing unwanted ones [24, 32, 35, 36, 45]. Apart from these classical connections, the thalamus, through its centromedian nucleus, establishes reciprocal connections with the striatum [30, 71] and is directly innervated by the cortex [40]. These excitatory glutamatergic signals add another layer of complexity to its functional connectivity.

1.2 Neurons of the striatum

Most striatal neurons are the previously introduced dSPNs and iSPNs [19, 33, 34] ($\sim$95% in the rodent striatum) [76]. The neuronal population is completed by a small number of interneurons ($\sim$5% in the rodent striatum) [76] that help modulate the overall neuronal activity

of the striatum. They can be divided into cholinergic and GABAergic interneurons, with the latter further divided into several subtypes such as: fast-spiking (FS), low-threshold spiking (LTS), fast-adapting (FA), calretinin (CR)-positive, spontaneously active bursty interneurons (SABI), neurogliaform (NGF)-like, and tyrosine hydroxylase (TH)-positive interneurons [49, 50, 77].

1.2.1 Spiny projection neurons

SPNs are the projection neurons of the striatum, responsible for relaying signals, mostly from the cortex and thalamus [30], to the globus pallidus. These GABAergic neurons [67] are divided into two sub-populations, the dSPNs and iSPNs, that perform complementary roles in the relay of information through the striatum, as described in section 1.1.1.

The two populations of SPNs are remarkably similar but not identical in their morphological and membrane properties [78–81]. The iSPNs exhibit a smaller dendritic arbour, with fewer primary dendrites compared to dSPNs [80] and are found to be marginally more excitable [81, 82].

On the biochemical level, dSPNs primarily express D1 receptors and substance P, while iSPNs express D2 receptors and enkephalin [33]. These neurochemical markers not only aid in differentiating these neuron types but also offer insights into their functional roles within the striatum, see section 1.1.1.

At the signalling level both share the ability to generate plateau potentials, a characteristic electrophysiological feature, generated by voltage-dependent N-methyl-D-aspartate receptor (NMDAR) in response to clustered glutamatergic synaptic input [83, 84]. Also, both have back-propagating action potentials that elicit a transient calcium ion (Ca^{2+}) influx, which decays with the distance from the soma in both types but at a noticeable faster rate in the dSPNs compared to the iSPNs [81].

1.2.2 Interneurons

Striatal interneurons, although representing a modest proportion of striatal neurons, play a pivotal role in striatal function by being involved in the modulation of striatal output, either directly, by forming inhibitory synapses with SPNs, or indirectly, by influencing the intrinsic properties of SPNs [49–52]. Despite recent advances showing that the major classes of striatal interneurons can be distinguished based on their firing patterns, synaptic connections, and expression of specific molecular markers, a comprehensive understanding of their diverse properties and distinctive roles within the striatal network is yet to be fully elucidated

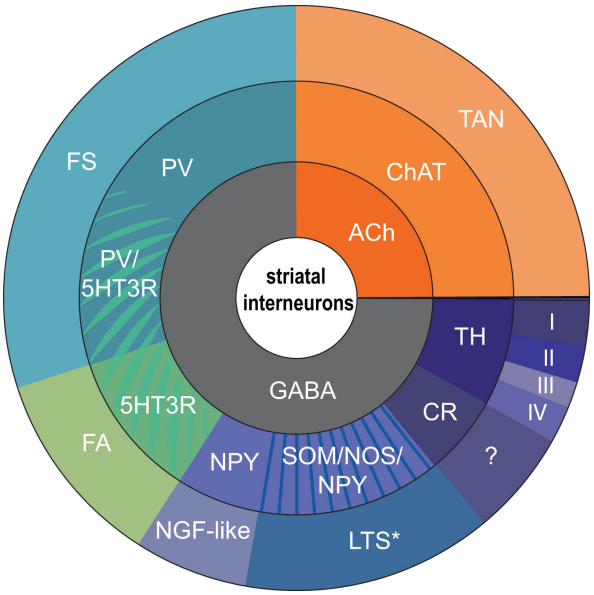


Figure 1.2: Proportions and characteristics of striatal interneurons. Inner donut: main neurotransmitter. Middle donut: Molecular markers. Outer donut: Electrophysiological properties. Abbreviations: acetylcholine (ACh), choline acetyltransferase (ChAT), tonically active neurons (TAN), parvalbumin (PV), fast-spiking (FS), 5-hydroxytryptamine (serotonin) type-3 receptor (5HT3R), fast-adapting (FA), neuropeptide Y (NPY), neurogliaform (NGF), low-threshold spiking (LTS), somatostatin (SOM), nitric oxide synthase (NOS), calretinin (CR), tyrosine hydroxylase (TH). Reproduced with permission from Burke *et al.* [77]

[49, 50], and new types of interneurons continue to be described in recent years [85].

GABAergic interneurons account for three quarters of the striatal interneuron population and are subdivided into different subtypes that will be briefly described in the following paragraphs, partly adapted from Tepper *et al.* [49, 50].

Fast-spiking interneurons

FS interneurons, which also account for about one quarter of the striatal interneurons [77] are not spontaneously active and express the Ca^{2+}-binding protein parvalbumin. There are at least two subtypes of FS interneurons. Morphologically, they present five to eight principal dendrites with their higher order dendrites forming a 200 to 300 μm diameter field, and a very dense highly branched axon 1.5 to 2 times wider than the dendritic field [86]. These neurons are distinguished by their brief action potentials and minimal spike frequency adaptation. As inhibitory neurons, they affect primarily SPNs within a few hundred micrometres from their soma [52], and to a lesser extent, LTS interneurons [53]. They are interconnected via gap junctions [54] and GABAergic synapses [51], which allow fast interneuron signalling, synchronized firing, and robust network interactions.

Low-threshold spiking interneurons

LTS interneurons account for around 20% of the striatal interneurons [77], are spontaneously active GABAergic cells and can be distinguished by the expression of somatostatin (SOM) [87], neuropeptide Y (NPY) [88, 89], or nitric oxide synthase (NOS) [90]. They project to SPNs and cholinergic interneurons (ChINs) and receive signals from the cortex and TH-positive interneurons [55]. With a medium-sized soma, these neurons feature a sparse dendritic arborization with only three to five primary dendrites and possess the most expansive yet least dense axonal arborization among all striatal neurons, with axons potentially extending up to 1 mm in a straight line [86, 91].

Calretinin-positive interneurons

CR-positive interneurons were identified early in the study of the striatum [92, 93], but remain one of the least well-understood types of striatal neurons. Their morphology is quite variable, as they can display a small to medium-sized soma, and their dendritic patterns can range from a few spiny dendrites to a multipolar arrangement of smooth aspiny dendrites [92]. Their spontaneous activity, connectivity with SPNs and with other interneurons is presently unknown [49, 50].

Neurogliaform-like interneurons

NGF-like interneurons are characterized by neuropeptide Y expression and make up a minor portion of striatal interneurons [77]. They are not spontaneously active, have high connectivity to SPNs, and connect to various other interneurons: NGF-like through gap junctions [56], FA through heterosynaptic electrotonic junctions [57], and ChINs through type 2 nicotinic acetylcholine (ACh) receptors [56]. Morphologically, they have a medium-sized soma, a highly compact and dense higher order dendritic field with five to nine aspiny primary dendrites, as well as a very dense axonal arborization extending further than the dendrites [50].

Tyrosine hydroxylase-positive interneurons

TH-positive interneurons, named for their expression of tyrosine hydroxylase, display spontaneous activity. They present low reciprocal connectivity with SPNs and establish connections with other types of interneurons such as LTS [55] and ChINs. They can be subcategorized into types I-IV, with type I as the most common, accounting for approximately

60%, followed by type IV (21%), type II (13%), and finally type III (6%) [94]. All types exhibit a medium-sized soma, and type I can be distinguished by emitting two to four primary dendrites, having higher-order sparsely spinous dendrites, and a dense, highly branched axon populated with varicosities [49, 50].

Fast-adapting interneurons and spontaneously active bursty interneurons

FA and SABI, express 5-hydroxytryptamine (serotonin) type-3 receptor (5HT3R), and have medium-sized soma [58, 95]. FA interneurons feature three to five aspiny varicose dendrites and display a dense axonal field [58], while SABIs emit three to four aspiny dendrites that form a compact dendritic arborization, with their axon extending up to 500 μm from the soma, forming a modestly dense arborization with varicose terminals [95]. While FA interneurons are not spontaneously active and have a moderate connectivity with SPNs [58], SABIs are spontaneously active but do not have relevant connections with SPNs [95]. The connectivity of both types of neurons with other interneurons is presently unknown [50].

Cholinergic interneurons

Finally, to complement the GABAergic interneurons, the ChINs represent about a quarter of the striatal interneurons [77]. They are characterized by their lack of spines and were traditionally referred to as large aspiny interneurons [96]. They act via both nicotinic and muscarinic ACh receptors and target multiple neurons in the striatum including the SPNs [59], LTS [60], FS neurons [61], and the dopaminergic axon terminals originating from the SNc [62]. They express D2R and are thought to be a conduit for LTD of both SPNs [97, 98]. They exhibit spontaneous activity both in vitro and in vivo [99], and transient 'pauses' concurrently with the occurrence of behaviourally salient stimuli and dopamine signalling [100]. Their dendritic arbour extends for approximately 200 μm and compared to SPNs and FS neurons, their dendrites feature fewer bifurcations, with typically around four primary dendrites [101].

1.3 Dendritic synapses of spiny projection neurons

The dendritic synapses of SPN represent a critical site for information integration and plasticity within the basal ganglia circuitry. These synapses are the location where inputs from cortical, thalamic, and midbrain regions converge and are processed to shape striatal output [24, 31, 32, 35, 36]. The synaptic environment is characterized by a complex

interplay of neurotransmitters, primarily glutamatergic signalling from corticostriatal and thalamostriatal afferents, and GABAergic inputs from local interneurons [30, 34]. This neurochemical landscape is further modulated by dopaminergic [19, 33, 37, 38] and cholinergic signalling [30], facilitating various forms of synaptic plasticity, including LTP and LTD [102], which are fundamental to striatal functions such as motor control, learning, and decision-making [25, 29, 64–66, 68–70]. Understanding the structural and functional properties of these synapses, as well as the molecular mechanisms underlying their plasticity, is crucial for elucidating striatal information processing in both physiologic and pathological states. This section will examine the neurotransmitter involved, and the mechanisms of synaptic plasticity, providing insight into how these elements contribute to striatal function.

1.3.1 Neurotransmitters and their receptors

Neurotransmitters constitute the primary chemical mediators of intercellular communication in the central nervous system, playing a fundamental role in the regulation of neurological functions [103]. These endogenous molecules, synthesized and released by neurons, interact with specific receptors to elicit targeted cellular responses [103]. The intricate interplay among diverse neurotransmitter systems forms the basis of neural signalling, modulating a broad spectrum of physiological and cognitive processes [103]. In the context of striatal and basal ganglia function, a comprehensive understanding of these neurotransmitter systems is essential for elucidating the mechanisms underlying motor control, learning, and reward processing [25, 29, 64–66, 68–70]. This subsection provides a detailed examination of the key neurotransmitters relevant to striatal function, their metabolic pathways, and the receptor systems through which they exert their effects.

1.3.1.1 Glutamate

Glutamate is the main neurotransmitter of the central nervous system. It is utilized by about 90% of the brain's neurons, it is involved in 80-90% of synaptic transmissions [104] and its use accounts for roughly 80% of the brain's energy expenditure [105, 106]. Apart from its role as a neurotransmitter, it acts as a precursor molecule for the inhibitory neurotransmitter GABA and participates in the synthesis of crucial proteins and peptides such as the antioxidant molecule glutathione [107].

In the brain, glutamate synthesis is primarily facilitated via the transamination of α-ketoglutarate, a crucial intermediate in the tricarboxylic acid (TCA) cycle. Since this pool of glutamate is constantly being degraded, its homeostatic regulation is achieved through a

process involving astrocytes known as the glutamate-glutamine cycle. In this pathway, glutamate released from presynaptic terminals is rapidly absorbed by adjacent astrocytes and converted to glutamine through the enzyme glutamine synthetase. The resultant glutamine is subsequently released into the extracellular space, reabsorbed by neurons and converted back to glutamate by the enzyme phosphate-activated glutaminase [107–109].

Glutamate's effects at the synaptic level are mediated through two classes of receptors: ionotropic and metabotropic receptors. An overview of its different types and their properties can be found in table 1.1.

Ionotropic glutamate receptors (iGluRs) are ligand-gated ion channels that control the influx or efflux of ions, leading the postsynaptic neuron to either depolarize or hyperpolarize. They are categorized into three types: NMDAR, α-amino-3-hydroxy-5-methyl-4-isoxazole propionic acid receptor (AMPAR), and Kainate receptors. Each receptor type is further divided into subtypes based on their gene families. The primary function of these receptors revolves around facilitating the influx of calcium ion (Ca^{2+}) and sodium ion (Na^{+}) ions [107, 110].

Metabotropic glutamate receptors (mGluRs), a group of G protein-coupled receptors, transduce signals by triggering a series of intracellular cascades that mediate the slower, modulatory actions of glutamate. They are also divided into three main classes: Class I, Class II, and Class III, each marked by specific gene families and performing distinct functions. For instance, activation of Class I mGluRs (which include mGluR1 and 5) leads to an increase in intracellular inositol 1,4,5-trisphosphate (IP_3) and Ca^{2+} levels [107, 111].

In the striatum, glutamate, originating mainly from the corticostriatal pathway, is the main source of excitation of the SPNs [24, 30]. The modulation of the strength of these connections is intimately related with reinforcement learning [30, 38, 69–71]. Their long term potentiation and depression are referred to as LTP and LTD, respectively [19, 38, 69], and further explained in section 1.3.2.

In the striatum, the differential expression of metabotropic and ionotropic glutamate receptors on the dSPNs and iSPNs contributes to the overall excitability of these neurons, thereby shaping the output of the basal ganglia circuitry [111]. These receptors, thus, may contribute to the functional differences observed between the dSPNs and iSPNs, especially in terms of their responses to dopamine and glutamate. Aberrant glutamatergic neurotransmission has been implicated in several neurodegenerative disorders that affect the basal ganglia, such as Parkinson's and Huntington's disease [38, 72, 73, 112].

In conclusion, glutamate serves as a critical neurotransmitter in the basal ganglia, par-

Type	Ionotropic			Metabotropic		
Class	NMDAR	AMPAR	Kainate	Class I	Class II	Class III
Subunits	GluN1 GluN2(A-D) GluN3(A/B)	GluA1-4	GluK1-7	mGluR1/5	mGluR2/3	mGluR4/6-8
Function summary	$\uparrow Ca^{2+}/Na^{+}$			$\uparrow IP_3/Ca^{2+}$	$\downarrow$ cAMP	

Table 1.1: Glutamate receptors are subdivided into two main types, ionotropic and metabotropic. Ionotropic receptors are divided into three classes, NMDAR, AMPAR and kainate. They generally facilitate the entrance of Ca^{2+} and sodium into the neuron but the specificity to specific ions depends on the gene families and receptor subgroups. Metabotropic receptors are also divided into three main classes and each class has various subtypes with similar function and ligands. Class I stimulates phospholipase C leading to IP_3 increase and Ca^{2+} release, while class II and III inhibit adenylyl cyclase, inactivate Ca^{2+} and activate K^{+} channels leading to cyclic adenosine monophosphate (cAMP) decrease [107, 110, 111].

ticularly in the striatum. Its role in regulating the activity of the dSPNs and iSPNs, as well as in mediating synaptic plasticity, underscores its central role in motor control and reinforcement learning. Moreover, the involvement of glutamate in several basal ganglia disorders highlights the importance of maintaining optimal glutamatergic neurotransmission for normal brain function.

Metabotropic glutamate receptor 1 and 5

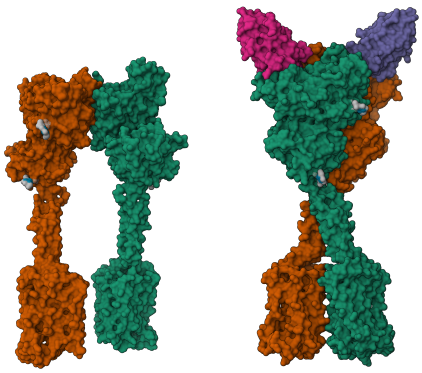
mGluRs are class C G-protein coupled receptors (GPCRs), that have L-glutamate as their endogenous agonist [111]. As other class C GPCRs, they form closely bound dimers and are constituted by an agonist-binding Venus flytrap domain (VFTD) linked to a guanine nucleotide-binding protein (G-protein)-activating seven-transmembrane domain (7TMD) [114]. Their structure can be seen in figure 1.3, VFTD on top, linker in the middle and 7TMD on the bottom. They have eight possible subunits divided into 3 classes as seen in table 1.1. Subunits 1 and 5 are mostly present in postsynaptic regions and bound to Gq while all the other subunits are mostly in presynaptic regions and bind to Gi/o [111]. Although traditionally described as forming only homodimers, it has been recently described that subunits of mGluRs can form heterodimers of more than one subunit [115].

Glutamate is the endogenous agonist of mGluR1 and 5 but they can also bind other synthetic drugs and natural compounds [111]. Of special interest for this thesis, as it is going to be used in the model of section 4.5, is (S)-3,5-dihydroxyphenylglycine, hereafter

referred as dihydroxyphenylglycine (DHPG). This synthetic ligand is an orthosteric agonist of mGluR5, meaning that it binds to the same site as the endogenous agonist, and can be used in experiments to elicit the activation of the protein in a similar fashion to glutamate [111].

After activation, mGluRs undergo a conformational change that brings the membrane region of their two subunits closer, illustrated in figure 1.3 for mGluR5 [113]. This change modifies the connection of intracellular proteins leading to the coupling of a G-protein and triggering its corresponding intracellular signalling [113, 116, 117].

The dynamics of its activation and inactivation have been characterized in detail. Its kinetics follow a curve of the form of $concentration = 1 - e^{-\frac{time}{\tau}}$, with its activation occurring in a agonist concentration-dependent dynamic ranging between 110 and 15 milliseconds and its inactivation is approximately 50 milliseconds regardless of agonist concentration. It can also be sensitized by repeated stimulus giving rise to a potentiated response [116].



(a) Inactive conformation (b) Active conformation

Figure 1.3: Molecular representation of mGluR5. a) mGluR5 in its inactive state before it comes into contact with ligands. b) mGluR5 stabilized in its active conformation bound to L-quisqualate and a synthetic protein coined nanobody 43 (Nb43). Orange and green: two identical receptor domains, blue and magenta: two nanobody 43 (Nb43) proteins, grey: carbohydrates. Rendered from Protein Data Bank (PDB) structure 6N52 and 6N51 respectively, original structure from Koehl *et al.* [113].

1.3.1.2 GABA

GABA, the primary inhibitory neurotransmitter in the central nervous system, plays a crucial role in regulating striatal function. Its importance in the striatum stems from its ability to modulate motor control, cognitive functions, and reward processing by maintaining the delicate balance between excitation and inhibition within striatal circuits. This balance is critical for proper motor function execution and learning process facilitation. Disruption of GABAergic signalling has been implicated in neuropsychiatric disorders such as Parkinson's and Huntington's diseases [35, 38, 43, 49–51, 56, 73, 77, 83].

In the striatum, GABA is predominantly synthesized by GABAergic interneurons and spiny projection neurons [67]. The synthesis pathway begins with glucose as a precursor, which is processed through the TCA cycle to glutamate. Glutamate is then transformed

into GABA by the enzyme glutamic acid decarboxylase (GAD), using pyridoxal phosphate (a derivative of vitamin B6) as a co-factor [118, 119]. This local synthesis allows for precise control of inhibitory signalling within striatal microcircuits.

GABA is removed from the extracellular space by neurons and glia using sodium ion (Na^+)-dependent co-transporters called GABA transporters (GATs). After re-uptake, GABA is converted to succinate and further metabolised in the TCA cycle [120]. In the striatum, this uptake is mainly mediated by GAT-1 and GAT-3. Inhibition of these transporters can lead to increased extracellular GABA concentrations, enhancing the inhibitory tone on striatal neurons. This modulation is critical for both synaptic and extra-synaptic GABAergic transmission, allowing for fine-tuning of striatal output [121].

GABA exerts its effects through two main types of receptors: $GABA_A$ and $GABA_B$. $GABA_A$ is a pentameric ionotropic receptor with chlorine ion (Cl^-) as the main permeant ion. Since the reversal potential for Cl^- is usually lower than the threshold for action potential, activation of these receptors leads to an influx of Cl^- that inhibits postsynaptic cells [122–124]. In the striatum, $GABA_A$ receptors are widely expressed on both projection neurons and interneurons, mediating fast inhibitory synaptic transmission [125, 126]. Interestingly, during early postnatal development, GABA can act as an excitatory neurotransmitter due to the high intracellular chloride concentration in immature neurons. This developmental switch from excitatory to inhibitory GABA action is critical for the proper formation of striatal circuits and may have implications for neurodevelopmental disorders [124, 127].

Conversely $GABA_B$, is a metabotropic receptor similar in structure to other metabotropic receptors, such as mGluR. It assembles into heterodimers of B1 and B2 subunits [117, 128] and, like mGluR, has Venus flytrap domains that bind GABA instead of glutamate [117, 128]. The activation of $GABA_B$ receptors leads, through intracellular cascades, to the activation of potassium ion (K^+) transporters both pre- and postsynaptically, modulating neurotransmitter release and neuronal excitability over longer time-scales than $GABA_A$ receptors [126].

The intricate interplay between GABA synthesis, transport, and receptor activation in the striatum underlies the complex regulation of striatal output. This GABAergic system interacts with other neurotransmitter systems, particularly dopamine, to fine-tune motor control and reward-related behaviours. Understanding these interactions is crucial for unravelling the mechanisms of striatal function in both health and disease.

1.3.1.3 Dopamine

Dopamine is a member of the biogenic amine neurotransmitter family alongside histamine, epinephrine, norepinephrine, and serotonin, which plays a crucial role in various neurological functions [129]. Its primary areas of concentration include the striatum, SNc, and select sympathetic ganglia. Dopamine significance extends to multiple domains of neural function, including movement coordination, motivation, reward processing, and reinforcement learning. Furthermore, it is implicated in the neurobiological mechanisms underlying drug addiction [129].

The dysregulation of dopaminergic signalling is associated with several neurological and psychiatric disorders. Dopamine depletion is a hallmark of Parkinson's disease [72], while its overexpression has been linked to schizophrenia [130]. Additionally, aberrant dopamine signalling is implicated in attention deficit hyperactivity disorder (ADHD) [131] and other movement disorders such as Segawa disease [132].

The biosynthesis of dopamine follows a well-characterized pathway. It begins with L-3,4-dihydroxyphenylalanine (L-DOPA), derived from tyrosine or phenylalanine. In the liver, phenylalanine is converted to tyrosine, which subsequently is taken up by dopaminergic neurons. Within these neurons, tyrosine is hydroxylated to L-DOPA by tyrosine hydroxylase. The final step involves the decarboxylation of L-DOPA to dopamine, catalyzed by Dihydroxyphenylalanine (DOPA) decarboxylase [133]. Notably, L-DOPA was the first molecule employed in the treatment of Parkinson's disease [134].

Following synthesis, dopamine is loaded into synaptic vesicles via vesicular monoamine transporter (VMAT) [135] and subsequently released into the synaptic cleft [136]. The termination of dopamine signalling involves its reuptake by neurons and glia through dopamine transporter (DAT) [137]. This process can be inhibited by substances such as cocaine or amphetamines [129]. The degradation of dopamine involves several enzymes, including monoamine oxidase (MAO), catechol-O-methyl transferase (COMT), alcohol dehydrogenase (ADH), and aldehyde dehydrogenase (ALDH). The final metabolite in this degradation pathway, homovanillic acid (HVA), is excreted in urine [133, 138].

Dopamine exerts its effects through five receptor subtypes, D1R to D5 dopamine receptor (D5R), which can be categorized into two families: D1-like (D1R and D5R) and D2-like (D2R, D3R, and D4R) [139, 140]. These receptor families differ in their G-protein coupling and downstream effects. D1-like receptors couple to G-protein type s (G_s) and G-protein type olf (G_{olf}) proteins, leading to the stimulation of adenylyl cyclase. In contrast, D2-like receptors couple to G-protein type i (G_i)/G-protein type o (G_o) proteins, inhibiting adenylyl

cyclase function [141, 142].

In the striatum, dopamine released from neurons originating in the SNc interacts with D1R on dSPNs and D2R on iSPNs [19, 33, 34, 37, 38]. This differential receptor expression pattern contributes significantly to synaptic plasticity, promoting either LTP or LTD depending on the concentration of dopamine and the presence of other neurotransmitters, which will be discussed in detail in section 1.3.2.

1.3.1.4 Acetylcholine

Acetylcholine (ACh) is the first identified neurotransmitter [143]. It functions at neuromuscular junctions, cardiac muscle synapses, visceral motor system ganglia, and various central nervous system (CNS) sites [144]. In the CNS, ACh primarily acts as a neuromodulator, altering neuronal excitability, modifying presynaptic neurotransmitter release, and coordinating neuronal firing [145–148].

The synthesis of ACh is catalysed by choline acetyltransferase (ChAT) from acetylcoenzyme A and choline [149]. After synthesis, ACh is loaded into synaptic vesicles via vesicular ACh transporter (VAChT) [135, 150] and subsequently released into the synaptic cleft, after an influx of calcium, in a process dependent on SNARE proteins [151, 152]. Acetylcholinesterase (AChE) terminates ACh signalling by hydrolysing it in the synaptic cleft into acetate and choline [153], with the latter being recycled for further ACh synthesis [144].

ACh exerts its effects through distinct receptors and mechanisms, influencing neuromuscular function, CNS activity, and autonomic control [144]. It can either interact with nicotinic ACh receptors (nAChRs) or muscarinic ACh receptors (mAChRs) [144].

nAChRs are ionotropic, nonselective cation channels activated by ACh and nicotine, and antagonised by α-bungarotoxin [154]. Their activation can elicit excitatory or inhibitory responses, depending on the neuronal population [155, 156]. They are pentameric structures composed of various subunit combinations (α, β, δ, γ, ϵ), with α subunits containing the ACh binding sites. The typical stoichiometry is 2α:1β:1δ:$1\gamma/\epsilon$ [144, 157, 158]. Neuronal nAChRs differ from muscle nAChRs in their α-bungarotoxin insensitivity and subunit composition (3α:2β) [144, 157, 158]. ACh binding induces conformational changes, opening the channel gate for ion flux [158].

mAChRs are GPCRs from class A's α-branch [159]. Five subtypes exist, all metabotropic and activated by muscarine. M1, M3, and M5 couple to G-protein type q (G_q)/G_{11} proteins, while M2 and M4 couple to G_i/G_o proteins [160, 161]. They regulate vital functions including cardiac rhythm, smooth muscle activity, glandular secretion, and numerous CNS operations

[162]. They mediate slow postsynaptic responses, modulating ion channels to either inhibit or stimulate neurons [144]. mAChRs are abundant in the forebrain, striatum, and hippocampus [144].

In the striatum, ACh serves as a neuromodulator similar to dopamine, modifying the excitability of SPNs to glutamatergic cortical signalling. It can act through muscarinic acetylcholine receptor M1 (M1R) to inhibit calcium channel, voltage-dependent, L type, alpha 1D subunit ($Cav_{1.3}$) and stimulate G_q pathway in both types of SPNs, and through muscarinic acetylcholine receptor M4 (M4R) to inhibit adenylyl cyclase (AC) and suppress cyclic adenosine monophosphate (cAMP) signalling in iSPNs. This influence is further discussed in sections 1.3.2 and 1.3.3.

1.3.2 Striatal plasticity in corticostriatal synapses

Synaptic plasticity, a fundamental property of neural circuits, enables synapses to modulate their efficacy in response to activity patterns. This mechanism is central to learning and memory processes and is primarily manifested through LTP and LTD.

LTP, first demonstrated in the hippocampal Schaffer collateral pathway [163], results in enhanced synaptic transmission. The induction of LTP is initiated by rapid successive glutamatergic signalling, which leads to substantial postsynaptic depolarization. This depolarization is sufficient to alleviate the voltage-dependent magnesium block of NMDAR, allowing for significant Ca^{2+} influx [164]. The resultant elevation in intracellular Ca^{2+} activates several key enzymes, including calcium-calmodulin-dependent protein kinase II (CaMKII), protein kinase C (PKC), and synaptotagmins [165]. These proteins facilitate the trafficking and insertion of AMPAR to the postsynaptic membrane, thereby increasing the sensitivity to glutamate. Concurrently, Ca^{2+} activation of protein kinases leads to elevated cAMP levels and subsequent activation of protein kinase A (PKA). PKA, in turn, activates the transcription regulator cAMP response element-binding protein (CREB), initiating protein synthesis necessary for enduring structural changes at the synapse [166].

In contrast, LTD diminishes synaptic strength and has been well-characterized in both hippocampal and cerebellar contexts. Hippocampal LTD is typically induced by prolonged low-frequency glutamate release, approximately 1 Hz for 10-15 minutes [167]. This stimulation pattern activates NMDARs to a lesser extent than in LTP, resulting in a moderate Ca^{2+} influx that preferentially activates protein phosphatases, particularly protein-phosphatase 1 (PP1) and protein-phosphatase 2B, also know as calcineurin or PPP3 (PP2B) [168]. These phosphatases initiate signalling cascades that culminate in the internalization of AMPAR via

clathrin-dependent endocytosis. Cerebellar LTD employs a distinct mechanism that requires the concurrent activation of parallel and climbing fibre synapses [169]. Climbing fibre activation elevates intracellular Ca^{2+}, while parallel fibre glutamate activates mGluRs. This dual activation triggers a cascade involving phospholipase C (PLC), which generates diacylglycerol (DAG) and inositol trisphosphate IP_3. IP_3 induces further Ca^{2+} release from the endoplasmic reticulum. The resultant elevated Ca^{2+} levels and DAG activate PKC, leading to substrate phosphorylation and subsequent AMPAR internalization [170].

The following sections will examine how these classical models of LTP and LTD manifest and potentially undergo modifications in corticostriatal synapses onto SPNs. The main focus will be the neurotransmitters involved, intracellular pathways responsible for plasticity, and differences in plasticity between iSPNs and dSPNs. Special attention is also going to be paid to spike-timing-dependent plasticity (STDP) mechanisms. This will provide valuable insight into the specific plasticity mechanisms operating within the basal ganglia circuitry.

1.3.2.1 Long-term depression

LTD in the corticostriatal synapses of SPNs is a complex process mediated by the intricate interplay of various neurotransmitters, including glutamate (Glu), DA, endocannabinoids (eCBs), and ACh.

LTD can be elicited through multiple experimental protocols, including high-frequency stimulation (HFS) and STDP. HFS-induced LTD typically involves trains of pulses at 100Hz in conjunction with postsynaptic neuronal firing [171–173]. This protocol requires postsynaptic depolarization, Ca^{2+} influx through L-type voltage-gated calcium channels (VGCCs), and Ca^{2+}-induced Ca^{2+} release from internal stores, leading to mGluR1/5 activation and increased intracellular Ca^{2+} [174, 175]. Alternatively, LTD can be induced through STDP protocols, where SPN depolarization follows corticostriatal synaptic activation (post before pre) [78, 176]. STDP is discussed in more detail in section 1.3.2.3.

The induction of striatal LTD necessitates the activation of mGluRs, particularly mGluR1 [171, 177, 178], as well as the concurrent activation of both D1 and D2 dopamine receptors [171].

Various intracellular pathways are involved in the onset of LTD in SPNs. Each of the main neurotransmitters involved triggers its own set of response cascades that all interconnect to affect LTD. Each of these is going to be discussed in detail and a visual overview can be seen in figure 1.4.

Glutamate signalling in corticostriatal LTD

Glutamate, particularly acting on mGluR1, is crucial for LTD induction [171, 177, 178]. As explained in section 1.3.3.1, the activation of mGluR leads to downstream activation of two important proteins for glutamate-induced LTD: PLC and PKC. PLC is important for eCB signalling, while PKC phosphorylates several other proteins that contribute to the maintenance of LTD for long periods of time.

Apart from this activity in relation to mGluR, glutamate also activates NMDAR receptors leading to increased intracellular Ca^{2+}. This intracellular Ca^{2+}, depending on its concentration, can have a potentiating (for high Ca^{2+} levels) or depressing (for low Ca^{2+} levels) effect on the SPN synapse [179]. Low Ca^{2+} levels bias the system to PP2B and PP2A activation, while high Ca^{2+} levels bias the system to RAS and CaM activation [179].

Dopamine receptor signalling and DARPP-32

The striatal dopaminergic system operates via two primary receptor types, D1R- and D2R-like, which activate distinct intracellular cascades, often leading to divergent cellular responses [180]. D1R receptors primarily couple to the α subunit of G_s and G_{olf}, promoting cAMP signalling through AC stimulation [181] promoting LTP. Conversely, D2R receptors connect to the α subunit of G_i and G_o, initiating an intracellular cascade that reduces cAMP production and PKA activity [182] promoting LTD.

These signalling pathways have multifaceted downstream effects. PKA, activated by the cAMP pathway, plays a crucial role in regulating various voltage-gated and ligand-gated ion channels through afferent signals [183], and its activity can lead the synapse to either LTP or LTD. A key component in this signalling network is the dopamine and cAMP regulated phosphoprotein 32 kDa (usually DARPP-32) (D32), which is highly expressed in striatal neurons. D32 functions as an inhibitor of PP1 when activated by PKA, which is in turn stimulated by increased cAMP levels [184]. High PP1 activity leads to LTD while reduced activity leads to LTP. The phosphorylation state of D32 is differentially regulated by dopamine receptor activation: D1R stimulation promotes D32 phosphorylation and activation, an effect enhanced by D2R receptor blockage [184, 185].

Endocannabinoid signalling in LTD

The eCB system plays a pivotal role in striatal LTD, particularly in D2R-induced LTD [186]. In the striatum, this signalling system integrates inputs from D2R receptors, mGluRs1/5, and

Ca^{2+} influx via NMDARs and VGCCs, leading to the production of the two primary eCBs, 2-arachidonoylglycerol (2-AG) and anandamide (AEA) [187].

They function as retrograde synaptic messengers, with 2-AG targeting cannabinoid receptor type 1 (CB1) receptors [187] and AEA, in turn, acting as an agonist for transient receptor potential vanilloid 1 (TRPV1) channels [188].

CB1 receptors are strategically expressed on the pre-synaptic membrane of both glutamatergic inputs from the cortex and GABAergic inputs from striatal interneurons [189, 190] and their activation leads to the suppression of neurotransmitter release [191, 192], a process mediated through various second messenger systems. TRPV1 channels are also expressed in the striatum and play a role in regulating glutamate release [193, 194].

CB1 receptor stimulation activates G-proteins, with the G-protein β and γ domain ($G_{\beta\gamma}$) subunit playing a crucial role in inhibiting pre-synaptic Ca^{2+} uptake [191]. This inhibition, along with the activation of G-protein-activated potassium channels, results in a reduction of neurotransmitter release probability [187]. Additionally, CB1 receptor activation leads to the inhibition of VGCCs, further contributing to the suppression of neurotransmitter release [187]. The effects of eCB signalling extend beyond immediate synaptic transmission. Notably, the maintenance of striatal LTD beyond the initial 150 minutes requires pre-synaptic protein synthesis, a process modulated by eCB and CB1 signalling [195].

The eCB system also plays a unique role in enabling biochemical communication between direct and indirect pathways in the striatum [196], underscoring its importance in integrating various aspects of striatal signalling and plasticity.

Cholinergic interneurons (ChINs) and ACh signalling

ChINs are autonomous pacemakers that play a pivotal role in striatal function through their release of acetylcholine (ACh) [197]. They express both D2R, D1R/D5R, and adenosine A2A receptor (A2AR)/D2R heterodimers, enabling them to respond to dopamine and adenosine inputs and modulate ACh release. This modulation affects both pre-synaptic glutamatergic terminals and post-synaptic SPNs [196, 198–200].

DA-controlled ChIN activity regulates eCB LTD in both types of SPNs. D2R activation on ChINs reduces ACh release, decreasing M1R activation on SPNs. This disinhibits Cav1.3 Ca^{2+} channels, increasing intracellular Ca^{2+} and triggering eCB synthesis. The released eCBs activate pre-synaptic CB1s receptors, inhibiting Glu release and promoting LTD in both dSPNs and iSPNs, making ChINs act as intermediaries between iSPNs and dSPNs [97, 196, 201].

In vitro studies have elucidated an intricate intrastriatal circuit where ACh released by ChINs drives DA release, while DA subsequently inhibits ChIN activity through D2R [202]. This feedback loop can result in brief pauses of ACh release when DA acts on ChIN pre-synaptic terminals, creating short windows of enhanced plasticity. This bidirectional interaction is particularly significant for behavioural outcomes [202, 203].

iSPNs

In iSPNs of the indirect pathway, which express D2R, LTD can be reliably induced through various experimental protocols. HFS at 100Hz and STDP protocols have proven effective in triggering LTD in these neurons [171, 174, 204] using a range of techniques, including minimal intrastriatal stimulation [78, 176], macroelectrode stimulation of the cortex [97, 205], and stimulation of the white matter overlaying the striatum [97, 172, 173]. Importantly, both STDP and HFS LTD in D2R iSPNs require the activation of CB1 and D2R, with D2R playing a crucial role in controlling eCB synthesis [97, 176, 187, 206].

The induction of LTD in iSPNs involves a complex signalling cascade. This process is initiated by the activation of post-synaptic G_q-coupled mGluR and L-type calcium channels, which together trigger the release of eCBs [78, 172, 176, 186, 207]. The released eCBs then travel retrogradely across synapses to activate presynaptic CB1 receptors [192, 207]. Sustained activation of CB1 receptors for several minutes induces pre-synaptic LTD, effectively reducing the probability of neurotransmitter release [208].

The regulation of LTD in D2R iSPNs is further modulated by the interplay between D2R and A2AR. Activation of A2AR negatively regulates LTD in these neurons [176, 209], while blockade of D2R by sulpiride inhibits both HFS and STDP [78, 176]. Successful induction of LTD in the indirect pathway requires D2R activation alongside A2AR inactivity [176, 209]. These two receptor types exert opposing effects on cAMP levels and LTD induction [210]. The importance of this signalling pathway is further underscored by studies on adenylyl cyclase type 5 (AC5) knockout mice, which exhibit impaired indirect-pathway LTD, with the deficit occurring upstream of mGluR activation [211]. In iSPNs, A2AR counteracts the inhibitory effect of D2R on cAMP, highlighting the intricate balance required for proper synaptic plasticity regulation.

dSPNs

The mechanisms underpinning LTD in dSPNs and iSPNs exhibit similarities, with the involvement of post-synaptic mGluRs, L-type calcium channels, and pre-synaptic CB1 re-

ceptors, although the activation of these components may be regulated by different neuromodulator receptors [210]. This similarity is corroborated by evidence demonstrating that the activation of L-type calcium channels induces LTD in both direct and indirect SPNs [172]. Moreover, dSPN LTD is inhibited by antagonists of both CB1 and Gq-coupled mGluRs [176], suggesting the involvement of these receptor types in the LTD process.

dSPNs display distinct receptor expression patterns compared to their iSPN counterparts. Notably, they lack D2R and A2AR, instead expressing G_s-coupled D1R and Gi-coupled M4R as their respective equivalents [20, 210, 212]. This configuration reverses the role of DA in modulating the cAMP pathway between iSPNs and dSPNs. In iSPNs, DA (via D2R) inhibits cAMP production while adenosine (via A2AR) promotes it. Conversely, in dSPNs, DA (via D1R) enhances cAMP production, while ACh (via M4R) inhibits it.

Experimental studies on LTD induction in dSPNs have yielded complex results. Shen *et al*. observed LTD in dSPNs during STDP protocols when D1R were inhibited [176]. This finding contrasted with previous studies that had failed to detect LTD [78], likely due to intrastriatal electrode-stimulated DA release activating D1R and consequently inhibiting direct-pathway LTD. In D1R dSPNs, intrastriatal stimulation did not induce HFS [78] or STDP LTD [176]. While cortical macroelectrode stimulation also failed to trigger STDP LTD [205], it did elicit HFS LTD dependent on CB1 and D2R [97, 172, 173]. Interestingly, blockade of D2R by sulpiride inhibits both HFS and STDP LTD in glutamatergic synapses of D1R dSPNs [173, 213], further complicating our understanding of the role of DA receptors in direct-pathway LTD.

1.3.2.2 Long-term potentiation

Corticostriatal synapses exhibit a distinct form of LTP that, while sharing some similarities with other glutamatergic synapses, possesses unique characteristics. The induction of corticostriatal LTP in ex-vivo preparations initially presented challenges, but specific conditions have been identified that facilitate its occurrence. Early studies employed sharp electrodes and magnesium ion (Mg^{2+})-free extracellular solutions in conjunction with HFS protocols to demonstrate LTP in SPNs [263, 264]. Subsequent research achieved LTP induction under physiological Mg^{2+} levels using perforated patch recordings and positive STDP paradigms in SPNs [176].

A defining feature of striatal LTP is its dependence on the concurrent activation of NMDAR and either dopamine or adenosine signalling [38]. The induction of LTP in this region relies on the engagement of NMDAR [176, 263, 265, 266] and mGluR1 [267]. Dopamine

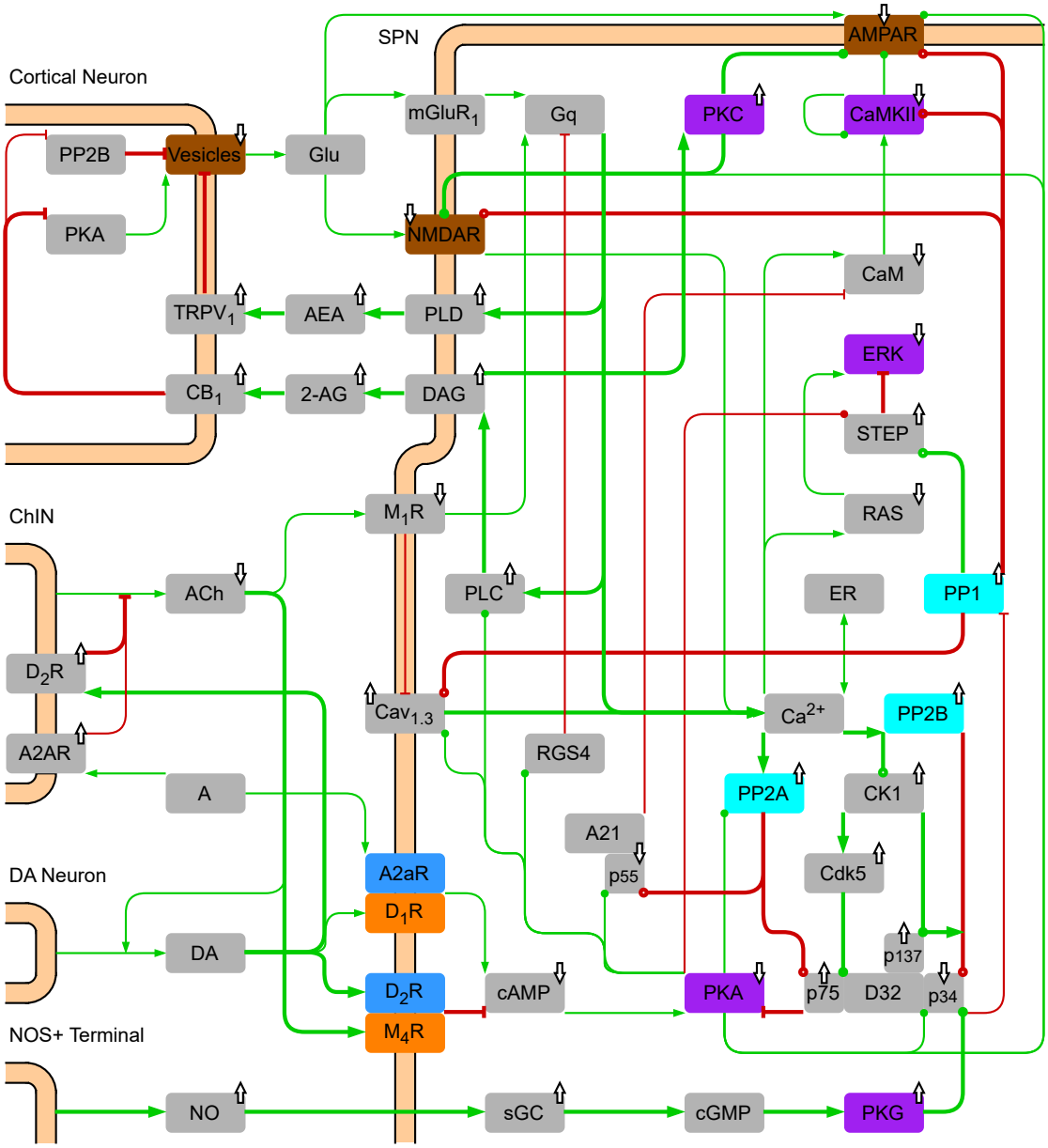


Figure 1.4: Corticostriatal LTD. This diagram illustrates the main biochemical pathways responsible for inducing LTD in the striatum. Up and down arrows represent species that are either up- or down-regulated in various types of LTD. Colour coding: Purple identifies important protein kinases. Light blue identifies important protein phosphatases. Brown represents important final effectors for LTD. Orange represents species only present in dSPNs. Dark blue denotes species only present in iSPNs. Gray boxes indicate other molecules and receptors involved in the pathways. Green and red lines denote activation or inactivation, respectively. Lines ending in arrowheads or bars represent simple activation or inactivation, while lines ending in circles represent either phosphorylation (full circles) or dephosphorylation (empty circles). Thicker lines indicate key pathways that are significantly enhanced during LTD induction. (continues on the next page)

signalling plays a pivotal role in striatal LTP, with studies revealing its necessity [268], specifically through the activation of D1R/D5R [267, 269–271], and its modulatory influence on the process [19, 33, 34, 37, 38]. Conversely, the activation of D2R has been shown to attenuate

Figure 1.4: (Continued) Abbreviations: 2-arachidonoylglycerol (2-AG), adenosine (A), cAMP-regulated phosphoprotein 21 kDa (usually ARPP-21) (A21), adenosine A2A receptor (A2AR), acetylcholine (ACh), anandamide (AEA), α-amino-3-hydroxy-5-methyl-4-isoxazole propionic acid receptor (AMPAR), calcium ion (Ca^{2+}), calcium-calmodulin-modulated protein (CaM), calcium-calmodulin-dependent protein kinase II (CaMKII), cyclic adenosine monophosphate (cAMP), calcium channel, voltage-dependent, L type, alpha 1D subunit ($Cav_{1.3}$), cannabinoid receptor type 1 (CB1), cyclin-dependent kinase 5 (Cdk5), cyclic guanosine monophosphate (cGMP), cholinergic interneuron (ChIN), casein kinase 1 (CK1), D1 dopamine receptor (D1R), D2 dopamine receptor (D2R), dopamine and cAMP regulated phosphoprotein 32 kDa (usually DARPP-32) (D32), D5 dopamine receptor (D5R), dopamine (DA), DAG lipase (DGL), endoplasmic reticulum (ER), extracellular signal-regulated kinase (ERK), glutamate (Glu), G-protein type q (G_q), muscarinic acetylcholine receptor M1 (M1R), muscarinic acetylcholine receptor M4 (M4R), metabotropic glutamate receptor (mGluR), N-methyl-D-aspartate receptor (NMDAR), nitric oxide (NO), nitric oxide synthase (NOS), protein kinase A (PKA), protein kinase G (PKG), phospholipase C (PLC), phospholipase D (PLD), protein-phosphatase 1 (PP1), protein-phosphatase 2A (PP2A), protein-phosphatase 2B, also know as calcineurin or PPP3 (PP2B), rat sarcoma virus (RAS), regulator of G-protein signalling 4 (RGS4), soluble guanylyl cyclase (sGC), spiny projection neuron (SPN), striatal-enriched protein tyrosine phosphatase (STEP), transient receptor potential vanilloid 1 (TRPV1). Original diagram made for this thesis with information from multiple sources [97, 107–110, 113, 116, 117, 175, 179, 182–184, 187, 188, 193, 194, 214–262].

LTP [269, 272].

Downstream signalling cascades have been identified as critical components in the establishment of striatal LTP. The induction of LTP by HFS is prevented by the application of PKA inhibitors [272], and the absence of the protein D32 impairs LTP [270, 273]. At the molecular level, PKA-mediated phosphorylation of the GluN2B subunit of NMDAR enhances its Ca^{2+} permeability, thereby facilitating LTP [274]. The importance of this signalling pathway is further underscored by the observation that mice lacking AC exhibit deficits in striatal LTD [211]. Additionally, LTP appears to involve the regulation of NMDAR and AMPAR by PP1 [184], and DA has been demonstrated to potentiate NMDAR currents via the D1R-D32 signalling axis [275].

Various other factors contribute to the modulation of striatal LTP. Fibroblast growth factor (FGF) and brain-derived neurotrophic factor (BDNF) enhance LTP in SPNs, potentially by augmenting NMDAR currents and activating the MAPK/ERK pathway when A2AR is concurrently activated [266, 276]. Muscarinic receptors also play a role, with M1-like receptors being essential for corticostriatal LTP [277], possibly through the activation of protein kinase C (PKC). Intriguingly, antagonists of M2-like receptors potentiate corticostriatal LTP [278], likely by attenuating the inhibitory influence on glutamate release from corticostriatal fibres.

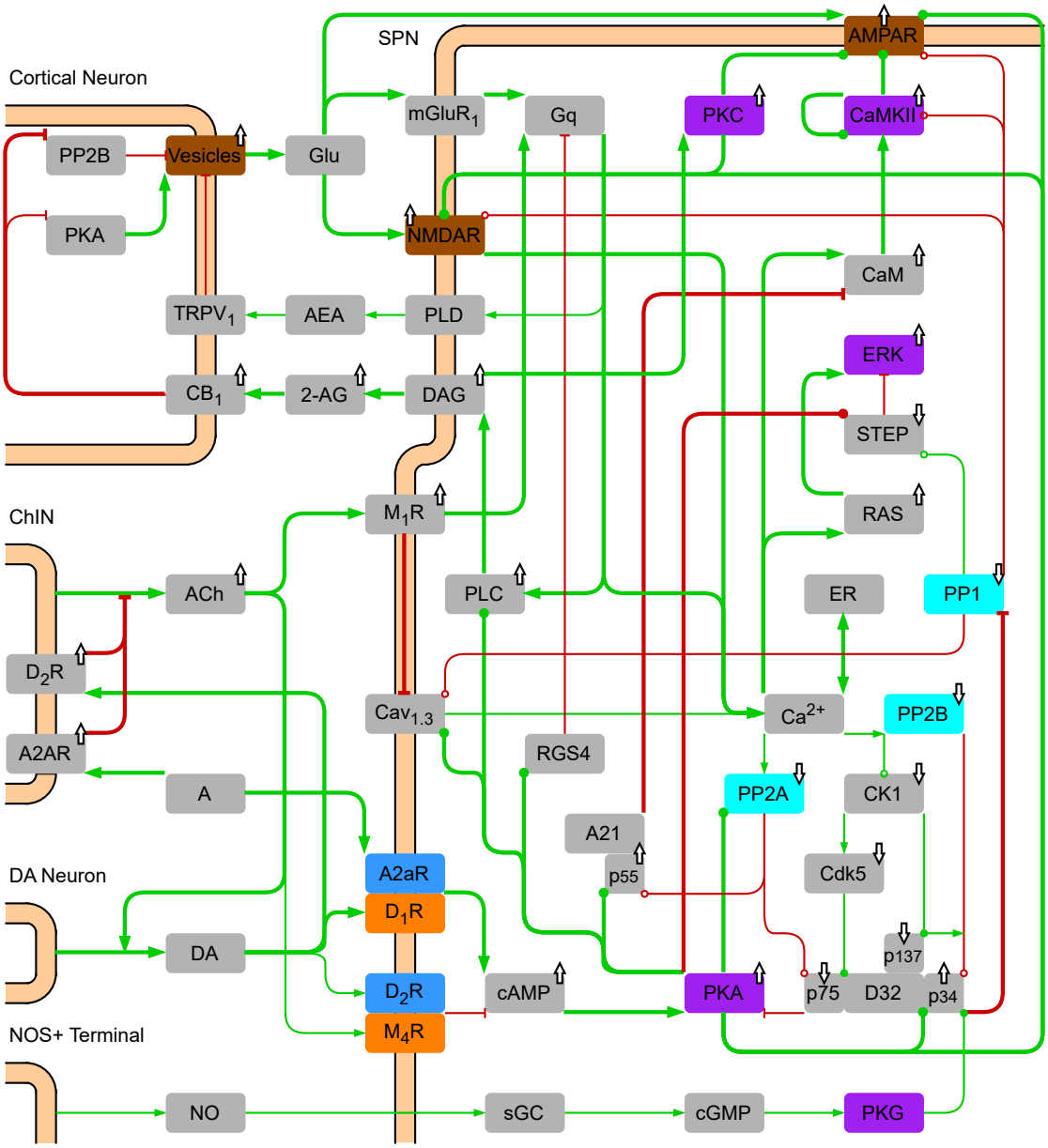


Figure 1.5: Corticostriatal LTP. This diagram illustrates the main biochemical pathways responsible for inducing LTP in the striatum. Up and down arrows represent species that are either up- or down-regulated in various types of LTP. Colour coding: Purple identifies important protein kinases. Light blue identifies important protein phosphatases. Brown represents important final effectors for LTP. Orange represents species only present in dSPNs. Dark blue denotes species only present in iSPNs. Gray boxes indicate other molecules and receptors involved in the pathways. Green and red lines denote activation or inactivation, respectively. Lines ending in arrowheads or bars represent simple activation or inactivation, while lines ending in circles represent either phosphorylation (full circles) or dephosphorylation (empty circles). Thicker lines indicate key pathways that are significantly enhanced during LTP induction. (continues on the next page)

LTP in corticostriatal synapses of iSPNs

The regulation of LTP and LTD in iSPNs is primarily mediated by D2R and A2AR. Synaptic plasticity in these neurons is modulated by the interplay between dopamine and adeno-

Figure 1.5: (Continued) Abbreviations: 2-arachidonoylglycerol (2-AG), adenosine (A), cAMP-regulated phosphoprotein 21 kDa (usually ARPP-21) (A21), adenosine A2A receptor (A2AR), acetylcholine (ACh), anandamide (AEA), α-amino-3-hydroxy-5-methyl-4-isoxazole propionic acid receptor (AMPAR), calcium ion (Ca^{2+}), calcium-calmodulin-modulated protein (CaM), calcium-calmodulin-dependent protein kinase II (CaMKII), cyclic adenosine monophosphate (cAMP), calcium channel, voltage-dependent, L type, alpha 1D subunit ($Cav_{1.3}$), cannabinoid receptor type 1 (CB1), cyclin-dependent kinase 5 (Cdk5), cyclic guanosine monophosphate (cGMP), cholinergic interneuron (ChIN), casein kinase 1 (CK1), D1 dopamine receptor (D1R), D2 dopamine receptor (D2R), dopamine and cAMP regulated phosphoprotein 32 kDa (usually DARPP-32) (D32), D5 dopamine receptor (D5R), dopamine (DA), DAG lipase (DGL), endoplasmic reticulum (ER), extracellular signal-regulated kinase (ERK), glutamate (Glu), G-protein type q (G_q), muscarinic acetylcholine receptor M1 (M1R), muscarinic acetylcholine receptor M4 (M4R), metabotropic glutamate receptor (mGluR), N-methyl-D-aspartate receptor (NMDAR), nitric oxide (NO), nitric oxide synthase (NOS), protein kinase A (PKA), protein kinase G (PKG), phospholipase C (PLC), phospholipase D (PLD), protein-phosphatase 1 (PP1), protein-phosphatase 2A (PP2A), protein-phosphatase 2B, also know as calcineurin or PPP3 (PP2B), rat sarcoma virus (RAS), regulator of G-protein signalling 4 (RGS4), soluble guanylyl cyclase (sGC), spiny projection neuron (SPN), striatal-enriched protein tyrosine phosphatase (STEP), transient receptor potential vanilloid 1 (TRPV1). Original diagram made for this thesis with information from multiple sources [97, 107–110, 113, 116, 117, 175, 179, 182–184, 187, 188, 193, 194, 214–262].

sine concentrations. Elevated dopamine levels coupled with low adenosine concentrations promote LTD, whereas the converse scenario is crucial for STDP-LTP. The latter specifically requires A2AR activation and D2R inactivation [176, 210, 266]. This modulatory mechanism is likely facilitated through the opposing effects of these receptors on NMDAR signalling, which is mediated by PKA modulation. Additionally, A2ARs enhance Ca^{2+} influx through NMDARs in iSPNs via their coupling to adenylyl cyclase [279].

LTP in corticostriatal synapses of dSPNs

The induction of LTP in dSPNs is fundamentally dependent on the intricate interplay between D1R and NMDAR. These receptors form extra-synaptic clusters that dissociate upon D1R activation, thereby facilitating the translocation of NMDAR to synaptic sites and promoting LTP [280]. This dynamic receptor interaction is further corroborated by observations in cultured striatal neurons, where NMDAR demonstrates the capacity to sequester diffusible D1R, although this phenomenon awaits confirmation in mature SPNs [281].

The induction of LTP in dSPNs necessitates the activation of D1R coupled to adenylyl cyclase (AC) and PKA [38]. This process is augmented by STDP (pre then post), which elicits LTP at direct-pathway synapses in perforated-patch recordings, requiring the concurrent activation of both NMDAR and D1R [176, 269]. Furthermore, D1R activation induces the

phosphorylation of mGluR5, leading to an enhancement of NMDAR currents through Pin1-mediated isomerization of that phosphorylated residue. This process is contingent upon Homer 1a-mediated disruption of mGluR5 scaffolding [282].

1.3.2.3 Spike timing-dependent plasticity

First described by Fino *et al.* in 2005, STDP arises when the firing of the pre-synaptic corticostriatal terminal fires in close time proximity with the post-synaptic SPN dendrite [204]. Pre-synaptic firing followed by post-synaptic firing is going to be referred to as pre-post and post-synaptic firing followed by pre-synaptic firing as post-pre.

Early studies found contradictory results as to what exactly should be the STDP effect. Fino *et al.* describe an anti-Hebbian paradigm, pre-post leading to LTD and post-pre leading to LTP, while different groups reported the opposite Hebbian paradigm, pre-post inducing LTP and post-pre LTD [176, 283]. Further studies revealed that GABAergic interneurons play an important role in deciding between Hebbian and anti-Hebbian behaviour, with normal function being anti-Hebbian and blockage of the GABA signal leading to Hebbian behaviour [264, 284].

More recently it has been shown that STDP is regulated by eCB and NMDAR, which presents a complex behaviour relating the amount of parings and their timings to either LTP or LTD. Pre-post STDP with more than 50 parings leads to eCB-mediated LTD, while post-pre, on the other hand, being more sensitive to the number of parings: a small number of parings (10-15) leads to eCB-mediated LTP, a large number of parings (greater than 75) leads to NMDAR-dependent LTP, and the intermediate number of parings produces no response [285–289].

Dopamine has been shown to be necessary for striatal Hebbian plasticity in vitro [176, 186, 283] and in vivo STDP [290, 291]. Xu *et al.* demonstrated that dopamine acting through D2R expressed in cortical pyramidal cells is necessary for STDP-eCB-LTP [288].

STDP in the striatum has been shown to make striatal neurons able to learn spike sequences and thus distinguish between different spiking sequences [292].

1.3.3 Intracellular pathways important for striatal plasticity

The intricate molecular mechanisms underlying striatal plasticity involve a complex interplay of numerous intracellular signalling pathways. These pathways, which include receptor-mediated cascades, second messenger systems, and protein kinase/phosphatase networks, collectively orchestrate the dynamic changes in synaptic strength that underpin learning and

memory processes in the striatum. This subsection focuses on the key molecular players and their interactions, providing a comprehensive overview of the biochemical basis for striatal plasticity. By examining these pathways in detail, we can better understand how various neurotransmitter systems converge to modulate synaptic function and ultimately influence striatal-dependent behaviours.

1.3.3.1 G_q-coupled receptor cascades

The mGluR and M1R signalling cascades are integral to striatal plasticity, primarily through their coupling to G_q proteins and subsequent activation of downstream effectors. These cascades exemplify the intricate molecular mechanisms underlying synaptic modulation in the striatum.

mGluR and M1R activation mechanisms

Glutamate (Glu), released from pre-synaptic vesicles [108, 109], activates multiple receptor types in striatal neurons. These include mGluR1 [113, 116], AMPAR, and NMDAR [107, 110]. Of particular significance to this cascade is the activation of mGluR1, which stimulates G_q [113, 116, 117], thereby initiating a signalling cascade that profoundly influences striatal plasticity.

Concurrently, M1R, activated by acetylcholine [214], also couples to and activates G_q [215]. Notably, M1R activation inhibits $Cav_{1.3}$ channels, and brief pauses in acetylcholine release can lead to disinhibition of these channels, resulting in elevated intracellular Ca^{2+} levels [97]. This mechanism establishes a critical link between cholinergic and glutamatergic signalling, and Ca^{2+}-dependent plasticity in striatal neurons.

G_q cascade

The activation of G_q, whether through mGluR1 or M1R, leads to the stimulation of PLC [215]. Activated PLC then catalyses the hydrolysis of phosphatidylinositol 4,5-bisphosphate (PIP_2) into two important secondary messengers: DAG [216] and IP_3 [217].

IP_3 plays a pivotal role in Ca^{2+} signalling by binding to inositol trisphosphate receptor (IP3R) on the endoplasmic reticulum, triggering the release of Ca^{2+} from intracellular stores [217]. This IP_3-mediated Ca^{2+} release significantly contributes to the complex Ca^{2+} dynamics underlying various forms of striatal plasticity.

DAG, another product of PIP_2 hydrolysis, is further metabolised by DAG lipase (DGL) to produce 2-AG [216, 218]. 2-AG serves as a crucial eCB in striatal signalling, potentially

influencing synaptic plasticity through retrograde signalling mechanisms.

Additionally, the G_q cascade and increased intracellular Ca^{2+} levels activate phospholipase D (PLD) [216]. Activated PLD can produce AEA [216, 218], another eCB that contributes to the modulation of striatal plasticity.

PIP_2 cycle

The replenishment of PIP_2 occurs through a series of enzymatic reactions. IP_3 is converted to inositol (Ino) [219], while DAG is transformed into phosphatidic acid (PA) [220]. PA is then converted to cytidine diphosphate-diacylglycerol (CDPDAG) [219]. CDPDAG and Ino are combined by phosphatidylinositol synthase (PIS) to form phosphatidylinositol (PI) [221].

The phosphorylation cascade continues with PI4-kinases converting PI to phosphatidylinositol 4-phosphate (PIP), followed by PIP5-kinases phosphorylating PIP to produce PIP_2. This process is bidirectional, with 4- and 5-phosphatases mediating the reverse reactions, respectively [219]. This cyclical interconversion of phosphoinositides is crucial for maintaining appropriate levels of PIP_2, thus regulating numerous cellular processes including ion channel function and synaptic plasticity.

M current

PIP_2 plays a crucial role in activating potassium voltage-gated 7 channel (Kv7.2/3) (encoded by the potassium channel NQ 2 or 3 (KCNQ2/3) gene) channels [222, 223]. Its reduction upon G_q activation leads to a decrease in M-type K^+ current [224], thereby enhancing neuronal excitability [225, 226]. This mechanism underscores the complex interplay between lipid signalling and ion channel function in regulating striatal neuron activity.

1.3.3.2 Endocannabinoid signalling

eCB signalling plays a crucial role in modulating striatal plasticity through multiple mechanisms. Two key eCBs, 2-AG and AEA, act on distinct targets to influence synaptic transmission and plasticity.

2-AG primarily exerts its effects by activating CB1 receptors [187]. These receptors are predominantly located on pre-synaptic terminals, allowing 2-AG to act as a retrograde messenger in striatal synapses. In contrast, AEA targets TRPV1 channels [188], providing an additional mechanism for eCB-mediated modulation of striatal function.

The impact of eCB signalling on synaptic plasticity exhibits a complex, concentration-dependent bi-directionality. Activation of CB1 receptors leads to a reduction in both PKA

and PP2B activity, primarily through attenuation of Ca^{2+} influx. The relative balance between PP2B and PKA activity, as represented by their ratio, determines the direction of plasticity. Ratios greater than 1 favour the diminished release of glutamate from vesicles, while ratios less than 1 promote this release, despite an overall reduction in the activity of both enzymes [227]. This nuanced regulation allows for fine-tuning of synaptic strength based on the precise levels of eCB signalling.

In addition to CB1-mediated effects, TRPV1 channels contribute to the regulation of glutamate release in striatal synapses [193, 194]. This dual action of eCBs on both CB1 and TRPV1 provides a multifaceted mechanism for modulating neurotransmitter release and synaptic plasticity in the striatum.

1.3.3.3 cAMP dynamics

In the context of striatal plasticity, the regulation of cAMP production serves as a critical modulator of intracellular signalling pathways. The D1R, predominantly expressed in direct pathway neurons, stimulates cAMP formation through its coupling with G_{olf} and subsequent activation of AC [228–230]. This increase in cAMP levels is associated with enhanced excitability and LTP in these neurons. Conversely, D2R, primarily found in indirect pathway neurons, exerts an inhibitory effect on cAMP production [228, 229], potentially contributing to synaptic depression. The M4R, which couples to G_i or G_o [214], similarly mediates the inhibition of AC, resulting in decreased cAMP levels [231] and modulating cholinergic influence on striatal plasticity. In contrast, A2AR activation leads to increased cAMP concentration through a similar mechanism to D1R through a G_{olf}-AC mechanism [230], particularly in indirect pathway neurons where it can counteract D2R-mediated effects. These diverse regulatory mechanisms underscore the complexity of cAMP-mediated signalling in striatal neurons and its crucial role in shaping synaptic plasticity and neuronal excitability.

1.3.3.4 Ca^{2+}-dependent signalling

Intracellular Ca^{2+} dynamics play a fundamental role in striatal plasticity, serving as a key regulator of various signalling cascades. The concentration of Ca^{2+} within striatal neurons determines the activation of specific pathways: low Ca^{2+} levels preferentially activate PP2B and protein-phosphatase 2A (PP2A), while high Ca^{2+} levels favour the activation of rat sarcoma virus (RAS) and calcium-calmodulin-modulated protein (CaM) [179]. This differential activation based on Ca^{2+} levels contributes to the bidirectional nature of synaptic plasticity in the striatum, influencing both long-term potentiation and depression.

RAS/RAF/MEK/ERK cascade

In the context of striatal plasticity, Ca^{2+}-mediated activation of RAS initiates a signalling cascade that culminates in the activation of extracellular signal-regulated kinase (ERK) via the RAS/RAF/MEK/ERK pathway [232]. This pathway is crucial for the induction and maintenance of long-term potentiation in striatal neurons. The process is further modulated by striatal-enriched protein tyrosine phosphatase (STEP), which is subject to phosphorylation by PKA [233, 234] and dephosphorylation by PP1 [235, 236]. In its non-phosphorylated state, STEP acts as an inhibitor of ERK activation [236, 237], thereby providing an additional layer of regulation to this plasticity-related signalling pathway.

CaM and CaMKII

The CaMKII pathway is another key mediator of striatal plasticity. The activation of CaMKII is initiated by its interaction with Ca^{2+}-bound CaM [238], which triggers a process of autophosphorylation [239]. This activated state of CaMKII is reversibly regulated through dephosphorylation by PP1 [239], allowing for dynamic control of its activity. One of the key downstream targets of activated CaMKII in striatal neurons is the AMPAR, which undergoes phosphorylation leading to its activation [240]. This CaMKII-mediated regulation of AMPAR represents an important mechanism in striatal synaptic plasticity, influencing both the strength and efficacy of glutamatergic transmission.

1.3.3.5 Phosphatase-mediated control

Protein phosphatases play crucial roles in regulating striatal plasticity by modulating the phosphorylation states of various key proteins involved in synaptic transmission and intracellular signalling cascades.

PP2A

PP2A is a critical regulator of striatal plasticity, activated by Ca^{2+} [241, 242]. In the context of striatal signalling, PP2A dephosphorylates two important regulatory proteins: cAMP-regulated phosphoprotein 21 kDa (usually ARPP-21) (A21) [243] and D32 [244, 245]. This dephosphorylation modulates their activity and, consequently, their influence on striatal plasticity. Interestingly, PP2A itself is subject to regulation through phosphorylation by PKA [246], creating a complex interplay between kinases and phosphatases in striatal neurons.

PP2B

PP2B, also known as calcineurin, is another key phosphatase in striatal plasticity mechanisms. Its activation is Ca^{2+}-dependent [247], specifically through interaction with CaM [248]. In striatal neurons, PP2B dephosphorylates and activates casein kinase 1 (CK1), as well as dephosphorylating D32 [249]. This creates a complex regulatory network, as the activated CK1 subsequently phosphorylates D32 and activates cyclin-dependent kinase 5 (Cdk5) [250], which in turn also phosphorylates D32 [251]. These intricate phosphorylation-dephosphorylation cycles are crucial for fine-tuning synaptic plasticity in the striatum.

PP1

PP1 is a major phosphatase involved in synaptic plasticity in the striatum. It dephosphorylates several key proteins involved in synaptic transmission and plasticity, including AMPAR, NMDAR, $Cav_{1.3}$ [184, 252], CaMKII [239], and STEP [235, 236]. The activity of PP1 is tightly regulated in striatal neurons, notably through inhibition by D32. This inhibition is enhanced when D32 is phosphorylated at threonine (Thr)-34 [244], providing a mechanism for dopamine-mediated modulation of synaptic plasticity through the regulation of PP1 activity.

1.3.3.6 Kinase-mediated regulation

Protein kinases play crucial roles in striatal plasticity by phosphorylating various target proteins, thereby modulating their activity and influencing synaptic transmission and intracellular signalling cascades.

PKC

PKC is a key player in striatal plasticity mechanisms. It is activated by DAG [253, 254], linking it to various receptor-mediated signalling pathways. In striatal neurons, PKC phosphorylates important synaptic proteins, including the AMPAR [255] and NMDAR [252]. These phosphorylation events can alter receptor properties and trafficking, thereby influencing synaptic strength and plasticity.

PKA

PKA is a central mediator of striatal plasticity, particularly in response to dopaminergic signalling. PKA is activated by cAMP [175, 182–184], linking it directly to D1R and A2AR ac-

tivation. PKA phosphorylates a wide array of substrates critical for striatal function, including AMPAR [255], NMDAR [252], $Cav_{1.3}$ [184, 252], D32 [244], A21 [243], regulator of G-protein signalling 4 (RGS4) [256], and STEP [233, 234]. These phosphorylation events can modulate receptor function, ion channel properties, and the activity of other signalling proteins, thereby influencing various aspects of striatal plasticity. Interestingly, PKA activity is subject to negative regulation by D32 [251], creating a feedback loop in the signalling cascade.

PKG

Protein kinase G (PKG) represents another important kinase in striatal plasticity mechanisms. PKG is activated by the NO/sGC/cGMP pathway [257], linking it to nitric oxide signalling in striatal neurons. A key substrate of PKG in the striatum is D32 [257, 258]. By phosphorylating D32, PKG can indirectly influence the activity of various other signalling molecules and effector proteins, contributing to the complex regulation of striatal plasticity.

1.3.3.7 Other integrator proteins

Beyond the major kinases and phosphatases, several other proteins play crucial roles in modulating striatal plasticity through complex phosphorylation-dependent mechanisms.

D32

D32 serves as a key integrator of various signalling pathways in striatal neurons, with its function finely tuned by phosphorylation at multiple residues. Phosphorylation at Thr-34 is mediated by PKA [244] and PKG [257, 258]. This phosphorylation enhances D32's inhibition of PP1 [244], while dephosphorylation is primarily carried out by PP2B [244, 245, 259]. At Thr-75, D32 is phosphorylated by Cdk5 [251] and dephosphorylated by PP2A [244, 245], with phosphorylation at this site reducing D32's inhibition of PKA [251]. Furthermore, phosphorylation at serine (Ser)-137 by CK1 [249] inhibits the dephosphorylation of Thr-34 by PP2B [249], adding another layer of regulation. This intricate phosphorylation pattern allows D32 to act as a sophisticated signal integrator in striatal plasticity.

A21

A21 represents another important regulatory protein in striatal signalling. It undergoes phosphorylation at Ser-55 by PKA and dephosphorylation by PP2A [243, 260]. When phosphorylated, A21 can bind to CaM, acting as a competitive inhibitor for other CaM-binding

proteins and thereby modulating downstream Ca^{2+} signalling effects [261]. This mechanism provides an additional means of regulating Ca^{2+}-dependent processes in striatal plasticity.

RGS4

RGS4 also plays a role in modulating G protein-coupled receptor signalling in striatal neurons. It is phosphorylated and activated by PKA [256], and functions to inhibit signalling in the G_q pathway [262]. This regulation of G protein signalling adds another dimension to the control of striatal plasticity, influencing the response to various neurotransmitters and neuromodulators.

1.3.3.8 Synaptic effectors

The culmination of various signalling cascades in striatal plasticity ultimately converges on several key effectors, primarily involved in glutamatergic transmission. These effectors, including pre-synaptic release machinery and post-synaptic receptors, integrate multiple regulatory inputs to fine-tune synaptic strength and plasticity.

Glutamate vesicles

The release of glutamate from pre-synaptic terminals is subject to retrograde regulation by eCBs. Differences in the activation of CB1 can lead either to the stimulation of release mediated by the activity of PP2B, or to inhibition through PKA activity [227]. Additionally, TRPV1 channels also regulate glutamate release [193, 194].

AMPAR

AMPAR, a key mediator of fast excitatory transmission in striatal neurons, is subject to both ligand-gated activation and phosphorylation-dependent regulation. Activated by Glu [107, 110], AMPAR function is modulated through phosphorylation by several kinases, including PKC [255], PKA [255], and CaMKII [240]. Conversely, PP1 acts to dephosphorylate AMPAR [184, 252]. This intricate balance of phosphorylation and dephosphorylation allows for precise control of AMPAR activity and trafficking to and from the membrane, directly influencing synaptic strength.

NMDAR

Similar to AMPAR, NMDAR plays a crucial role in striatal plasticity and is subject to both ligand-gated activation and phosphorylation-dependent regulation. Activated by Glu [107, 110], NMDAR function is modulated through phosphorylation by PKC and PKA [252]. PP1 counteracts this phosphorylation through its dephosphorylating activity [184, 252]. The regulation of NMDAR function is particularly important given its role in Ca^{2+} influx and subsequent activation of Ca^{2+}-dependent signalling cascades that are critical for various forms of striatal plasticity.

1.4 Modelling in neuroscience

Computational models in neuroscience span multiple scales, from molecular interactions to whole-brain dynamics, providing critical insights into neural systems [12–14]. These models enable researchers to examine biological processes across different organizational levels, elucidating how cellular mechanisms contribute to overall brain function. A prominent application is the study of synaptic plasticity, fundamental to learning and memory. Models range from detailed kinetic simulations of long-term potentiation and depression to network-level analyses exploring plasticity's impact on neural circuitry.

The methodologies employed include ordinary differential equation (ODE)s for biochemical reactions, stochastic simulations for small-volume random effects, and spatial modelling for neuronal morphology. Integration of these methods with advanced live-cell imaging has revealed emergent properties of signalling networks, such as differential responses to stimuli and activity maintenance through feedback loops [15]. This computational-experimental synergy allows neuroscientists to generate hypotheses, guide experiments, and interpret complex data, crucial for unravelling neural function and dysfunction. The following sections will explore specific modelling approaches and their applications in understanding neuronal processes.

1.4.1 Multicellular and network models

The complex functionality of the nervous system emerges from the coordinated activity of large populations of interconnected neurons. To elucidate these intricate dynamics, researchers have developed multicellular and network models that investigate the collective behaviour of neuronal ensembles and their interactions [293, 294]. This section will discuss three primary approaches to modelling neuronal populations: population/mean-field models,

spiking neural networks, and connectome-based models, each offering distinct insights into the emergent properties of brain networks at different scales and levels of abstraction.

1.4.1.1 Population models

Population models encapsulate the collective dynamics of neuronal ensembles, representing them through mathematical frameworks such as differential equations. These models are invaluable for examining the overall behaviour of neural circuits, particularly in relation to brain rhythms and large-scale dynamics. One pivotal model is the Wilson-Cowan model, which formulates the interactions between excitatory and inhibitory neuronal populations. This model has been instrumental in elucidating the emergence of oscillatory behaviour within neural networks and provides insights into the regulatory mechanisms of brain states and transitions [295–297].

Extensions of the Wilson-Cowan framework have been adapted to explore complex brain phenomena, such as the dynamics of neural oscillations and the influence of neuromodulators like dopamine on network activity [298–300]. These models offer a profound understanding of how brain-wide interactions contribute to cognitive functions and pathological states [301, 302].

Furthermore, mean-field approximations are applied specifically to the striatum to model its intricate dynamics, demonstrating how this region responds to various brain rhythms and the modulatory effects of dopamine [300]. This approach allows researchers to predict and analyse the collective output of large neuronal populations, offering a macroscopic view of brain activity essential for both theoretical and applied neuroscience.

1.4.1.2 Spiking neural networks

Spiking neural network models, instrumental in simulating the dynamics of interconnected neurons, primarily emphasize the temporal characteristics of action potentials. These models integrate diverse synaptic models and plasticity rules, facilitating simulations through both event-driven and time-driven methodologies [303–305].

Spiking neural networks are pivotal for elucidating neural coding and computational theories, as they effectively capture the temporal intricacies of neuronal activities [306, 307]. Incorporating biologically pertinent plasticity rules, such as STDP [308], these networks extend our understanding of the underlying mechanisms of learning and memory within the brain [309, 310]. The advent of neuromorphic hardware has further propelled the capabilities to implement large-scale spiking neural networks, thus opening new avenues in the

emulation of brain-like computations [311, 312].

1.4.1.3 Connectome-based models

Connectome-based models represent an advanced computational methodology that integrates detailed anatomical and functional information about neuronal connections to simulate brain-wide activity patterns and information flow [12, 293, 313, 314]. By combining structural and functional connectivity data, these models aim to provide a comprehensive understanding of brain disorders and cognitive processes [315, 316]. Recent advancements in this field include the development of software platforms such as Snudda, which enable the creation and simulation of large-scale networks of detailed microcircuits, incorporating neuromodulation at the single-cell level [317].

In recent years, significant progress has been made in applying connectome-based models to subcortical structures, particularly the striatum. A near full-scale computational model of the mouse striatum has been developed, incorporating detailed synaptic connectivity, cellular morphology, and electrophysiological properties [101]. This model simulates approximately 10,000 striatal neurons, including SPNs and various interneurons, with inputs from the cortex, thalamus, and the dopaminergic system. It provides insights into how surround inhibition from striatal projection neurons shapes dendritic computation, network dynamics, and competition between neuronal sub-populations [318].

Furthermore, open-source computational tools such as Neuromodcell and Snudda have enhanced the capability to model complex neuromodulatory interactions in large-scale simulations [319]. These tools enable the simulation of dopaminergic and muscarinic modulation of striatal neurons under different neuromodulatory states, facilitating the investigation of pathological conditions such as Parkinson's disease, which is associated with dopaminergic deficits in the basal ganglia [38, 72, 73, 320].

Despite these advancements, scaling up connectome-based models to encompass the entire brain presents significant challenges in terms of computational resources and model complexity [321, 322]. Ongoing efforts to optimize computational efficiency and develop novel simulation techniques aim to address these challenges, with initiatives such as the Human Brain Project and EBRAINS working towards the creation of large-scale, biologically realistic brain models [323, 324].

The continued development of connectome-based models and associated computational tools underscores their potential in bridging scales between experimental data, bottom-up predictions, and top-down modelling, particularly in understanding subcortical nuclei like the

basal ganglia [317]. These approaches provide a powerful framework for investigating complex neural dynamics and their implications for both healthy brain function and neurological disorders, thereby enhancing our understanding of brain function and disease mechanisms.

1.4.2 Cellular and electrical models

To bridge the gap between molecular processes and neuronal behaviour, cellular and electrical models simulate the biophysical properties of individual neurons. These models are fundamental tools in neuroscience, enabling the exploration of how neurons process information and contribute to network dynamics [325, 326].

1.4.2.1 Single-compartment models

Single-compartment models, such as the Hodgkin-Huxley model and its various simplifications [327], represent neurons as electrically uniform units without spatial differentiation. These models are computationally efficient and are invaluable for studying basic neuronal properties, including action potential generation and excitability [328, 329].

Within this category, integrate-and-fire models and their numerous variants are particularly popular. They capture the essential features of neuronal excitability while maintaining computational simplicity [330–332]. Such models have been extensively used to investigate the dynamics of large-scale neuronal networks, shedding light on mechanisms of synchronization and information processing in the brain [333, 334].

1.4.2.2 Multi-compartment models

Multi-compartment models, on the other hand, divide neurons into multiple interconnected segments. This approach allows for a detailed study of spatial aspects of neuronal computation, including dendritic integration and action potential back-propagation [335, 336]. Grounded in cable theory, which treats neurons as electrical wires, these models are crucial for understanding how dendritic structures contribute to neuronal processing [337, 338].

The increased complexity of multi-compartment models enables researchers to investigate the effects of dendritic morphology and the spatial distribution of synaptic inputs on neuronal responses [339, 340]. However, this added detail comes with greater computational demands, necessitating a careful balance between biological realism and computational efficiency [304, 341].

In the striatum, multi-compartment models have been instrumental in exploring the computational capabilities of individual SPNs. By incorporating dendritic non-linearities and local

synaptic learning rules, these models demonstrate that single SPNs can perform complex computations, such as solving non-linear feature-binding problems – tasks traditionally attributed to neural networks. They also highlight the roles of dendritic plateau potentials, calcium dynamics, and inhibitory plasticity in enhancing the computational efficiency of SPNs [342].

These models have further elucidated cell-type-specific mechanisms of GABAergic inhibition that precisely regulate the amplitude, kinetics, and duration of dendritic plateau potentials, thereby controlling the spiking output of SPNs [83]. Additionally, biophysically detailed models of striatal neurons have been employed to study the combined effects of dopamine and acetylcholine neuromodulation on dendritic integration. This interaction can induce complex spiking patterns similar to those observed in hippocampal neurons during behavioural time-scale plasticity [343].

Overall, multi-compartment models provide a powerful framework for understanding how the spatial distribution of synaptic inputs and dendritic properties shape neuronal function. They offer valuable insights into the cellular and biophysical mechanisms underlying complex neuronal computations, thereby contributing significantly to our understanding of striatal function and basal ganglia circuitry.

1.4.3 Sub-cellular and biochemical models

At the sub-cellular level, biochemical models are instrumental in elucidating the molecular processes within neurons, including signalling cascades, metabolic pathways, and protein interactions [16, 17, 344, 345]. These models are crucial for understanding the fundamental mechanisms that underlie neuronal function and behaviour.

1.4.3.1 Reaction-diffusion models

Reaction-diffusion models describe how the concentration of substances changes over time and space due to chemical reactions and diffusion processes. They are particularly useful for investigating the spatiotemporal dynamics of intracellular signalling pathways, such as calcium dynamics and neurotransmitter diffusion within neurons [346].

The application of reaction-diffusion models to intracellular signalling has provided valuable insights into mechanisms of synaptic plasticity, neurotransmitter release, and the propagation of signalling molecules within neuronal structures [347]. However, challenges remain in parameter estimation and achieving adequate spatial resolution in these models. Recent advances have focused on employing sophisticated numerical methods and parallel com-

puting techniques to simulate large-scale reaction-diffusion systems more effectively [348].

1.4.3.2 Kinetic models

Kinetic models are mathematical frameworks that describe the rates of chemical reactions, commonly used to study molecular interactions such as receptor-ligand binding and enzyme activity [349]. These models often incorporate Michaelis-Menten kinetics and chemical kinetics equations, typically represented as systems of ODEs, and are frequently combined with signalling cascade models to provide a comprehensive understanding of intracellular processes [350]. By integrating kinetic models with experimental data, researchers can quantitatively analyse molecular dynamics and predict the effects of perturbations on cellular functions [351].

In the context of the striatum, kinetic models have been particularly useful in exploring intracellular signalling dynamics, including the integration of glutamatergic and dopaminergic inputs [179]. These models help to reveal the intricate interplay between neurotransmitters and their influence on the molecular and cellular properties of striatal neurons. For instance, mass-action kinetics models combined with live-cell imaging have demonstrated an ultra-sensitive responsiveness of nuclear PKA in striatal neurons, indicating a positive feed-forward mechanism that enhances the detection of transient dopamine signals during reward-mediated learning [352].

Furthermore, quantitative kinetic modelling has been employed to investigate the interactions between key neuromodulators – such as dopamine, acetylcholine, and adenosine in striatal medium spiny neurons, revealing complex cAMP/PKA signalling dynamics and underscoring the role of GPCR interactions in encoding reward prediction errors [353]. Computational models have also examined the roles of striatal phosphoproteins like D32 and A21 in modulating the integration of calcium ion and dopamine signals during reward learning, providing critical insights into the cellular mechanisms underlying synaptic plasticity [2].

The relatively low computational cost of mass-action models allows for extensive analyses, such as global sensitivity analysis, which can further elucidate the dynamics of neuromodulatory systems within the basal ganglia [354]. These modelling approaches offer a robust framework for exploring the complex biochemical signalling pathways that underpin neuronal function and behaviour.

1.4.3.3 Stochastic models

Stochastic models incorporate randomness to account for the inherent variability in biological systems, which is particularly important when dealing with small numbers of molecules or rare events [355, 356]. The Gillespie algorithm and its variants are commonly used in stochastic simulations to model the probabilistic nature of chemical reactions, providing insights into the role of noise in cellular processes [357, 358].

These stochastic modelling approaches have been applied to study the variability in gene expression, the dynamics of synaptic transmission, and the effects of molecular noise on neuronal excitability [359–361]. By capturing the probabilistic nature of biological processes, these models provide a framework for understanding the robustness and adaptability of neuronal systems in the presence of noise [361, 362]. In the context of the striatum, stochastic models have been used to investigate complex signalling pathways in neurons, such as those involved in synaptic plasticity. For example, they enable the simulation of molecular interactions within and between dendritic spines and can predict the occurrence of long-term potentiation or depression under various conditions [363].

1.4.4 Cross-scale integration

Understanding the brain's complex functions requires an approach that bridges multiple levels of biological organization, from molecular interactions to behavioural outcomes. Cross-scale integration in neuroscience modelling seeks to connect these different scales, providing a holistic perspective on neural systems and their emergent properties [12, 297, 364]. This integrative approach is essential for elucidating how cellular and molecular processes influence higher-order brain functions and vice-versa, offering insights that isolated single-scale models cannot achieve.

Multi-scale modelling combines molecular-level simulations with cellular and network models, enabling researchers to investigate how alterations at one level can affect behaviour at another [364, 365]. For instance, integrating models of synaptic plasticity with network simulations has facilitated the study of systemic mechanisms underlying learning and memory [309, 366]. Similarly, multi-scale simulations of adenylyl cyclase, incorporating quantum mechanics, molecular dynamics, and systems biology approaches, have revealed how molecular events propagate through cellular signalling pathways to influence neuronal behaviour [367].

In the context of the striatum and basal ganglia, multi-scale models play a pivotal role in linking cellular mechanisms to network dynamics. As previously discussed, the basal

ganglia are involved in motor control and reinforcement learning, functions that emerge from interactions across multiple scales. Striatal modelling often employs spiking network models that inform mean-field approximations, connecting the activity of individual neurons to large-scale dynamics involved in learning and motor control [300]. Studies of neuromodulation within the basal ganglia have integrated biochemical signalling cascades with multi-compartmental neuron models to explore how neurotransmitters like dopamine affect membrane conductances and cellular behaviour across different time scales [368]. These integrative models provide essential insights into the interactions between molecular signalling, cellular responses, and the broader network-level dynamics that underlie complex behaviours.

Despite the significant potential of cross-scale integration, several challenges persist, including the reconciliation of differing time scales, spatial resolutions, and computational demands [365, 369]. Co-simulation techniques have been developed to address these challenges, enabling interoperability between diverse modelling tools and distributed computing frameworks. For example, multi-rate algorithms optimize simulation efficiency by assigning appropriate time steps to different components within a model, enhancing both accuracy and computational performance when dealing with systems characterized by widely varying time constants [370].

Recent advancements in computational methods and modelling frameworks continue to refine multi-scale approaches, striving to provide a more integrated understanding of brain function across levels of biological organization [369, 371]. Additionally, developments in machine learning and data-driven modelling have opened new avenues for integrating diverse data sources across scales, facilitating the construction of more comprehensive and predictive models of brain function [372, 373]. These innovations hold promise for enhancing the fidelity of cross-scale models, bridging the gap between detailed cellular mechanisms and observable behaviour, and contributing to our understanding of complex neurological processes.

1.5 FAIR Principles in Computational Neuroscience

The exponential growth of data and increasing complexity of models in computational neuroscience necessitate robust data management and sharing practices. The FAIR principles, introduced by Wilkinson *et al*. [374], provide a framework to enhance data management, promote transparency, and facilitate collaboration. These are especially important for this thesis since it focuses on developing computational tools for analysing biochemical

pathway models that will be used by other researchers in the field. By adhering to these principles, this work not only advances scientific knowledge but also contributes to the broader neuroscience community by making data and models more discoverable, accessible, interoperable, and reusable. This ensures that the computational tools and models developed in this work, or other models developed using these tools in the future, can be effectively integrated into existing workflows, reproduced by other researchers, and applied in future studies, thereby maximizing their potential impact and utility within the field.

1.5.1 Findability

Findability is crucial in computational neuroscience due to the vast and complex nature of datasets and models in the field. Enhancing the discoverability of these resources by both humans and machines is essential for advancing research. To this end, neuroscience-specific metadata standards have been developed, such as the Brain Imaging Data Structure (BIDS) [375], which provides a standardized approach to organizing and describing neuroimaging data, thereby facilitating data sharing and searchability across various platforms.

In the context of biochemical pathway models, findability is enhanced through the use of detailed metadata descriptions encompassing species, reactions, parameters, and experimental conditions. The adoption of standardized formats like Systems Biology Markup Language (SBML) [376] and Systems Biology table format (SBtab) [377] ensures consistency and richness in metadata, significantly aiding model discovery. Public repositories such as ModelDB [378] and the BioModels Database [379] further contribute to findability by assigning unique identifiers to models and providing search functionalities based on specific criteria.

1.5.2 Accessibility

Accessibility refers to the ease with which data and metadata can be retrieved once discovered. In the field of computational neuroscience, ensuring data accessibility involves the adoption of standardized protocols and platforms that facilitate efficient data retrieval. Several data-sharing platforms have been developed to address this need, including OpenNeuro [380], NeuroMorpho.Org [381], and the Allen Brain Atlas [382]. These platforms implement robust accessibility features by providing open access to datasets through standardized web interfaces and Application Programming Interfaces (APIs).

The utilization of widely used and open-source software and formats significantly en-

hances accessibility. For instance, MATLAB, although not open-source, is a commonly used tool in computational neuroscience, which supports various data formats, such as MAT-files, and provides functionalities for importing and exporting SBML and other standardized formats [383]. By employing widely supported data formats and ensuring that the software used to access the data is either open-source or commonly used within the field, researchers can substantially improve the accessibility of their data.

While accessibility is paramount, it is also crucial to implement appropriate authentication and authorization mechanisms to protect sensitive data. Many platforms address this by providing tiered access levels, allowing public access to non-sensitive data while maintaining controlled access to data requiring permissions. The Human Connectome Project exemplifies this approach by providing open access to anonymized datasets while safeguarding participant confidentiality [384].

1.5.3 Interoperability

Interoperability emphasizes the capacity for data and tools from diverse sources to function cohesively. This is particularly crucial in computational neuroscience due to its multidisciplinary nature, which necessitates the integration of data from various experimental and computational methodologies.

The adoption of standardized data formats plays a pivotal role in enhancing interoperability. Formats such as SBML [376], NeuroML [385], and SBtab [377] provide common languages for model representation, facilitating seamless data exchange and integration.

To bridge the gap between different data formats and modelling platforms, the development of conversion tools is essential. For instance, during this project, tools were created to translate SBtab files into MATLAB code, SBML files, or NEURON simulation files [1]. Such tools significantly enhance the ability of researchers to collaborate effectively and integrate diverse datasets across different platforms and formats.

The representation of models as standardized systems of ODEs provides a common mathematical framework in computational neuroscience, further promoting interoperability. This standardization facilitates the integration of models with other datasets and computational tools, enhancing the overall coherence of research efforts in the field.

Additionally, the use of web services and APIs significantly contributes to interoperability by enabling communication between different software tools and platforms. An example of this approach is the Open Source Brain platform, which provides collaborative tools and model repositories supporting multiple modelling formats and simulation environments [386].

Such platforms play a crucial role in fostering a more interconnected and collaborative research ecosystem in computational neuroscience.

1.5.4 Reusability

Reusability emphasizes the importance of ensuring that data and models can be effectively utilized in future research endeavours. This principle necessitates that data is well-documented, accompanied by comprehensive metadata, and shared under appropriate licensing terms.

In the field of computational neuroscience, reusable datasets have played a pivotal role in advancing research [387]. A prime example is the Human Connectome Project [384], which provides open access to high-quality neuroimaging data, enabling researchers worldwide to conduct analyses without the need for primary data collection. Similarly, repositories such as ModelDB [378] offer access to a diverse range of computational models that can be adapted and extended for novel studies, significantly accelerating the pace of research.

The implementation of clear licensing is crucial for ensuring reusability. Data should be shared under licenses that explicitly specify the terms of use, such as Creative Commons licenses. This clarity in licensing terms facilitates the appropriate use and attribution of data in subsequent research. Furthermore, detailed documentation, encompassing methodological details, parameters, and code, is essential for enabling others to comprehend and build upon existing work [388, 389].

Version control systems, such as Git, and its implementations such as GitHub, play a vital role in enhancing reusability by allowing for the tracking of changes and facilitating collaborative development. This approach is particularly important for maintaining the integrity of models and ensuring that improvements or modifications are meticulously documented. By providing a transparent history of model development, version control systems contribute significantly to the long-term reusability and reproducibility of computational neuroscience research [390, 391].

1.5.5 Importance and implementation of FAIR in neuroscience research

The adoption of FAIR principles in neuroscience research is crucial for addressing several key challenges in the field [392]. Primarily, these principles help mitigate the reproducibility crisis by ensuring transparency and accessibility of data and methods [393]. By enhancing the findability and accessibility of datasets, researchers enable the replication of experiments, validation of findings, and extension of previous work, thereby fostering a more

robust scientific process.

Furthermore, FAIR principles facilitate interdisciplinary collaboration, which is essential in the multifaceted field of neuroscience. By promoting interoperable data practices, these principles enable the integration of diverse datasets from various subdisciplines, including biology, psychology, physics, and computer science. This integration allows for more comprehensive analyses and encourages innovative research approaches [394].

Practical implementation of FAIR principles in neuroscience involves the use of standardized data formats and metadata schemas. For instance, the BIDS standardizes the organization and description of neuroimaging data, significantly enhancing interoperability and reusability [375]. Similarly, the Neurodata Without Borders (NWB) initiative aims to establish a unified data format for cellular neurophysiology data [395], further promoting standardization in the field.

Open data repositories such as OpenNeuro and the Allen Brain Atlas exemplify the application of FAIR principles by providing platforms for data sharing that adhere to these guidelines [380, 382]. These resources not only make data available but also offer tools and documentation to enhance usability, thereby promoting both accessibility and reusability.

The increasing recognition of the importance of FAIR principles is reflected in the policies of funding agencies and journals, which are increasingly mandating data sharing practices aligned with these principles. By embracing FAIR practices, the neuroscience community can significantly improve data management, enhance reproducibility, and accelerate scientific progress, ultimately advancing our understanding of the brain and nervous system.

1.5.6 FAIR principles in the context of this thesis

The work presented in this thesis adhered to the FAIR principles throughout its development. The implementation of these principles enhances the reproducibility, accessibility, and potential impact of the research presented. In the subsequent sections, specific examples will be provided to illustrate how each of the FAIR principles has been integrated into the methodology and outcomes of this work, demonstrating their practical application in advancing the field of computational neuroscience.

1.5.6.1 Findability

The models developed in this thesis—Ligand-Receptor Model (LiRMo), Ligand-Receptor-Effector Model (LiREM), Timing Model-Modified (TM-M), and the GPCR and G-protein (G), PLC (P), PI/PIP/PIP2 (PIP), IP3 (IP), and Kv7.2/3 with its associated current (K) (G-PIPK)

Model—adhere to the principle of findability through several key implementations. Each model is assigned a unique identifier within dedicated GitHub repositories, ensuring persistent and globally unique references. Specifically, TM-M and the G-PIPK Model are stored in individual repositories, while LiRMo and LiREM are housed in a subdirectory of the main software repository. These repositories utilize GitHub's robust versioning and tagging mechanisms, enhancing traceability and identification [8, 10, 11].

To enhance the findability of these models, rich metadata is provided using the SBtab format [377]. This standardized format includes comprehensive information on species, reactions, parameters, and experimental conditions. For example, each species within a model is annotated with its chemical identity, functional role in the pathway, and initial concentrations. Similarly, reactions are described with their kinetic equations, rate constants, and relevant literature references.

The implementation of this detailed metadata significantly enhances the searchability and discoverability of the models. Researchers can efficiently search the repositories using keywords related to specific species, reactions, or experimental conditions. Moreover, the structured nature of the SBtab format facilitates indexing by search engines and metadata harvesters, further improving the findability of these computational resources in the broader scientific community.

1.5.6.2 Accessibility

The principle of accessibility is rigorously implemented in this thesis through the utilization of widely adopted software platforms and open-access practices. MATLAB, a prevalent tool in computational neuroscience, serves as the primary platform for simulations and analyses [396]. The choice of MATLAB not only leverages its extensive capabilities but also ensures long-term accessibility of the developed computational tools, supported by its established user community and ongoing support.

To maximize accessibility, all software, models, and data used to constrain the models, developed or created, are made publicly available on GitHub under open-source licenses, specifically the GNU General Public License version 3 (GPLv3) [3, 5, 7–11]. This approach removes barriers to access, encourages collaboration, and enables other researchers to freely use, modify, and build upon the developed tools and models.

The adoption of the SBtab format for model specifications further enhances accessibility [377]. This human-readable and machine-parsable tab-delimited text format facilitates easy sharing and interpretation of model information and accompanying data without requiring

proprietary software. The use of such open formats ensures that data remains accessible and usable even as software technologies evolve over time.

To further support accessibility, comprehensive documentation is provided alongside the models and software. This documentation in conjunction with this thesis includes detailed explanations of the code structure, descriptions of the model files used, and guidance on utilizing MATLAB and other resources [397]. By offering this level of documentation, the work becomes accessible to users with varying levels of expertise, thereby broadening its potential impact and utility within the research community.

1.5.6.3 Interoperability

Interoperability was a key focus from the start of the project presented in this thesis. This thesis presents tools that convert SBtab into MATLAB models. Parallel work not discussed here was also done to create tools that convert SBtab into SBML files, vector field (.vf) files, and NEURON (.mod) files [1, 8, 9]. This multi-format approach significantly enhances the interoperability of the developed models.

The conversion capabilities enable the utilization of these models across diverse simulation platforms. For instance, the models can be seamlessly integrated into NEURON for detailed neuronal simulations [305] or COPASI for an open source alternative to MATLAB in biochemical network analysis [398]. By supporting these standardized formats and languages, the models can effectively interact with a wide array of tools and datasets, thereby expanding their utility and applicability in the field.

Moreover, the representation of models as standardized systems of ODEs further enhances interoperability. This approach employs a common mathematical framework widely used in computational neuroscience, facilitating the integration of these models with other existing models and datasets. Such standardization not only promotes consistency across different studies but also enables researchers to more easily compare, combine, and build upon these models in future work.

1.5.6.4 Reusability

Reusability was also taken into consideration on the elaboration of this thesis, implemented through several key strategies. The models are structured in a modular fashion, with distinct components defined in the SBtab format. This modular approach enables researchers to easily extract and re-purpose individual components, such as a specific number of reactions from a model or only specific datasets, for use in their own work. Such modu-

larity significantly enhances the versatility and reusability of the developed models.

To facilitate reuse, comprehensive documentation accompanies all models and software. This includes detailed user manuals and extensive code comments, providing clear insights into the underlying methods, assumptions, and implementation details [397]. By offering this level of documentation, the work becomes more accessible and easier to replicate, modify, or extend, thus promoting its reuse in future research endeavours.

Version control is rigorously maintained through GitHub, allowing for precise tracking of changes over time and facilitating collaborative development [8, 9]. This approach not only ensures transparency in the development process but also enables other researchers to fork the repositories, contribute improvements, or report issues, thereby enhancing the collective value and longevity of the models.

The adoption of the GPLv3 License for all code and models clearly specifies the terms of use, permitting modification and redistribution. This open licensing approach removes barriers to reuse and encourages broader adoption within the research community. Furthermore, the inclusion of detailed provenance information, such as data origins, parameter sources, and version histories, ensures that users can fully trace the development of the models, enhancing their reliability and reusability in diverse research contexts.

2 Aims

The primary objective of this thesis will be to develop a computational framework specifically tailored to the modelling and analysis of biochemical pathways within the field of neuroscience, specifically the basal ganglia. By bridging the gap between intricate biological systems and computational models, this work will aim to provide researchers with a rigorous, adaptable, and accessible platform that addresses key challenges in understanding neurobiological processes. The objectives outlined here will be intended to enhance both the usability and functionality of biochemical pathway modelling tools, making them more accessible to a broader spectrum of researchers while enabling the handling of diverse modelling demands with increased efficiency and accuracy.

A software platform will be developed in strict adherence to FAIR (Findable, Accessible, Interoperable, Reusable) principles, thereby ensuring transparency, reproducibility, and reusability in computational research. The software will be designed with an intuitive, user-centric interface and streamlined model import functionalities to reduce technical barriers, particularly for researchers with limited coding experience. The software's flexible architecture will be developed to accommodate various types of biochemical models based on mass action kinetics, ranging from simple ligand-receptor interactions to complex neurobiological signalling pathways, thereby enabling its adaptation to diverse research needs within neuroscience.

A comprehensive suite of analytical tools will be integrated into the software to facilitate robust analysis and validation of created ordinary differential equations models. Parameter estimation techniques will be employed to align models with experimental data accurately, thereby enhancing predictive capabilities. Sensitivity analysis methods, encompassing both local and global approaches, will be included to identify key parameters and reactions that significantly influence model behaviour, providing valuable insights into system stability and robustness. Additionally, profile likelihood analysis will be implemented to assess parameter identifiability, guiding researchers to determine the most influential elements of the model. Collectively, these tools will be developed to support precise model refinement and contribute to the reliability of resulting interpretations.

The flexibility and scalability of the platform will be demonstrated through applications to models of varying complexity. Core functionalities will first be validated using relatively simple models, such as LiRMo and LiREM, to ensure foundational reliability. Subsequently, intermediate models, such as TM-M, will be employed to evaluate the platform's ability to

manage larger parameter sets effectively. Finally, the platform's capacity to handle sophisticated, data-rich systems will be showcased using a complex model like G-PIPK. This progression will be designed to underscore the adaptability of the platform across different experimental contexts, highlighting its potential as a comprehensive tool for neurobiological modelling.

The platform's predictive capabilities will be utilized to facilitate hypothesis generation and support experimental design. Through detailed analysis of the G-PIPK model, key parameters, interactions, and pathways will be systematically identified, offering targeted insights to guide experimental validation efforts and prioritize experimental focus areas.

Finally, this thesis will establish an extensible workflow that will serve as a foundation for future advancements in computational neuroscience. By fostering cumulative research and facilitating model sharing, this workflow will provide a versatile and collaborative tool-kit, promoting integration and innovation within the computational neuroscience community.

In summary, the work presented in this thesis will aim to create a comprehensive, adaptable, and user-friendly platform designed to meet the evolving needs of biochemical pathway modelling in neuroscience. By enhancing accessibility, analytical power, and scalability, this thesis will provide a valuable tool-kit with the potential to significantly advance both fundamental research and practical applications in computational neuroscience, ultimately fostering a deeper understanding of complex neurobiological systems.

3 Methods

3.1 Biochemical pathway models

Historically, mathematical and computational models have been vital in dissecting the complexities of biological systems. The earliest models were largely phenomenological, drawing on observed patterns and behaviours without delving into underlying mechanisms [399]. However, as our comprehension of biology deepened, models started to encompass the biochemistry and biophysics that underpin these phenomena [327]. Presently, with leaps in computational power, we are capable of building and analysing sophisticated multi-scale models spanning from molecules to ecosystems, thereby providing a comprehensive understanding of biological systems [400]. This progression in modelling has set the stage for diverse scales of neuronal models, including network, neural masses, cellular, and sub-cellular models, with the primary focus here being the smallest scale, the biochemical pathway models.

Neuronal systems can be modelled at various levels of detail [12]. At the large scale, there is the modelling of entire brains, referred to as network models, where most neurons are normally modelled as single points, or as three dimensional neurons [303] [314]. Then, there are neural masses models [295] [296], in which excitatory and inhibitory neurons are modelled not as single neurons but as groups with average firing rates and dynamics interacting with one another. At the intermediate scale, the modelling of the interactions of small groups of neurons, referred to as cellular models, where the interactions are described as electrochemical processes modelled based on the Hodgkin and Huxley formalism [327]. And at the small scale, the modelling of intracellular processes and biochemical reactions usually occurring inside a single cell, referred to as sub-cellular models.

The models considered in the modelling paradigm presented in this dissertation, one of the possible at this smaller level, are referred to as biochemical pathway models [344]. They are built upon the principle that interactions of small molecules and proteins can be modelled using the rate law and law of mass action (described in section 3.3), and that the behaviour of the ordinary differential equations (ODEs) that are thus generated can approximate and predict experimentally obtained data for such small molecules and proteins.

3.1.1 Model components

All the models in this dissertation are constructed from the same basic building blocks: species (S), reaction (R), parameter (P), initial conditions, events, and outputs.

The species are all the physical substances that will be modelled. These can be chemical elements such as calcium, small molecules such as adenosine triphosphate (ATP) or glutamate, or proteins such as the metabotropic glutamate receptor (mGluR) or calmodulin.

The reactions describe how the species interact with one another. Each reaction is divided into the species existing before the reaction takes place (reactants), and those formed after the reaction is completed (products). As an example, the binding of glutamate to its receptor, mGluR, would be modelled by a reaction where the reactants would be $glutamate$ and $mGluR$ and the product $mGluR_glutamate$ (the agglutination indicates a complex), formally represented in equation (3.1). Reactions with only reactants or products are possible when species are continually introduced or removed from the model, for example there could be a model were ATP is continually introduced and adenosine diphosphate (ADP) continually removed, formally represented in equations (3.2) and (3.3) respectively. An explanation of how these reactions can be mathematically treated can be found in sections 3.3.1 and 3.3.2.

$$glutamate + mGluR \quad \rightleftharpoons \quad MGluR_glutamate \tag{3.1}$$

$$null \quad \longrightarrow \quad ATP \tag{3.2}$$

$$ADP \quad \longrightarrow \quad null \tag{3.3}$$

The parameters are all the unknown variables that govern the behaviour of the model, such as the undefined constants of the equations describing the reaction of the models, k, α, and β in equations (3.5) and (3.6).

The initial conditions are all the known variables used to initialize the models. In the models of this dissertation these are the species' initial concentrations. Note that in the case of an unknown starting point for a species' concentration, this concentration could be treated as another parameter and not an initial condition. This behaviour, however, is not present in the models presented in this dissertation and is not contemplated in the model analysis.

The events represent changes in the modelling system that occur while the simulation is ongoing and which are not accounted by the ODE system, for example the concentration of a species changing abruptly at a specific time point. These events are the simulation equivalent of experimental procedures executed by a lab technician during the recording of

data, such as the introduction of glutamate to a medium with neurons while measuring their firing properties.

The outputs of the model are all the variables that can be measured from the model either during or after the simulation. The outputs can be the concentration of the species during model simulation, or the result of a mathematical operation performed on this concentration, such as the ratio of the concentration of two species. These model outputs are usually mapped to some biological value. If the behaviour of such biological value is known, we can use it to tune the parameters of our model, so that it is able to reproduce it, through parameter estimation. With enough data from known biological values, the model can then be used to produce output variables analogous to unknown biological processes and be used to predict the behaviour of these systems.

3.2 Model Data

The information needed for the models in this dissertation can be divided into two important sub-categories: data for the model itself (referred from now on as "model structure") and the data that the model should reproduce (referred from now on as "model data"). The first subset of data has been regarded as the most important so far and, as such, there are various formats to store it (SBML [376], CellML [401–403], SBtab [377], MATLAB SimBiology model [404]) and repositories to easily share it (BioModels [379], ModelDB [378]). Unfortunately, the second subset has not been as standardized and, therefore, it is still cumbersome to find the correct model data and to subsequently share it. This probably stems from the fact that the model structure is effectively independent from external factors. A model structure can exist independently and, in some cases, does not even need to map to any existing biological phenomenon, while the model data is dependent on, or needs a mapping to a specific model and/or to a specific experimental condition. To add to this phenomenon, most formats can only store the model structure, leaving the model data as an extra that is often neglected, being in some cases stored as figures in a scientific article or absent altogether.

3.2.1 SBtab

The Systems Biology table format (SBtab) was chosen to store all the relevant information for the models included in this dissertation. This is a previously published standard [377] where the biochemical model structure and its data can be easily stored. Its structure follows syntax rules and conventions that make it human readable, easily shareable, and editable. These rules are flexible enough to be easily expandable, which allowed the addition of extra

fields when needed. There is an on-line tool to convert SBtab to Systems Biology Markup Language (SBML) that ensures the interoperability of the format [405], but due to the expansions that have been performed, the models included in this dissertation are not compatible with it. To address this problem, a few custom conversion tools were developed to convert SBtab into native MATLAB formats (discussed further in section 4.2.1), and additionally to convert SBtab into SBML, vector field .vf and NEURON .mod files (out of scope of this dissertation) [1]. Once in a MATLAB format, the model can also be converted through MATLAB built-in tools into SBML, preserving interoperability. Due to the straightforward organization of SBtab, this parser could easily be further expanded or customized in the future.

The SBtab file can be constructed out of many different sheets, although usually only some of them are needed for a specific use case. In this dissertation, the models have separate sheets for: compounds, reactions, parameters, outputs, experiments, individual experiments' data, and individual experiments' inputs. In each of these sheets, as per SBtab specification, the first row is used for the declaration of file attributes as in the following example: "Document='Example' TableName='Compound' TableType='Compound' TableTitle='Example Compounds'" (for the compound table in LiREM). The second row is used for column headers and all following rows are used to store the relevant data relative to each header. The first column in each sheet is special as it should always contain an unique identifier for the entries. A description of all the headers used in the various sheets can be found in table 3.1 and supplementary tables A.1 to A.6. The headers that start with "!" are unique to all the different sheets, and the headers that start with "$>>$" refer to the identification (ID) of an entry on another sheet. It should be noted that some of the SBtab files used in this dissertation have extra fields or sheets in order to be compatible with software that is out of the scope of this dissertation.

3.3 Simulation

In order to create a simulation of molecular reactions capable of producing meaningful results, the processes happening in biological systems need to be converted into equations. Each of the biological processes that describe the system is translated into a series of biochemical reactions ruled by the principles of the rate law. Then, ODE are derived from these biochemical reactions to create a system of ODEs that can be solved to simulate the behaviour of the system.

Compound Fields	
Field Name	**Description**
!ID	Unique identifier for the compound within the SBtab
!Name	Name of the compound
!Unit	Measurement unit for the compound's concentration or quantity
!InitialValue	Default initial concentration or quantity of the compound
!IsConstant	Boolean value (True/False) indicating whether the compound's concentration remains constant during simulation
!Location	Cellular compartment to which the compound is assigned
!ReferenceDOI	Digital Object Identifier (DOI) for the source of compound information

Table 3.1: Fields utilized in the 'Compound' sheet of the SBtab format. Each field provides specific information about model compounds, including their properties, initial conditions, and associated metadata. This structure enables a comprehensive representation of the chemical species involved in the modeled biochemical system.

3.3.1 Reaction kinetics

Considering the following non-reversible reaction:

$$aA + bB \longrightarrow cC \tag{3.4}$$

A and B are reactants, C is a product (together referred to as "species"); and a, b and c their stoichiometric constants. This reaction is governed by how fast the reactants can be converted into products, which is called the reaction rate and, according to the rate law, it is defined as:

$$rate = -\frac{1}{a}\frac{d[A]}{dt} = -\frac{1}{b}\frac{d[B]}{dt} = \frac{1}{c}\frac{d[C]}{dt} = k[A]^{\alpha}[B]^{\beta} \tag{3.5}$$

where $[A]$ is the concentration of species A and $\frac{d[A]}{dt}$ its rate of change (same notation for other species). α and β can be determined experimentally, but are usually considered to be the stoichiometric constants for the reaction, respectively a and b, particularly when considering the common approximation that the system follows the law of mass action. The reaction order is the sum of the rate coefficients ($\alpha + \beta$ in this case). These rate equations derive from the Law of Mass Action, which posits that the rate of a reaction is directly proportional to the concentration of each reactant, raised to a power equal to its stoichiometric coefficient in the balanced chemical equation [406].

In order to model the system, it is necessary to take into account how all the species of the reaction evolve over time. This can be achieved by solving a system of ODEs that in this simple case are the following:

$$\begin{aligned} \tfrac{d[A]}{dt} &= -a\left[k[A]^{\alpha}[B]^{\beta}\right] = -a \times rate \\ \tfrac{d[B]}{dt} &= -b\left[k[A]^{\alpha}[B]^{\beta}\right] = -b \times rate \\ \tfrac{d[C]}{dt} &= c\left[k[A]^{\alpha}[B]^{\beta}\right] = c \times rate \end{aligned} \tag{3.6}$$

If the reaction was reversible, the total reaction rate would be $k_f[A]^{\alpha}[B]^{\beta} - k_r[C]^{\gamma}$, which is composed of a forward (positive portion) and a reverse (negative portion) rate [406].

3.3.2 Ordinary differential equation systems

One way of accurately representing the dynamics of biochemical pathways in computational simulations of biological systems is to solve ODE systems. These equations are central to capturing the changes in concentration of reactants and products over time, as seen in reaction kinetics. This subsection will outline the construction and application of ODE systems to model these biological interactions. Specifically, ODE systems like the one referenced in equation (3.6) must be solved.

Given a starting point for the simulation of the reaction represented in equation (3.4)

$$[A] = [A_0] \quad [B] = [B_0] \quad [C] = [C_0]$$

each subsequent iteration of the simulation would have the species evolving according to the ODE. In this simple case it would be

$$\begin{aligned} [A^{new}] &= [A] - a \times rate \times time_step \\ [B^{new}] &= [B] - b \times rate \times time_step \\ [C^{new}] &= [C] + c \times rate \times time_step \end{aligned} \tag{3.7}$$

where the $time_step$ would determine how many iterations would be needed to simulate the reaction for a certain period of time. Since at each iteration all species' concentrations are modified and these will be used to calculate the concentrations at next iteration, there is a trade off between time step size and solver accuracy, where smaller time steps correspond to greater accuracy and vice-versa. This incremental update approach, known as the Euler method, is one of many possible techniques for approximating solutions to ODEs, shown here for its simplicity [407].

3.4 Scoring functions

The output (O) of the model is treated as a function (f) with

$$\theta = \Big[lb_1 \leqslant P_1 \leqslant ub_1, \quad lb_2 \leqslant P_2 \leqslant ub_2, \quad \cdots, \quad lb_p \leqslant P_p \leqslant ub_p, \quad \cdots, \quad lb_q \leqslant P_q \leqslant ub_q \Big] \tag{3.8}$$

as inputs; where a parameterization (θ) with q parameters (Ps) constrained between a lower bound (lb) and an upper bound (ub), and index p; such that

$$O = f(\theta) = f(P_1, P_2, \cdots, P_p, \cdots, P_q) \tag{3.9}$$

In this dissertation, O can be either $\hat{y}(\theta)$, the time series model output given a specific θ, or one of the scoring groupings in equations (3.12), (3.13) and (3.14).

The main scoring function (s_f) chosen as the foundation of the comparison between model outputs and experiments is based on the weighted sum of square residuals (WSSR), which is derived from the Least Squares method [408]:

$$WSSR(\theta|y) = \sum_{i=1}^{n} \left(\frac{y_i - \hat{y}_i(\theta)}{\sigma_i} \right)^2 \tag{3.10}$$

the various symbols represent the data points collected from the experiment (y), the standard deviation (σ) associated with these data points, the points generated by the model

under parametrization θ ($\hat{y}(\theta)$), and the index and number (i/n) of measured points in the simulation and experiment.

Artificial data was generated for some experiments, in which case the amount of points generated for each experimental output was arbitrary. In order to account for this, the WSSR has been slightly modified by dividing it by the number of points in each experimental output. This new function corresponds then to the mean of the weighted sum of squared residuals:

$$s_f(\theta|y,\sigma) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i(\theta)}{\sigma_i}\right)^2 = \sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i(\theta)}{\sigma_i\sqrt{n}}\right)^2 \tag{3.11}$$

This s_f provides us with a metric for a single time series comparison between model and data. Generally, there are multiple experimental outputs in multiple experiments, each one of them with an associated score. These are grouped in three different ways: score per experimental output (s_{Eo}), score per experiment (s_E), and total score (s_t). The group of all the scores of all the experimental outputs will be referred to as s_{Eo}:

$$s_{Eo}(\theta|y,\sigma) = \begin{bmatrix} s_{f_{1,1}} & s_{f_{1,2}} & \cdots & s_{f_{1,l}} \\ s_{f_{2,1}} & s_{f_{2,2}} & \cdots & s_{f_{2,l}} \\ \cdots & \cdots & \cdots & \cdots \\ s_{f_{m,1}} & s_{f_{m,2}} & \cdots & s_{f_{m,l}} \end{bmatrix} \tag{3.12}$$

where m corresponds to the number of experimental outputs and l corresponds to the number of experiments. The sum of the scores of the experimental outputs of each experiment is referred to as s_E:

$$s_E(\theta|y,\sigma) = \sum_{j=1}^{m} {s_{Eo}}^{\mathrm{T}} = \begin{bmatrix} \sum_{j=1}^{m} s_{f_{j,1}} \\ \sum_{j=1}^{m} s_{f_{j,2}} \\ \cdots \\ \sum_{j=1}^{m} s_{f_{j,l}} \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{m} s_{f_{j,1}} & \sum_{j=1}^{m} s_{f_{j,2}} & \cdots & \sum_{j=1}^{m} s_{f_{j,l}} \end{bmatrix}^{\mathrm{T}} \tag{3.13}$$

where j corresponds to the index of experimental outputs and T corresponds to the transpose operation. Finally, the sum of all the experimental scores is referred to as s_t:

$$s_t(\theta|y,\sigma) = \sum_{k=1}^{l} s_{Eo} = \sum_{k=1}^{l}\sum_{j=1}^{m} s_{f_{j,k}} \tag{3.14}$$

where k corresponds to the index of experiments. The s_t has been used for parameter estimation.

3.5 Parameter estimation

Using the scoring functions defined in section 3.4, it becomes possible to approximate the behaviour of the models to the behaviour of the model data by performing parameter

estimation. The models in this dissertation are complex multidimensional systems so this process must be accomplished through optimization algorithms. In this section, the optimization algorithms used inside MATLAB are briefly explained, and a Pseudocode for each of the algorithms is presented.

3.5.1 Interior point algorithm

The interior point algorithm is a type of optimization algorithm that can deal well with inequality constraints. The algorithm travels the parameter space throughout the interior or exterior of the feasible parameter space but never crosses the boundary. Using the barrier method, the algorithm transforms a problem with inequality constraints in such a way that these can be included implicitly into the objective function. Then the algorithm uses Newton's method of optimization to approximate the optimal solution. The following explanation is derived from the MATLAB documentation [409].

Given a function to minimize with inequality constraints such as:

$$f(X), \quad g_i(X) \leq 0$$

the barrier method would transform the inequality constraints into a barrier function that could be added to the function being minimized. This would generate a new minimization problem that would take into account the constraints so that solutions that approximate the constraints would get an exponentially increasing penalty. The barrier function for the example given above would be:

$$-\mu \sum_{i}^{m} log(-g_i(x))$$

This function increases exponentially as $g_i(x)$ approaches 0. The variable μ is added so this term can be reduced during optimization in order to let the solver approach an unbiased solution. The overall minimization function will then be:

$$f(X) - \mu \sum_{i}^{m} log(-g_i(x))$$

The steps needed for this algorithm are the following:

(A) Convert the minimization function and inequality constraints into a new minimization function that incorporates the inequality constraints implicitly;

(B) Execute points C and D for a certain number of iterations;

(C) Optimize this new minimization function using Newton's method;

(D) Reduce the importance of the inequality constraints factor;

(E) Output the final parametrization after all the iterations.

These steps can also be expressed in the form of pseudocode represented in algorithm 1.

Algorithm 1 Interior point method pseudocode. k is the iteration number of the algorithm; μ is a factor governing the importance of the barrier portion of the new minimization function, it is reduced at every iteration through a $reducing_factor$ that is a number between 0 and 1 (non-inclusive); and X, x, p, q, lb, ub and $f(X)$ are the same as in equations (3.8) and (3.9).

$X = X^0$

Given $f(X), \quad X > lb, \quad X < ub$

Transform to $f(X) - \mu \sum_p^q [log(x_p - lb_p) + log(ub_p - x_p)]$

for $k = 0$ through k_{max} **do**

$X^{new} \leftarrow Newton_method(f(X) - \mu \sum_p^q [log(x_p - lb_p) + log(ub_p - x_p)])$ ▷

$Newton_method$ minimizes the function given utilizing the Newton's method

$\mu^{new} \leftarrow \mu \times reducing_factor$

end for

Output the final X

3.5.2 Simulated annealing

The simulated annealing algorithm is inspired by the physical process of annealing in metallurgy, where a material, such as metal or glass, is heated to a high temperature and then gradually cooled to remove defects and reach a more stable structure. The gradual cooling allows the system to explore a wide range of configurations, potentially avoiding local minima and finding a configuration with lower internal energy. This principle is adapted in simulated annealing to solve optimization problems by iteratively searching for a global minimum.

The simulated annealing process begins with an initial random solution and a high temperature term, which allows the algorithm to explore a broad solution space. As the temperature term decreases, the algorithm becomes more selective, refining the solution. At each optimization step, the algorithm decides whether to accept a new candidate solution based on an acceptance function that depends on the temperature and the change in the objective function value. This acceptance of sometimes suboptimal solutions helps the algorithm avoid getting trapped in local optima.

The following steps, adapted from the MATLAB documentation [410], outline the simulated annealing algorithm:

(A) Start with a random parametrization and a temperature above 0;

(B) Execute points C through F for a certain number of iterations;

(C) Create a new parametrization at a certain geometric distance from the previous one dependent on the temperature (higher temperatures, larger distances and the reverse for lower temperatures);

(D) Depending on a selection function, either select the original parametrization or the new one;

(E) Lower the temperature;

(F) Output the final parametrization after all the iterations.

Pseudocode for the algorithm using an example selection function and temperature minimization procedure taken from MATLAB's documentation [410] can be found in algorithm 2.

Algorithm 2 Simulated annealing pseudocode. T is a temperature; k is the iteration number of the algorithm; d_X is a scalar representing a geometric distance from the previous parametrization; and X, x, p, and $f(X)$ are the same as in equations (3.8) and (3.9).

$X = X^0$

$T = T^0$

for $k = 1$ through k_{max} **do**

  for p through $Parameter_number$ **do** ▷ Iterate over all elements of X.

    $d_{x_p} \leftarrow x_p \times T \times random(0, 1)$ ▷ $random$ function selects a random number uniformly between the first and second entries of the function

  end for

  if $\frac{1}{1+exp\left(\frac{f(X)-f(X+d_X)}{T_0}\right)} < random(0, 1)$ **then** ▷ MATLAB's selection function.

    $X^{new} \leftarrow X + d_X$

  end if

  $T^{new} \leftarrow T \times 0.95^k$ ▷ MATLAB's formula for temperature minimization.

end for

Output the final X

3.5.3 Pattern search

The pattern search algorithm is a derivative-free optimization method designed to find the optimal parameter values that minimize a given objective function. By exploring multiple directions from an initial guess, pattern search evaluates the function iteratively at mesh points (neighbouring points within a defined grid). This algorithm is particularly useful in cases where the objective function is noisy, discontinuous, or lacks a known gradient, and where more conventional optimization methods might struggle to converge.

At each iteration, pattern search polls nearby points in a systematic pattern to locate an improved function value, moving the current solution toward a minimum. The algorithm progresses as follows, with the key steps adapted from MATLAB documentation [411]:

(A) Start with a random parametrization;

(B) Execute points C and D for a certain number of iterations;

(C) Go through a series of polling vectors (an example of such polling vectors can be found in figure 3.1 using MATLAB's GPSPositiveBasis2N polling method [411]), add the vector to the parametrization and evaluate the output function. If the output function has a lower value than the previous best, then the polling is considered successful. If one vector is chosen, the remaining vectors can be discarded to speed up the iteration;

(D) If the polling is successful, multiply the polling vectors by a constant. Otherwise divide it by a constant;

(E) Output the final parametrization after all the iterations.

These steps can also be described in the form of a pseudocode exemplified in algorithm 3.

3.5.4 Genetic algorithm

The genetic algorithm is an optimization method inspired by natural selection, where a population of candidate solutions evolves over successive generations. Each individual in the population represents a possible solution, defined by a specific set of parameters, and is assigned a fitness score based on its effectiveness in solving the target problem. Guided by processes analogous to biological reproduction, selection, and mutation, the algorithm iteratively refines the population, promoting high-fitness individuals and generating new solutions through crossover (combining pairs) and mutation (random alteration).

$$
\begin{array}{cccc}
 & & & V_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \\
 & V_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} & & V_2 = \begin{bmatrix} 0 & 1 & \cdots & 0 \end{bmatrix} \\
V_1 = \begin{bmatrix} 1 & 0 \end{bmatrix} & V_2 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} & & \vdots \quad \vdots \quad \vdots \quad \vdots \\
V_2 = \begin{bmatrix} 0 & 1 \end{bmatrix} & V_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} & & V_q = \begin{bmatrix} 0 & 0 & \cdots & 1 \end{bmatrix} \\
V_3 = \begin{bmatrix} -1 & 0 \end{bmatrix} & V_4 = \begin{bmatrix} -1 & 0 & 0 \end{bmatrix} & & V_{q+1} = \begin{bmatrix} -1 & 0 & \cdots & 0 \end{bmatrix} \\
V_4 = \begin{bmatrix} 0 & -1 \end{bmatrix} & V_5 = \begin{bmatrix} 0 & -1 & 0 \end{bmatrix} & & V_{q+2} = \begin{bmatrix} 0 & -1 & \cdots & 0 \end{bmatrix} \\
 & V_6 = \begin{bmatrix} 0 & 0 & -1 \end{bmatrix} & & \vdots \quad \vdots \quad \vdots \quad \vdots \\
 & & & V_{2q} = \begin{bmatrix} 0 & 0 & \cdots & -1 \end{bmatrix}
\end{array}
$$

Figure 3.1: Example polling vectors generated in MATLAB when using the GPSPositiveBasis2N polling method [411], for 2, 3, and q parameters.

At each iteration, the genetic algorithm selects individuals as parents based on their fitness, preserves the highest-performing solutions as elite, and creates children by applying crossover and mutation to the parent genes. This population renewal process continues until a specified stopping condition is met, such as reaching a maximum number of generations or achieving a target fitness level.

These steps, adapted from the MATLAB documentation [412], outline the algorithm's process:

(A) Start with a random population of parametrizations;

(B) Execute points C-I for a certain number of iterations;

(C) Create a new population based on the current one following the steps described in points D-H;

(D) Compute $f(X)$ for all the elements of the current population;

(E) Select a group of this population based on their their scores, these are called the "parents" of the new generation;

(F) Choose the parametrizations with the best score and pass them unchanged to the new population, these are called the "elite";

(G) Use the parent parametrization to create a set of new parametrizations, called "children", either by the process of making random changes to the individual parameters of

Algorithm 3 Pattern search pseudocode. k is the iteration number of the algorithm; V represents a set of polling vectors that define the search grid; ξ is a scaling factor for expanding or contracting the polling vectors; and X, p, q, and $f(X)$ are the same as in equations (3.8) and (3.9).

$X = X^0$
$V = V^0$
for $k = 1$ through k_{max} **do**
  while $p < 2q$ **do**
    if $f(X + V_p) < f(X)$ **then**
      $X^{new} \leftarrow X + V_p$
      $p = 2q$ ▷ Optional, depending on complete or greedy search.
    end if
    $p = p + 1$
  end while
  if $f(X^{new}) < f(X)$ **then**
    $V \leftarrow V/\xi_1$
  else
    $V \leftarrow V \times \xi_2$
  end if
end for
Output the final X

a parametrization, process referred to as "mutation", or by combining the parameters of a pair of parents, process referred to as "crossover";

(H) Replace the current population by a new population consisting of the children and the elite;

(I) Output either the final population of parametrizations or the best parametrization of that population after all the iterations.

These steps can also be expressed in the form of pseudocode represented in algorithm 4.

Algorithm 4 Genetic algorithm pseudocode. k is the iteration number of the algorithm; Pop represents a population of individuals and can have different sub categories (Elite, Parent, Crossover, Mutated); and X, p, lb, ub and $f(X)$ are the same as in equations (3.8) and (3.9).

$Pop = Pop^0 = [X_1^0, X_2^0, \cdots, X_j^0, \cdots, X_{Pop_size}^0]$

for $k = 0$ through k_{max} **do**

$Sorted_Pop = sort\,(Pop \mid f(Pop))$

divider = $floor(Pop_size \times 0.1)$ ▷ $floor$ function rounds to the lowest integer

if $divider > 1$ **then**

$Elite_Pop = Sorted_Pop_{1-divider}$

end if

for $j = 1$ through $Pop_size - divider$ **do**

$Parent_1 = weightedSample(Sorted_Pop)$ ▷ $weightedSample$ function samples the sorted population favouring the individuals with better scores.

$Parent_2 = weightedSample(Sorted_Pop)$

if $random(0, 1) < 0.5$ **then**

for $p = 0$ through $Parameter_number$ **do**

$Mutated_Children_p = max(min(Parent_1_p + random(0, 1), ub_p), lb_p)$

end for

$Children_Pop_j = Mutated_Children$

else

for $p = 0$ through $Parameter_number$ **do**

if $random(0, 1) < 0.5$ **then**

$Crossover_Children_p = Parent_1_p$

else

$Crossover_Children_p = Parent_2_p$

end if

end for

$Children_Pop_j = Crossover_Children$

end if

end for

$Pop^{New} = [Elite_Pop, Children_Pop]$

end for

Output the final Pop or the best X of the final Pop

3.5.5 Particle swarm

The particle swarm optimization algorithm, rooted in the behavioural patterns of flocking birds or schooling fish, models optimization as a collective, goal-oriented movement through a solution space. Particle swarm optimization enhances solutions iteratively based on interactions among particles (potential solutions) within a defined space. Each particle adjusts its velocity and position based on both its own best-known position and the best-known positions within its neighbourhood.

The particle swarm optimization process begins with particles initialized randomly within solution bounds, followed by repeated updates to particle velocities and positions toward optimal solutions. This method is guided by parameters controlling individual and social behaviour, neighbourhood structure, and inertia. The iterative update continues until a stopping criterion is met, such as a minimum improvement threshold or a time limit.

It consists of the following steps, adapted from the MATLAB documentation [413]:

(A) Start with a random group of parametrizations, uniformly distributed inside the bounds and each with a corresponding velocity vector. Each parametrization is going to be referred to as "particle" and the grouping of all parametrizations as "swarm";

(B) Execute points C-E for a certain number of iterations;

(C) Evaluate $f(X)$ for each particle;

(D) Choose new velocity vectors based on: the function value and the previous velocity of the particle, and the function values and velocities of a neighbouring group (in a way that the entire population tends to go toward the best point found by the entire population);

(E) Output new positions for each particle combining the particle position and velocity (the velocity is modified if it would otherwise result in the particle going out of bounds);

(F) Output either the final swarm of parametrizations or the best parametrization of that swarm after all the iterations.

These steps can also be expressed in the form of pseudocode represented in algorithm 5.

Algorithm 5 Particle swarm pseudocode. k is the iteration number of the algorithm; $Innertia$, X_weight, and $Group_weight$ are scalars between 0 and 1 that represent respectively the importance of: the previous velocities, the last best position for a particle, and the best score of the group of particles that each particle samples at each iteration; V is the velocity factor for parameters; $Swarm_size$ and $Group_size$ are, respectively, the swarm and group size; and X, x, p, q, lb, ub and $f(X)$ are the same as in equations (3.8) and (3.9).

$Swarm = Swarm^0 = [X_1^0, X_2^0, \cdots, X_j^0, \cdots, X_{Swarm_size}^0]$

$V_* = V_*^0 = [V_1^0, V_2^0, \cdots, V_j^0, \cdots, V_{Swarm_size}^0]$ ▷ each V contains a v, $V = \left[0 \leqslant v_1 \leqslant V_{max} \quad 0 \leqslant v_2 \leqslant V_{max} \quad \cdots \quad 0 \leqslant v_p \leqslant V_{max} \quad \cdots \quad 0 \leqslant v_q \leqslant V_{max}\right]$ analogous with equation (3.8).

for $k = 1$ through k_{max} **do**

  for $j = 1$ through $Swarm_size$ **do**

    $X_j^{best} = X_j$

    for $i = 1$ through $Group_size$ **do**

      $a = floor(random(1, Swarm_size))$

      $Group_i = X_a$

    end for

    $[Score_Group^{best}, Group^{best}] = min(f(Group))$ ▷ such that $Score_Group^{best} = f(Group^{best})$.

    $V_j^{new} = Innertia \times V_j + X_weight \times random(0,1) \times (X_j^{Best} - X_j) + Group_weight \times random(0,1) \times (Group^{Best} - X_j)$ ▷ MATLAB's formula for updating the velocity.

    $X_j^{new} = X_j + V_j^{new}$

    for $p = 1$ through $Parameter_number$ **do**

      if $x_{j,p}^{new} < lb_p \bigvee x_{j,p}^{new} > ub_p$ **then**

        $x_{j,p}^{new} = max(min(x_{j,p}^{new}, ub_p), lb_p)$

        $v_{j,p}^{new} = 0$

      end if

    end for

    if $f(X_j^{new}) < f(X_j^{Best})$ **then**

      $X_j^{best} = X_j^{new}$

    end if

  end for

  $Swarm^{new} = [X_1^{new}, X_2^{new}, \cdots, X_j^{new}, \cdots, X_{Swarm_size}^{new}]$

end for

Output the final $Swarm$ or the best X of the final $Swarm$

3.5.6 Surrogate optimization

The surrogate optimization algorithm is designed to minimize objective functions that are computationally expensive to evaluate, by approximating them with a surrogate model that allows for faster optimization. This approach leverages a surrogate function that mimics the behaviour of the actual objective function across selected parameter points, significantly reducing computation time for complex optimization problems.

In each iteration, the algorithm alternates between constructing the surrogate and conducting a search for the objective function's minimum. By progressively refining the surrogate model with points that yield promising outcomes, the algorithm narrows down on an optimal solution efficiently, while minimizing costly evaluations of the objective function itself.

The following steps, adapted from MATLAB documentation [414], outline the algorithm's process:

(A) Execute points B-D for a certain number of iterations;

(B) Construct the surrogate by evaluating the objective function in a set number of points greater than the number of parameters;

(C) Search for a minimum of the surrogate function through any optimization procedure;

(D) Evaluate the original function with the parametrization that best fits the surrogate function and update the surrogate function to include the result of this evaluation;

(E) Output the best parametrization after all the iterations.

These steps can also be expressed in the form of pseudocode represented in algorithm 6.

Algorithm 6 Surrogate optimization pseudocode. k is the iteration number of the algorithm; $sample$ are parametrizations created to be evaluated by the surrogate function; $range$ defines the area around the best parametrization, evaluated with the expensive function (f(X)), where sample parametrizations can be created; $sample_number$, and $Parameter_number$ are respectively the number of samples to be drawn to construct the surrogate, and number of parameters in the model; the superscript $best$, $score_best$, $scaled$, and $adaptive$ are respectively, the parameter set that minimizes a certain function, the minimum point of a function minimized by a parametrization with superscript $best$, a scaled variable, and a parameter set that is obtained by the process of minimizing the surrogate function; and X, x, p, lb, ub and $f(X)$ are the same as in equations (3.8) and (3.9).

$X_* = X_*^0 = [X_1^0, X_2^0, \cdots, X_j^0, \cdots, X_{X_number}^0]$

for $k = 1$ through k_{max} **do**

$X^{best} = min(f(X_*))$

for $j = 1$ through $sample_number$ **do**

for $p = 1$ through $Parameter_number$ **do**

$sample_{j,p} = max(min(random(x_p^{best} - range, x_p^{best} + range), ub_p), lb_p)$

end for

$g_j = surrogate(sample_{j,*} \mid [X_* \wedge f(X_*)])$ ▷ $surrogate$ function calculates a score using a fast to evaluate function dependent on the evaluations of the expensive function, MATLAB uses a Radial Basis Function interpolation to give surrogate scores.

$d_j = distance(X^{best}, sample_{j,*})$ ▷ $distance$ function calculates the geometric distance between two parametrizations.

end for

for $j = 1$ through $sample_number$ **do**

$g_j^{scaled} = \frac{g_j - min(g)}{max(g) - min(g)}$

$d_j^{scaled} = \frac{d_j - min(d)}{max(d) - min(d)}$

$m_j = merit(sample_j) = w \times g_j^{scaled} + (1 - w) \times d_j^{scaled}$

end for

$[sample^{score_best}, sample^{best}] = min(m_*)$ ▷ such that $sample^{score_best} = merit(sample^{best})$.

$X^{adaptive} = sample^{best}$

$X_*^{new} = [X_*, X^{adaptive}]$

end for

Output the final X

3.6 Sensitivity analysis

The sensitivity calculation assumes that O (see equation 3.9) is a scalar. For representations that are a group of scalars, a sensitivity index is calculated for each one of them.

3.6.1 Local sensitivity analysis

The local sensitivity analysis was implemented as described in the 1991 paper by Morris [415] and commonly known as the Morris method.

The θ were sampled as described in section 3.6.1.1. Each θ is varied one factor at a time (OAT approach) by a certain value Δ, defining T trajectories. The elementary effect (d_p) of these changes is calculated as

$$d_p(\theta) = \frac{f(x_1, x_2, ..., x_{p-1}, x_p + \Delta, x_{p+1}, ..., x_q) - f(\theta)}{\Delta}, \quad \begin{cases} x_p \leq x_{p_{ub}} - \Delta, & \text{if } \Delta > 0 \\ x_p \geq x_{p_{lb}} - \Delta, & \text{otherwise} \end{cases}$$

where p is the index of the parameter being worked on and $x_{p_{ub}}$ and $x_{p_{lb}}$ denote the upper and lower bounds of said parameter. Given the d_p of all T we can then calculate the the mean

$$\mu_{d_p} = \frac{1}{T} \sum_{1=1}^{T} d_p$$

and the standard deviation

$$\sigma_{d_p} = \sqrt{\frac{1}{T-1} \sum_{1=1}^{T} (d_p - \mu_p)^2} \tag{3.15}$$

Given that the sum of different elementary effect (d_p) can nullify each other, another relevant statistic is the mean absolute effect ($\mu^*_{d_p}$) calculated as:

$$\mu^*_{d_p} = \frac{1}{T} \sum_{1=1}^{T} |d_p| \tag{3.16}$$

According to Morio [416], the effect of x_p on $f(\theta)$ is characterized by $\mu^*_{d_p}$, and σ_{d_p} serves as a measure of non-linear effects and interactions of x_p with other parameters ($x_{\sim p}$). These effects can combine to be interpreted in three different ways:

- Small values of $\mu^*_{d_p}$ should be interpreted as having a x_p of negligible importance.
- Large values of $\mu^*_{d_p}$ and small values of σ_{d_p} should let us conclude that x_p is important but has only linear effects and small or no amount of interactions with other parameters.

- Large values of both $\mu^*_{d_p}$ and σ_{d_p} prompt the interpretation that x_p is both important and has non-linear effects and/or interactions with other parameters.

3.6.1.1 Parameter sampling for local sensitivity analysis

This sampling method was suggested by Morris [415] as a way to reduce the number of model evaluations needed to compute T trajectories for each parameter x_i. The naive method assumes that we need to evaluate the model 2 times for each d_p, one for the original value of θ and one for $x_i + \Delta$, for a total of $2qT$ model evaluations. Here instead a $q+1$-by-q matrix, $\boldsymbol{A}^*$, was generated, so that for each initial θ it generates one elementary effect for all parameters (one at a time). T matrices $\boldsymbol{A}^*$ are then concatenated to generate matrix

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{A}^*_1 \\ \boldsymbol{A}^*_2 \\ \vdots \\ \boldsymbol{A}^*_T \end{bmatrix}$$

that is used for all the necessary calculations of d_p. The number of model evaluations is then reduced from $2qT$ to $T(q+1)$.

Matrices $\boldsymbol{A}^*$ are constructed in the following manner: first, a $q+1$-by-q matrix, $\boldsymbol{A}$, was generated, where all elements are either 0's or 1's and where two rows differ from one another by the pth column element, for example:

$$\boldsymbol{A}_{q+1,q} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix}$$

Then, the following transformation is applied to $\boldsymbol{A}$:

$$\boldsymbol{A}^\dagger = \left[Z^* \boldsymbol{J}_{q+1,1} + \frac{\Delta^\dagger}{2} \left[(2\boldsymbol{A} - \boldsymbol{J})\boldsymbol{D} + \boldsymbol{J} \right] \boldsymbol{P} \right]$$

where $\boldsymbol{J}_{q+1,q}$ is a matrix of ones; $\boldsymbol{D}_{q,q}$ is a diagonal matrix in which its elements have equal probability of being either 1 or -1; Z^* is a vector of length q where its values are randomly sampled, with equal probability, from the set $\left\{0, \frac{1}{g-1}, \frac{2}{g-1}, ..., 1 - \Delta^\dagger\right\}$; g is a non-zero natural number, that defines the search grid from which parameter values are chosen. It is chosen in conjunction with $\Delta^\dagger$ to define the possible values that the entries of $\boldsymbol{A}^\dagger$ can take, as shown in

equation (3.17). $\Delta^\dagger$ is a value chosen from the set $\left\{\frac{1}{g-1}, \frac{2}{g-1}, ..., \frac{g-2}{g-1}\right\}$, and $\boldsymbol{P}_{q,q}$ is a random permutation matrix where one and only one entry of all rows and columns are equal to 1 and all others are equal to 0, with equal probability of choosing each such matrix.

The values for the elements of $\boldsymbol{A}^\dagger$ will be taken from the set

$$\begin{cases} \left\{0, \frac{1}{g-1}, \frac{2}{g-1}, ..., \frac{g-2}{g-1}, 1\right\}, \text{with double frequency for } \left\{\Delta^\dagger, 2\Delta^\dagger, ..., 1-\Delta^\dagger\right\}, & \text{if } \Delta^\dagger < \frac{g/2}{g-1} \\ \left\{0, \frac{1}{g-1}, \frac{2}{g-1}, ..., \frac{g-2}{g-1}, 1\right\}, \text{with equal frequency for all its elements,} & \text{if } \Delta^\dagger = \frac{g/2}{g-1} \\ \left\{0, \frac{1}{g-1}, ..., 1-\Delta^\dagger, 0+\Delta^\dagger, ..., \frac{g-2}{g-1}, 1\right\}, & \text{otherwise} \end{cases} \tag{3.17}$$

and values from two consecutive rows will always differ in one and only one of its columns by $\Delta^\dagger$.

Since the parameters in this dissertation do not always vary between 0 and 1, $\boldsymbol{A}^\dagger$ and $\Delta^\dagger$ are scaled in such a way that ensures they conform to the required parameter bounds, resulting in

$$\boldsymbol{A}^* = \boldsymbol{A}^\dagger \odot (\theta_{ub} - \theta_{lb}) + \theta_{lb}, \quad \Delta = \Delta^\dagger(\theta_{ub} - \theta_{lb})$$

where θ_{lb} and θ_{ub} are the lower and upper bounds respectively of the possible parametrizations θ, and $\odot$ represents the element-wise product.

3.6.2 Global sensitivity analysis

The procedure used to obtain the sensitivity indices depicted in this section was based on the Sobol and Saltelli method [417, 418], and adapted from Halnes *et al*. [419].

3.6.2.1 First and total order sensitivities calculation

To calculate the sensitivities, two matrices:

$$\boldsymbol{A} = \begin{bmatrix} A_{1,*} \\ A_{2,*} \\ \vdots \\ A_{r,*} \end{bmatrix} = \begin{bmatrix} X_1' \\ X_2' \\ \vdots \\ X_r' \end{bmatrix} = \begin{bmatrix} x_{1,1}' & x_{1,2}' & \cdots & x_{1,q}' \\ x_{2,1}' & x_{2,2}' & \cdots & x_{2,q}' \\ \vdots & \vdots & \ddots & \vdots \\ x_{r,1}' & x_{r,2}' & \cdots & x_{r,q}' \end{bmatrix} \tag{3.18}$$

$$\boldsymbol{B} = \begin{bmatrix} B_{1,*} \\ B_{2,*} \\ \vdots \\ B_{r,*} \end{bmatrix} = \begin{bmatrix} X_1'' \\ X_2'' \\ \vdots \\ X_r'' \end{bmatrix} = \begin{bmatrix} x_{1,1}'' & x_{1,2}'' & \cdots & x_{1,q}'' \\ x_{2,1}'' & x_{2,2}'' & \cdots & x_{2,q}'' \\ \vdots & \vdots & \ddots & \vdots \\ x_{r,1}'' & x_{r,2}'' & \cdots & x_{r,q}'' \end{bmatrix} \tag{3.19}$$

are generated by performing a Monte Carlo sampling, using one of the distributions mentioned in section 3.6.2.2, until there are r model parametrizations for each of the matrices. These matrices are then combined to construct a third matrix:

$$\boldsymbol{C}^p = \begin{bmatrix} C^p_{1,*} \\ C^p_{2,*} \\ \vdots \\ C^p_{r,*} \end{bmatrix} = \begin{bmatrix} B_{1,1} & B_{1,2} & \cdots & A_{1,p} & \cdots & B_{1,q} \\ B_{2,1} & B_{2,2} & \cdots & A_{2,p} & \cdots & B_{2,q} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ B_{r,1} & B_{r,2} & \cdots & A_{r,p} & \cdots & B_{r,q} \end{bmatrix} \tag{3.20}$$

by replacing the pth parameter of matrix $\boldsymbol{B}$ by the corresponding entry of matrix $\boldsymbol{A}$ for each of the p layers of matrix $\boldsymbol{C}$, with p as the index of the parameter for which the sensitivity is going to be calculated.

An estimation of the first order sensitivities of parameter p is then:

$$\widehat{S}_p = \frac{\widehat{V}\left[\widehat{E}(O|x_p)\right]}{\widehat{V}(O)} = \frac{\widehat{U}_p - \widehat{E}^2(O)}{\widehat{V}(O)}$$

where $\widehat{V}$ is an estimator for the variance, $\widehat{E}$ is an estimator for the expected value, and x_p is the model pth parameter. The factor $\widehat{U}_p$ is given by:

$$\widehat{U}_p = \frac{1}{r-1}\sum_{s=1}^{r} f(A_{s,*}) f(C^p_{s,*})$$

where s is the index of the samples. An estimation of the square of the expected value $\widehat{E}^2(O)$ is given by:

$$\widehat{E}^2(O) = \frac{1}{r}\sum_{s=1}^{r} f(A_{s,*}) f(B_{s,*}) \tag{3.21}$$

and the estimation of the Variance $\widehat{V}(O)$ is given by:

$$\widehat{V}(O) = \frac{1}{r-1}\sum_{s=1}^{r} f^2(A_{s,*}) - \widehat{E}^2(O)$$

An estimation of the total order sensitivities of parameter p is respectively:

$$\widehat{S}_{Tp} = \frac{\widehat{V}(O) - \widehat{V}\left[\widehat{E}(O|x_p)\right]}{\widehat{V}(O)} = 1 - \frac{\widehat{U}_{\sim p} - \widehat{E}^2(O)}{\widehat{V}_T(O)}$$

with the $\widehat{U}_{\sim p}$ factor:

$$\widehat{U}_{\sim p} = \frac{1}{r-1}\sum_{s=1}^{r} f(B_{s,*}) f(C^p_{s,*})$$

the same estimation of the expected value $\widehat{E}^2(O)$ as in equation (3.21) and the estimation of variance as:

$$\widehat{V}_T(O) = \frac{1}{r-1}\sum_{s=1}^{r} f^2(B_{s,*}) - \widehat{E}^2(O)$$

The number of model evaluations (N_{m_eval}) needed to compute all the first and total order sensitivities for all parameters correspond to the number of parametrizations in matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ and depends on the number of samples (r) and the number of parameters (q) with the relation $N_{m_eval} = r(q+2)$.

3.6.2.2 Parameter sampling for construction of sensitivity matrices

To generate the $\boldsymbol{A}$ and $\boldsymbol{B}$ matrices referred in section 3.6.2.1 (equations (3.18) and (3.19)), a sample D from a probability distribution must be made for each parameter x for each matrix. $\boldsymbol{A}$ and $\boldsymbol{B}$ are then constructed from concatenation of all these samples.

$$\boldsymbol{A} = \left[D_1^{'}, \quad D_2^{'}, \quad \cdots, \quad D_q^{'}\right], \quad \boldsymbol{B} = \left[D_1^{''}, \quad D_2^{''}, \quad \cdots, \quad D_q^{''}\right]$$

In this dissertation, three distributions were chosen as possibilities for this sampling, the reciprocal distribution:

$$D_p \sim Reciprocal(a_p, b_p)$$

where a_p and b_p are the lower and upper bound respectively of parameters p; the log normal distribution:

$$D_p \sim LogNormal(\mu, \sigma)$$

with μ as the mean and σ as the standard deviation; and the truncated Log normal distribution:

$$D_p \sim TruncatedLogNormal(\mu, \sigma, a_p, b_p)$$

with definitions as before. Examples of sampling 10000 parameters from each of these distributions can be found in figure 3.2.

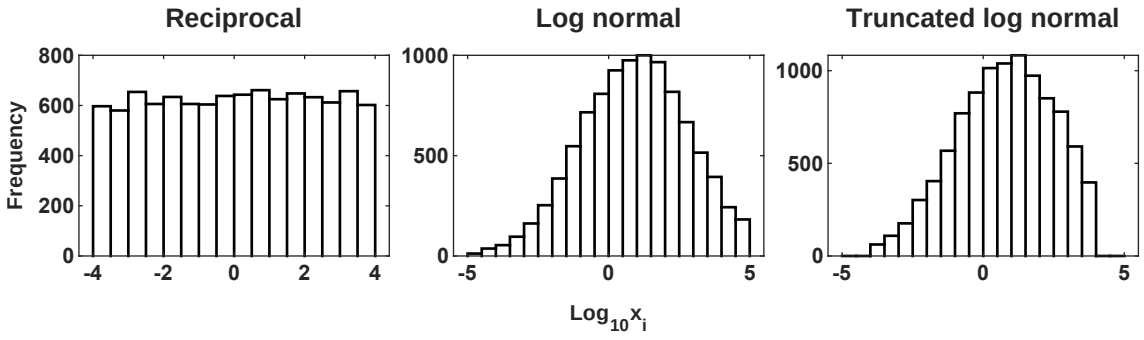


Figure 3.2: Distribution of 10000 parameters sampled of three different distributions, $Reciprocal(a, b)$ distribution with $a = -4$ and $b = 4$, $LogNormal(\mu, \sigma)$ distribution with $\mu = 1$ and $\sigma = 1$, and $TruncatedLogNormal(\mu, \sigma, a, b)$ distribution with $\mu = 1$, $\sigma = 1$, $a = -4$ and $b = 4$.

Note that the Log normal distributions are not centred at 0. The log normal can either be centred at the parameter value that minimizes the scoring function or at the center of the

parameter bounds.

3.6.2.3 Bootstrapping

Bootstrapping was used as a means of gauging the accuracy of the sensitivity indices, and to better compare different sample sizes and parameter regions.

It was performed as follows: first, matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ were re-sampled, such that the resulting matrices have, at each entry, one of the entries of the original matrix with equal probability:

$$\boldsymbol{A}^{\dagger} = \begin{bmatrix} A_{\lceil a_1 \rceil,*} \\ A_{\lceil a_2 \rceil,*} \\ \vdots \\ A_{\lceil a_r \rceil,*} \end{bmatrix} \quad \boldsymbol{B}^{\dagger} = \begin{bmatrix} B_{\lceil a_1 \rceil,*} \\ B_{\lceil a_2 \rceil,*} \\ \vdots \\ B_{\lceil a_r \rceil,*} \end{bmatrix} \quad \boldsymbol{C}^{\dagger} = \begin{bmatrix} C_{\lceil a_1 \rceil,*} \\ C_{\lceil a_2 \rceil,*} \\ \vdots \\ C_{\lceil a_r \rceil,*} \end{bmatrix}$$

where a is a value/sample form the distribution $\omega \sim U(0, r)$. This re-sampling process is repeated $\lceil \sqrt{r} \rceil$ times and these new matrices are used to calculated the sensitivity indices as described in section 3.6.2.1. These sensitivity indices are then used to calculate a bootstrapping mean and to generate boxplot graphs of the sensitivity index distributions. Note that the re-sampling must be made so that each row of the three re-sampled matrices corresponds to the same row from the three original matrices, to guarantee that a relation equivalent to equation (3.20) still exists between matrix $\boldsymbol{C}^{\dagger}$ and matrices $\boldsymbol{B}^{\dagger}$ and $\boldsymbol{A}^{\dagger}$. Larger sample sizes tend to produce smaller intervals for the bootstrapped sensitivity indices, which seem to converge around what can be assumed to be the true sensitivity value, for a given parameter region, sampling method, and sample size.

3.7 Profile likelihood analysis

The method presented here was implemented and is described based on the works of Venzon and Moolgavkar [420] and Royston [421].

The derivation of confidence intervals for parameters of non-linear models is a difficult proposition due to the complexity of deriving exact numerical formulas for them. Additionally, non-normally distributed parameters (with asymmetry and skewness) make traditional formulas inapplicable, as classic confidence intervals (Ci) depend on asymptotic properties that cannot be guaranteed. Moreover, computationally intractable problems can lead to challenges in determining if a parameter is identifiable or not.

Linear and non-linear least squares problems, including model parameter optimization, can be examined using maximum likelihood (ML) estimators. Assuming that the model residuals (i.e., the differences between the observed data and model predictions) are normally distributed, the ML function can be expressed as:

$$-2LL(y|\theta) = \sum_{i=1}^{n} \left(\frac{y_i - \hat{y}_i(\theta)}{\sigma_i} \right)^2 + c \tag{3.22}$$

where y refers to the data points collected from the experiment, σ to the standard deviation associated with these data points, $\hat{y}(\theta)$ to the points generated by the model under parameterization (θ), and c to a constant. The WSSR (see equation (3.10)) is identical to equation (3.22), apart from the constant, and so the maximum likelihood estimate (MLE) $\hat{\theta}$ is equivalent to the minimization of the WSSR [422]:

$$\hat{\theta} = arg \max_{\theta} LL(y|\theta) \equiv arg \min_{\theta} WSSR(y|\theta)$$

where $\arg\max_\theta$ represents the value of θ that maximizes the likelihood function $LL(y|\theta)$, and $\arg\min_\theta$ represents the value of θ that minimizes the weighted sum of squared residuals $WSSR(y|\theta)$.

This likelihood function allows for the construction of likelihood-based confidence regions (Cr). Crs offer several theoretical advantages, especially in cases involving non-linear models or non-normally distributed parameters. These advantages include the ability to capture asymmetry in the parameter estimates, more accurate behaviour as the sample size increases (better asymptotic properties), and their applicability even in cases where traditional methods struggle due to non-normality or other complications in the likelihood function. The formula for these confidence regions is:

$$Cr_\alpha(\theta|y) = \left[\theta \colon 2 \left\{ LL\left(\hat{\theta}\right) - LL\left(\theta\right) \right\} \leq \chi^2_{df;1-\alpha} \right]$$

where $\chi^2_{df;1-\alpha}$ represents $1-\alpha$ quantile of the χ^2 distribution with df degrees of freedom. This range includes all parameter values θ where the LL deviates by no more than $\frac{\chi^2_{df;1-\alpha}}{2}$ from the LL associated with the MLE.

However, in complex models with multiple parameters such as those tackled in this dissertation, the calculation of likelihood is still problematic due to interdependencies between multiple parameters, problems with identifiability, and the feasibility of computing a confidence region in a large dimensional space. Identifiability refers to whether distinct parameter values can be reliably inferred from the data, which is crucial for meaningful parameter estimation. PL helps evaluate parameter identifiability by isolating the effect of individual

parameters on the likelihood. In this case, we can resort to profile likelihood as described by Venzon and Moolgavkar [420], and Royston [421]. The PL function:

$$PL(\theta_j) = \underset{\theta_{j\neq i}}{max} LL(\theta|y)$$

simplifies the optimization by focusing on a single parameter of interest, θ_j, while treating the remaining parameters, $\theta_{j\neq i}$, as nuisance parameters that can be used to maximize the LL at each point of θ_j. By reducing the dimensionality of the problem, this method isolates the influence of the parameter θ_j, making it easier to evaluate its contribution to the overall likelihood.

Similar to the ML approach, the PL function enables the estimation of Crs based on likelihood values. These Crs are derived by comparing the likelihood at different parameter values to the maximum likelihood, adjusted by a chi-squared threshold that accounts for uncertainty in the estimates:

$$Cr_\alpha(\theta_j|y) = \left[\theta_j : 2\left\{LL\left(\hat{\theta}\right) - PL\left(\theta_j\right)\right\} \leq \chi^2_{df;1-\alpha}\right] \tag{3.23}$$

The PL-derived Crs are particularly useful because they accommodate asymmetry and can form discontinuous intervals, as demonstrated in figure 3.3. The PL is used to assess parameter identifiability and designate likely candidates for reduction.

To relate the confidence regions from equation (3.23) to the actual scores obtained from the model, the values of the PL and ML must be transformed into scores. So it can be written that:

$$-2LL\left(\hat{\theta}\right) = best\ score \tag{3.24}$$

where best score refers to the highest log-likelihood value obtained during parameter estimation, calculated using one of the score functions from section 3.4. And

$$-2PL\left(\theta_j\right) = score \tag{3.25}$$

where the score in equation (3.25) is obtained by performing local parameter estimations for all model parameters except the one for which profile likelihood analysis (PLA) is being performed. The score is calculated using one of the options from section 3.4. And that the confidence region threshold is given by:

$$-2LL\left(\hat{\theta}\right) + \chi^2_{df;1-\alpha} = best\ score + \chi^2_{df;1-\alpha} \tag{3.26}$$

For $\alpha = 0.05$ (corresponding to a 95% confidence level) and 1 degree of freedom, the threshold is

$$best\ score + \chi^2_{1;0.95} = best\ score + 3.841 \tag{3.27}$$

This threshold represents the boundary for the confidence region at a 95% confidence level, where any score within this range is considered statistically consistent with the best score.

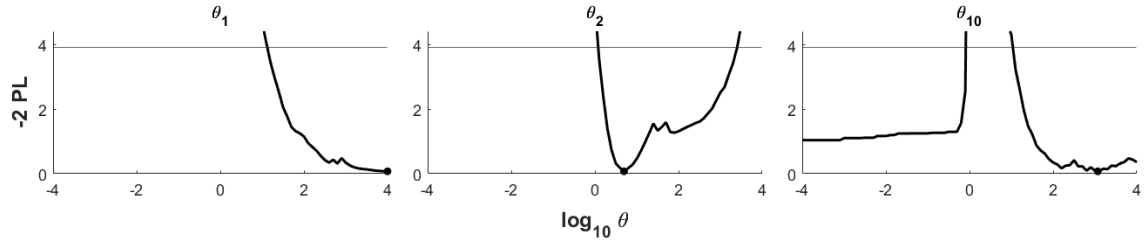


Figure 3.3: Examples of varied parameter profile likelihoods for LiREM. Each subplot shows the curves for a single parameter. The x-axis represents the log-transformed parameter values (log10 P) and the y-axis depicts the normalized total score (s_t). The confidence regions are delimited by the intersection of the thick and dashed black lines, with the region falling below the horizontal dashed threshold. Note that this region can be discontinuous. The thick black line represents $-2PL\left(\theta_j\right)$, which correspond to the scores being measured as shown in equation (3.25). The horizontal dashed line represents the critical threshold, calculated as $\chi^2_{1;0.95} - 2LL(\hat{\theta})$, derived from the maximum log-likelihood value ($\hat{\theta}$) obtained with the model score, adjusted by the 95th percentile of the χ^2 distribution, reflecting the 95% confidence level and 1 degree of freedom as shown in equations (3.26) and (3.27). The black point is $-2LL\left(\hat{\theta}\right)$, which represents the best log-likelihood score obtained during parameter estimation, as shown in equation (3.24).

4 Results and Discussion

This chapter presents the software developed for this dissertation. The software's capabilities are initially demonstrated using two example models specifically created for this demonstration, and subsequently applied to two biologically relevant models of increasing complexity.

The corresponding code for the main software [8, 9], and the two implemented models (TM-M [11], G-PIPK [10]) are available in their respective GitHub repositories.

4.1 Example models

The two example models created to demonstrate the software consist of very simple hypothetical biological pathways. The first includes only a ligand and a receptor and is thus called LiRMo. The second includes an effector after the receptor, mimicking the start of a biological cascade and receives the name LiREM. Both examples will be described in the following sections.

The files used for these two models can be accessed on the main software GitHub [8] inside the folder Matlab/Model/. They are the SBtab (described in section 3.2.1) file in .xls format, a settings file in .m format, and a folder containing the SBtab files parsed as .tsv format.

4.1.1 LiRMo

LiRMo is presented in this section as a basic model to serve as an example to illustrate the functionality of the software. It is designed to simulate a simple biological interaction involving five species and four reversible reactions. The model includes one input, which is the concentration of a hypothetical ligand, and three outputs that track the states of a hypothetical receptor. The receptor in this model can exist in active or inactive forms and can be bound or unbound to the ligand. The transitions between these states are the core focus of the model, and are detailed in the species list and reaction descriptions. Schematics of the model can be found in figure 4.1A.

All the components of the model can be consulted in the SBtab file but, for better comprehension, the most important ones will be listed in this section. The species utilized in this model are: the inactive receptor (iR), the active receptor (aR), the Ligand (L) that binds to the receptor in order to facilitate its activation, the inactive receptor bound to ligand (iRL),

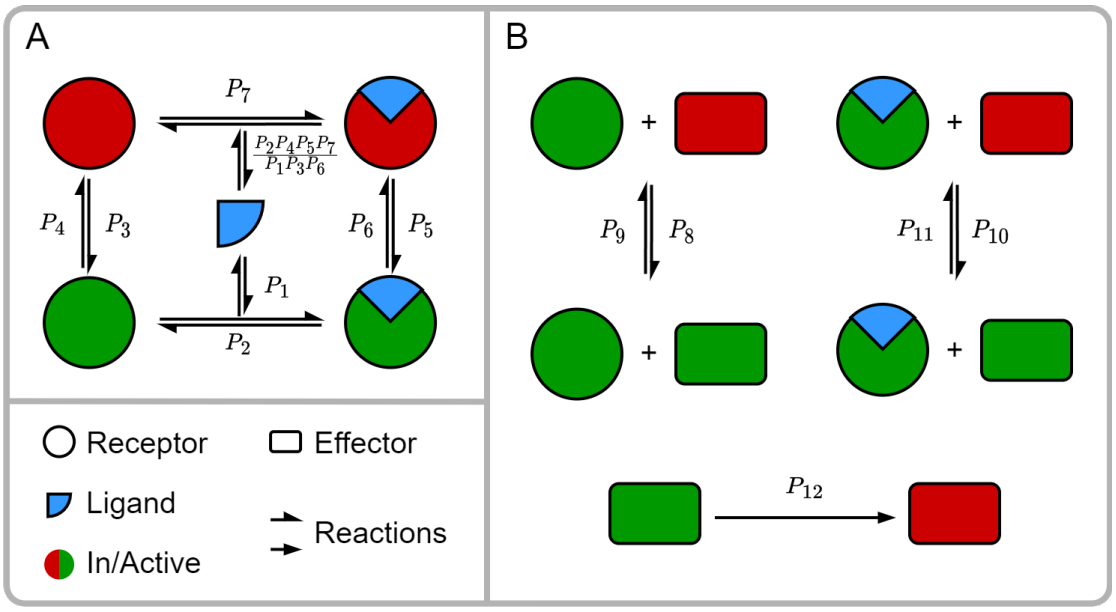


Figure 4.1: LiRMo and LiREM schematics. A – Schematic of LiRMo. B – Additional reactions that are included for LiRMo. $P_{1–12}$ – Model parameters, two per reversible reaction and one for the irreversible reaction. The parameter that defines the rate of separation of the ligand from the inactive receptor is calculated from parameters 1–7 in order to fulfil thermodynamic constraints.

and the active receptor bound to ligand (aRL).

As for the four reversible Rs used in this model, they have the following reaction formulas and reaction rates:

R_0 – Reaction formula: $aR + L \rightleftharpoons aRL$

Reaction rate: $kf_0 \times aR \times L - kr_0 \times aRL$

R_1 – Reaction formula: $iR \rightleftharpoons aR$

Reaction rate: $kf_1 \times iR - kr_1 \times aR$

R_2 – Reaction formula: $iRL \rightleftharpoons aRL$

Reaction rate: $kf_2 \times iRL - kr_2 \times aRL$

R_3 – Reaction formula: $iR + L \rightleftharpoons iRL$

Reaction rate: $kf_3 \times iR \times L - kr_3 \times iRL$

Despite there being 8 parameters in these reactions, only 7 of them are free parameters due to the thermodynamic constraints $\frac{kf_0}{kr_0} \times \frac{kf_1}{kr_1} = \frac{kf_2}{kr_2} \times \frac{kf_3}{kr_3}$ ensuring that one of the parameters must be dependent on the others, in this case parameter kr_3 is constrained to be $kr_3 = \frac{kr_0 \times kr_1 \times kf_2 \times kf_3}{kf_0 \times kf_1 \times kr_2}$. The names of the parameters used in the model file and in the reaction list can be linked to the parameters used in figure 4.1 and the analysis through table 4.1.

Furthermore, the model has one input:

- L – Represents the amount of the hypothetical ligand present in the model. Its concentration is kept at the value defined in the SBtab and it is not affected by the model reactions;

and three outputs:

- inactive receptor total amount (iRT), expressed as $iR + iRL$;
- active receptor total amount (aRT), expressed as $aR + aRL$;
- ratio of active receptor to total receptor amount (RaRTR), expressed as $\frac{aR+aRL}{aR+aRL+iR+iRL}$.

Finally, there are 5 Es defined to demonstrate the correct functioning of the model. In these experiments, all experimental outputs were measured and the input was varied in the following manner:

- E_0 – Input of Ligand at 1 nmol/L starting at 1 second and ending at 7 seconds;
- E_1 – Input of Ligand at 3 nmol/L starting at 1 second and ending at 6 seconds;
- E_2 – Input of Ligand at 7 nmol/L starting at 3 seconds and ending at 5 seconds;
- E_3 – Input of Ligand at 25 nmol/L starting at 3 seconds and ending at 4 seconds;
- E_4 – Input of Ligand following a sine curve, oscillating between 0 and 1 nmol/L.

4.1.2 LiREM

Building upon the foundational structure of LiRMo, this section introduces LiREM, a more intricate representation that extends the basic molecular interactions by incorporating an effector. The effector, integral to the signalling pathway, is modelled to exhibit both active and inactive states and is capable of inter-converting between these states spontaneously or under the influence of the receptor-ligand interactions. This model aims to emulate the initial stages of a biological cascade, presenting two additional species and five reactions to elucidate the dynamic processes involved, as compared to LiRMo.

The following aims to clarify the essential species and reactions, alongside the overall structure of the LiREM model, as depicted in the schematics provided in figure 4.1B. Similarly to LiRMo, the specifications of this model are documented in its SBtab file, and listed bellow are the main additions made to LiRMo. Two species were added, active effector (aE) and inactive effector (iE), and three new reactions were added, with the following reaction formulas and reaction rates:

R_4 – Reaction formula: $iE + aR \rightleftharpoons aE + aR$

Reaction rate: $kf_4 \times iE \times aR - kr_4 \times aE \times aR$

R_5 – Reaction formula: $iE + aRL \rightleftharpoons aE + aRL$

Reaction rate: $kf_5 \times iE \times aRL - kr_5 \times aE \times aRL$

R_6 – Reaction formula: $aE \rightleftharpoons iE$

Reaction rate: $kf_6 \times aE$

As before the names of the parameters used in the model file and in the reaction list can be linked to the parameters used in figure 4.1 and the analysis through table 4.1. Two new outputs were added to follow the behaviour of the newly added effector species:

- active effector total amount (aET), expressed as aE;
- inactive effector total amount (iET), expressed as iE.

while the output "RaRTR" was removed. The same 5 experiments shown in the previous section were used in order to demonstrate the correct functioning of the model.

4.1.3 Discussion

In this section, the structural and functional rationales underlying the development of the LiRMo and LiREM models will be discussed. It will explore the strategic decisions leading to the creation of two distinct models and the selection of specific structural elements. These choices were pivotal for enhancing instructional clarity, improving simulation efficacy, and ensuring the precise application of results to more complex biological systems.

The decision to develop two separate models, LiRMo and LiREM, rather than a single comprehensive framework, facilitates a detailed examination of individual components and provides clearer instructional demonstrations. By isolating LiRMo, the simpler of the two, explanations in section 4.2 are streamlined and more accessible, while LiREM is employed in section 4.3 to analyse more complex interactions. This approach allows each model to function optimally within its scope, enhancing the overall educational and analytical value of this dissertation.

The choice to incorporate a small cyclical structure in LiRMo was driven by its prevalence in natural biological systems. This configuration facilitates the application and testing of conservation laws, which are crucial for extending these principles to more complex models. This design not only mirrors real biological processes but also enhances the model's applicability and relevance as a foundation for more complex models.

Analysis/Figure	Model	Description
P_1	kf_0	Ligand binds to active receptor
P_2	kr_0	Ligand unbinds from active receptor
P_3	kf_1	Inactive receptor converts to active receptor
P_4	kr_1	Active receptor converts to inactive receptor
P_5	kf_2	Inactive receptor converts to active receptor while bound to the ligand
P_6	kr_2	Active receptor converts to inactive receptor while bound to the ligand
P_7	kf_3	Ligand binds to inactive receptor
$\frac{P_2 \times P_4 \times P_5 \times P_7}{P_1 \times P_3 \times P_6}$	kr_3	Ligand unbinds from inactive receptor
P_8	kf_4	Inactive effector turns into active effector catalysed by active receptor
P_9	kr_4	Active effector turns into inactive effector catalysed by active receptor
P_{10}	kf_5	Inactive effector turns into active effector catalysed by active receptor bound to ligand
P_{11}	kr_5	Active effector turns into inactive effector catalysed by active receptor bound to ligand
P_{12}	kf_6	Active receptor turns into inactive receptor

Table 4.1: LiRMo and LiREM parameters. The table shows the mapping between different parameter notations: Analysis/Figure: P_{1-7} represent the parameters used in model analysis and in figure 4.1 for LiRMo, and P_{8-12} are the added parameters for LiREM. Model: kf_X and kr_X correspond to the rate constants in the SBtab model file, where X is the reaction number (1–6). Description: briefly explains the biological meaning of each parameter. f indicates forward reactions, while r denotes reverse reactions.

In LiREM, the activated receptor is designed to function as a catalyst rather than directly interacting with the effector. This design choice ensures that the new reactions introduced do not interfere with the parameters established in LiRMo. This separation maintains the applicability of parameters derived from the simpler model, ensuring their relevance and consistency when integrated into the larger, more complex framework.

4.2 The modelling system

4.2.1 Model initialization into MATLAB

MATLAB code was developed in order to successfully parse the SBtab file and the settings file, generating MATLAB files that are usable for all of the developed software analysis. This code creates the following files:

- SBtab_LiRMo – Contains the SBtab file stored as a table (.mat file);
- data_LiRMo – Contains three MATLAB structs (.mat file):
  - sb – contains the SBtab in struct format;
  - data – contains, for each experiment, the time when the measurements were performed, the measured output, and the measured standard deviation;
  - sbtab – contains all the relevant properties of the model species, the default parameters, inputs and outputs to the model;
- model_LiRMo – Main model file, contains all components common to all the experiments defined in the SBtab: species, reactions, default set of parameters, and compartments. It does not contain any inputs (.mat, .sbproj, and .xml files);
- Model_"eq_ / _ / detail_"LiRMo_"experiment number" - Three files per experiment of the model: one each for equilibration, simulation and detailed simulation, respectively. The models for simulation and detailed simulation include the inputs to the system necessary for proper simulation and the set of outputs to be measured out of the system (.mat files);
- Input_LiRMo – Contains a table with the input for experiments where it changes a significant number of times (.mat file);
- LiRMo_input_creator - Function that creates the previous file (.m file)
- LiRMo_input"experiment number"_Ligand - Function called at run time by the code for experiments where the inputs of the model change a significant number of times. It is used to improve performance as further explained in section 4.2.4 (.m file).

Additional information about the organization of these files and the working of the import scripts can be found on the on-line code documentation [397].

```
1 Run_main("LiRMo",8,"LiRMo.m");
```

```
The model folder chosen was: LiRMo
The analysis option chosen was: Import model files
The settings file chosen was: LiRMo.m
Generating model files and folder from SBtab
SBtab with 24 sheets parsed successfully
Model files and folders generated successfully
Starting Import model files
Import model files completed successfully
```

Figure 4.2: LiRMo MATLAB import code. Necessary code and relevant output when importing a model into MATLAB demonstrated using LiRMo. The Run_main function takes 3 arguments: the name of the model folder, LiRMo in this case; the selection of the analysis to perform, 8 represents the import code; and the name of the settings file used, LiRMo.m in this example.

As a prerequisite to use this code, the user must have the model structured in a specific manner: a folder with a meaningful name containing the SBtab file in its root and as many settings files as needed in a folder called "settings". After this is prepared, the code is executed in a single line, running the function Run_main using the following parameters, in order: the name of the folder where the model is saved, the analysis that should be performed (hard-coded as analysis number 8 in the case of import functions), and the name of the settings file that should be used, an example of such code being run is presented in figure 4.2.

This initialization code has been written specifically for this dissertation and, as such, its correct implementation must be demonstrated. In order to achieve this, manual checks of the MATLAB model were conducted comparing the various core components of the model, species (table 4.2 and figure 4.3), reactions (table B.1 and figure B.1), and parameters (table B.2 and figure B.2). In the following sections, the model imported here (LiRMo) will be used for other demonstrations and these will also serve as a validation for correct import of the experiments, experimental inputs, and outputs (figure 4.5).

4.2.2 Model simulation

4.2.2.1 MATLAB simulation environment

The MATLAB function used for model simulation is called sbiosimulate [423]. In order to use this function, two inputs were provided: a SimBiology model [404, 424], and a con-

!ID	!Name	!Unit	!IsConstant	!Location
S0	iR	nanomolarity	FALSE	Compartment
S1	aR	nanomolarity	FALSE	Compartment
S2	iRL	nanomolarity	FALSE	Compartment
S3	aRL	nanomolarity	FALSE	Compartment
S4	L	nanomolarity	TRUE	Compartment
S5	iE	nanomolarity	FALSE	Compartment
S6	aE	nanomolarity	FALSE	Compartment

Table 4.2: LiRMo and LiREM species as present in the SBtab. Each row defines a unique species with its identification (ID), name, unit, constancy state, and compartmental location. species (S)0-4 are common to both models, representing the core ligand-receptor interaction components. S5-6 are exclusive to LiREM, illustrating its extended scope to include effector molecules. The 'IsConstant' field denotes whether a species' concentration remains fixed during simulations, with only the Ligand (L) set as constant. Some columns have been omitted for brevity.

```
1 load(pwd + "\Matlab\Model\LiRMo\Matlab\Data\model_LiRMo.mat")
2 disp(modelobj.Species)
```

```
SimBiology Species Array

  Index:    Compartment:    Name:    Value:    Units:
  1         Compartment     iR       2         nanomolarity
  2         Compartment     aR       2         nanomolarity
  3         Compartment     iRL      0         nanomolarity
  4         Compartment     aRL      0         nanomolarity
  5         Compartment     L        0         nanomolarity
```

Figure 4.3: Species of the LiRMo model as displayed in MATLAB SimBiology toolbox after import from SBtab. The output shows the index, compartment, name, initial value, and units for each species. Note the initial conditions: iR and aR are set to 2 nanomolar, while iRL, aRL, and L are initialized at 0. This representation demonstrates the successful translation of species information and initial conditions during the import process, setting the stage for model simulations.

figuration set object [425]. Its output is a MATLAB simdata object [426] with the simulated results.

The configuration set object contains options that are used by sbiosimulate during model simulation. Only a subset of the possible options provided in this configuration set object were selected for the simulations described in this dissertation. These are:

- MaximumWallClock [427] – Maximum time to run a single simulation on the CPU;
- TimeUnits [428] – Time units of the simulation;
- StopTime [429] – End time of the simulation in the units defined in "TimeUnits";
- DimensionalAnalysis [430] – Boolean that toggles dimensional analysis;
- UnitConversion [431] – Boolean that toggles unit conversion;
- OutputTimes [432] – Timepoints in which the outputs of the model should be logged, these usually correspond to the timepoints for which data was provided;
- MaxStep [433] – Maximum size of the time step used in the simulation;
- RelativeTolerance [434] – Value of the relative tolerance used in the simulation;
- AbsoluteTolerance [435] – Value of the absolute tolerance used in the simulation;
- AbsoluteToleranceScaling [436] – Boolean that toggles absolute tolerance scaling;
- AbsoluteToleranceStepSize [437] – Value of the absolute tolerance step size used in the simulation.

The choice of "RelativeTolerance", "AbsoluteTolerance", "AbsoluteToleranceScaling", and "AbsoluteToleranceStepSize", was made by taking into account the amount of precision needed and the trade-off between more precision and longer simulation time. For more information on this topic, please refer to the MATLAB documentation [438]. In addition to the options listed above, the solver used for solving the ODEs is the default one of SimBiology, *ode15s* [439].

The simdata object contains the results of the simulation run and metadata relating to the options chosen in the configuration set object. Of particular relevance are the *Time* and *Data* arrays stored in this object. They contain all the time points and species values, respectively, for which results were logged during the simulation run. These time series data can then be used to compare the simulation results with the the experimental (or simulated) data.

4.2.2.2 Model equilibration

The model equilibration step is important to ensure that the model is in a steady state prior to the introduction of a perturbation. True steady states do not exist, but it is assumed that a biological system is maintaining a quasi steady state prior to any perturbation. This is reproduced in the simulations by allowing them to run for a set equilibration time, after which the initial values of the model species are set to the final state of the species at the end of the equilibration period. Although there are ways to determine if a model will enter a steady state, and how long it will take to reach it, such methods were not implemented and care should be taken when manually choosing an equilibration time.

Figure 4.4 demonstrates LiRMo run without any equilibration step and with equilibration steps of subsequently larger times. In this case, a reasonable equilibration time would be approximately one second.

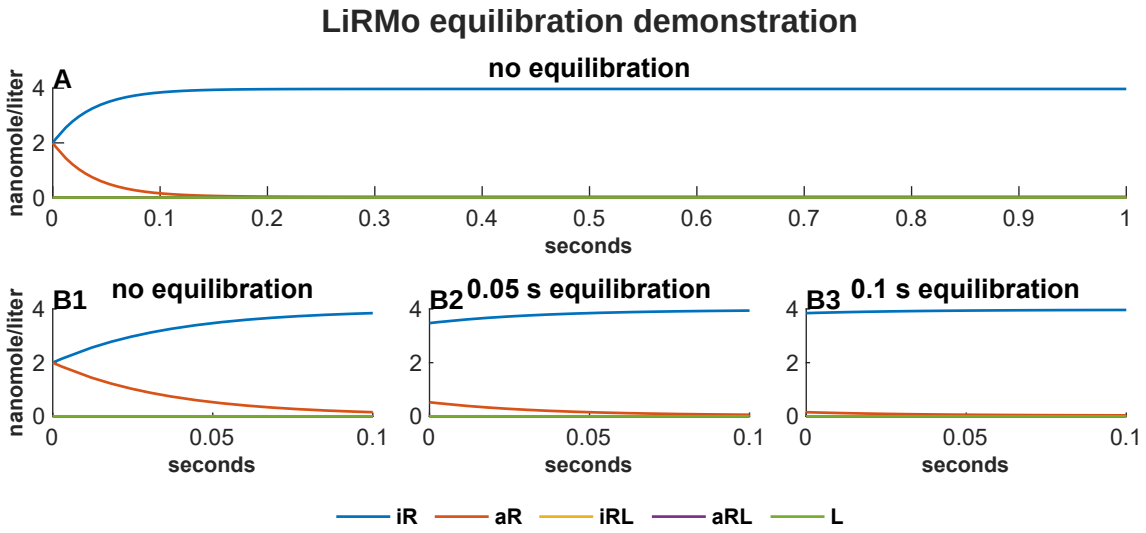


Figure 4.4: LiRMo equilibration demonstration. A – One-second simulation of LiRMo, which in this case is sufficient to achieve an approximated model steady state or equilibrated state. B(1–3) – Effect of different equilibration times: B1 with no equilibration, B2 after a brief 0.05-second simulation, and B3 following a 0.1-second simulation. Each subsequent run represents a model progressively closer to a steady state at the start. Note: in practical applications, significantly longer equilibration durations, such as 100 or 1000 seconds, are typically preferred to ensure maximal approximation to a steady state.

4.2.2.3 Model inputs and outputs

A useful model should be able to be simulated with the same θ in the conditions of various experiments. For this reason, the model must be repeatedly tested with different inputs and outputs, and it must also be validated. In this section, different experimental scenarios will be compared, specified in the SBtab as five experiments, and demonstrated that they are all

correctly simulated.

The inputs given to the model at run time were defined in two different ways depending on the complexity of the signal. Simple inputs that change values only a few times during the simulation were presented as events: time-value pairs representing the change of input that should occur at a specific time. Complex inputs that continuously change value during the simulation were represented as functions: these functions, at each time point of the simulation, either calculate the correct input or interpolate from a table of values the best one for that specific time point. Due to performance constraints, it is preferred to use a function when there are more than approximately one hundred events for a given experiment.

The measured outputs from the model can have various forms: they can be any set of model species or input and the result of any operation done on any model species or input, as long as these operations result in positive real values.

Figure 4.5 represents all the experiments present in LiRMo SBtab. These give examples of inputs represented as events (A), in this case, the introduction of a ligand at specific times or at specific concentrations; and inputs represented as functions (B), in this case, the introduction of the ligand as a continuous sine function. For the outputs, we can see two examples of an output constructed as the sum of two species, the blue and red lines representing the total amount of inactive and active receptor, respectively, and one example of an output constructed as a ratio of the sum of various species, the green line representing the ratio between active and inactive receptor.

4.2.2.4 Model parameter substitution

In the previous section, it was shown that the model could be correctly simulated in different experiments with the same θ. In this section, it is demonstrated that parameter changes are also correctly accounted for. θs for the model can be set manually in the settings file, there can be multiple θs specified at the same time and all the θs specified are run when the model is executed. Figure 4.6 exemplifies four different θs executed in one of the LiRMo experiments, for which the parameters used in the settings file are specified in table 4.3. Automatic parameter substitution will be demonstrated in further sections, as well as all the analysis software needed to estimate improved parameters.

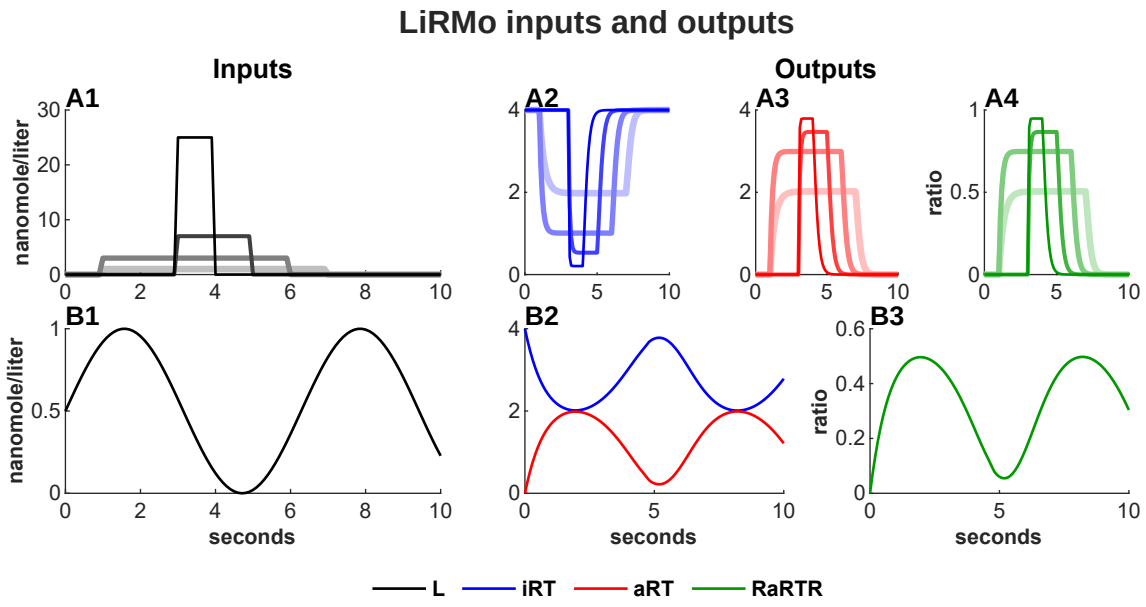


Figure 4.5: LiRMo inputs and outputs. Each plot represents one experiment specified in the SBtab, showing how the model responds to a varied array of inputs. A – Plots the input (A1) and output (A2–4) of experiments 0 to 3, the curves are the lightest for experiment 0 and get progressively more saturated up to experiment 3. B – Represents the input (B1) and output (B2–3) of experiment 4. Experiments 0–3 were used for optimization while experiment 4 is presented to exemplify more exotic inputs.

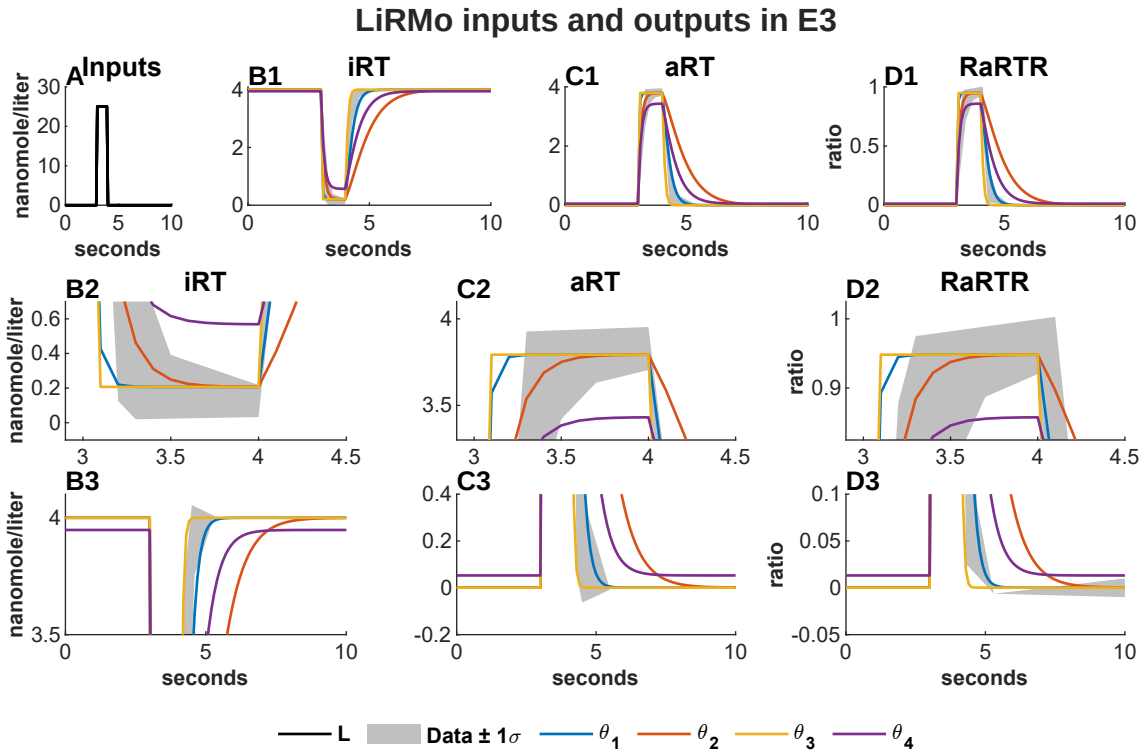


Figure 4.6: LiRMo inputs and outputs with focus on E_3. A – Concentration of input over time. B to D – Concentration and ratio for the outputs measured after running the model with θ_{1-4}, detailed in table 4.3. 1 – Full curves. 2 and 3 – Zoom into areas of interest so standard deviation is visible.

Parameter	θ_1	θ_2	θ_3	θ_4	Unit (log_{10})
P_1	4.36	3.86	4.86	2.18	$\frac{1}{nMolarity \times second}$
P_2	0.60	0.10	1.10	0.30	$\frac{1}{second}$
P_3	-2.94	-3.44	-2.44	-1.47	$\frac{1}{second}$
P_4	0.81	0.31	1.31	0.40	$\frac{1}{second}$
P_5	1.96	1.46	2.46	0.98	$\frac{1}{second}$
P_6	0.17	-0.33	0.67	0.08	$\frac{1}{second}$
P_7	3.37	2.87	3.87	1.68	$\frac{1}{nMolarity \times second}$
$\frac{P_2 P_4 P_5 P_7}{P_1 P_3 P_6}$	5.15	4.65	5.65	2.57	$\frac{1}{second}$

Table 4.3: LiRMo θ used in figure 4.6. There are 7 independent parameters (P_{1-7}) and one dependent parameter that is computed from the others. Parameter values are in logarithmic scale of base 10. A visual representation of the position of these parameters in the model can be seen in figure 4.1. θ_1 represents an optimized parameter set and θ_{2-4} were obtained by slightly modifying θ_1.

4.2.3 Model parameter set scoring

In order to compare output curves produced by different θs, it is important to have a reliable scoring mechanism (even if manually choosing between two different θ). When comparing two similar outputs given by two different θ as seen in figure 4.6, for example, θ_2 and θ_4, it is not immediately apparent which one produces the best fit for the data provided. θ_2 produces a better fit for the basal levels, while θ_4 produces a better fit for the ascending and descending phases of the transient response to an external input. In this case, the scores presented in table 4.4, calculated using the formulas from section 3.4, inform us that θ_4 is indeed a better fit than θ_2, while θ_3 is the best of the four. This process is useful for manually tuning a model but it becomes indispensable for automatic parameter estimation, as only after assessing which of two curves better fits the model data, can the algorithmic parameter optimization process begin.

Score	θ_1	θ_2	θ_3	θ_4
iRT	0.39	468.69	0.12	72.71
aRT	0.16	278.11	0.14	47.49
RaRTR	0.10	37.53	0.17	5.94
E_3	0.66	784.32	0.43	126.13

Table 4.4: LiRMo scores in figure 4.6 given θ from table 4.3. The scores are calculated across different experimental outputs and also consolidated for the total of all the scores of E_3. Lower scores indicate a better fit to model data. For a visual representation of this experiment, please refer to figure 4.6.

4.2.4 Discussion

This discussion addresses the key decisions made in the development and implementation of the modelling system. These include the choices made regarding the use of MATLAB, the import and initialization process, the simulation methods, the approach to equilibration, the handling of model inputs, parameter substitution, and scoring methods. These decisions were crucial in ensuring the robustness, flexibility, and accuracy of the modelling system.

MATLAB was chosen for this project due to its extensive capabilities in numerical computing, built-in toolboxes and ready-made optimization tools. Its SimBiology toolbox is particularly suited for systems biology applications, providing a comprehensive set of tools for model building, simulation, and analysis. Additionally, MATLAB built-in functions and toolboxes allow for seamless integration of various optimization functions and easy code parallelization, which is essential for this project. While MATLAB is a proprietary environment, which can be a downside in terms of cost and accessibility, it offers robust support and documentation that can significantly accelerate development. Furthermore, this dissertation open-source software can now serve as a base template to build increasingly powerful tools, either by expanding the code in MATLAB or by porting it to open-source languages such as Python or R.

The import and initialization procedures were designed to be modular and scalable, allowing for easy expansion and adaptation of the system. The use of structured files and specific folders ensures that the models are organized and that all necessary components are readily accessible. This organization not only facilitates the current analysis but also makes it straightforward to incorporate additional models or expand existing ones in the fu-

ture. For instance, new species, reactions, or experimental conditions can be added to a model with minimal modifications to a specific SBtab file. Additionally, this standardization makes it simpler to debug a model after import, as well as to develop new analysis like the ones demonstrated in section 4.3. During the development of this project, six different models were translated into the files needed to run in this code, and now it can take as little as a few weeks to perform this translation for other published models.

The sbiosimulate function in MATLAB was selected for its efficiency and integration with the SimBiology toolbox. It allows for the simulation of complex biological models with a high degree of precision and control over various simulation parameters. Performance considerations are critical, especially when dealing with large datasets or complex models. In order to achieve the best performance, this function was combined with sbioaccelerate, which can enhance performance by compiling models into optimized C code, guaranteeing the flexibility and speed needed for this dissertation.

Equilibration is essential to ensure that the model reaches a steady state before any perturbations are introduced. This mimics the natural state of biological systems, which are typically in a quasi steady state before experimental manipulations. Determining the best equilibration time involves running preliminary simulations and analysing when the model's outputs stabilize for some set of reasonable θ. Alternative methods, such as using analytical solutions or more sophisticated numerical techniques, could serve as alternative methods to find a steady state. However, these methods often come with a significant performance trade-off that made them not feasible a hypothesis within the scope of this project.

The need for two different methods to handle model inputs arises from the varying complexity of the input signals. Simple inputs, which change infrequently, can be efficiently managed as events in MATLAB. In contrast, complex inputs that change continuously require function-based representations to ensure smooth and accurate simulations without the extreme performance degradation associated with a high number of events. This dual approach ensures that the model can handle a wide range of experimental conditions without compromising performance. Other challenges in input handling include ensuring that the simulation accurately reflects rapid input changes and does not overshoot or lag behind the intended values due to inadequate time steps. As such, adequate time steps and integration tolerances are also another aspect that one must be attentive to when preparing a model for analysis.

The ability to rapidly and dynamically substitute parameters is crucial for testing different hypotheses and optimizing model performance. Without this capability it would be impracti-

cal to perform most model analyses. Currently, the software is only capable of exchanging model parameters chosen based on their biological significance and their impact on the model's behaviour. Future expansions could include initial amounts of species as adjustable variables, allowing for even more flexible and detailed models.

The scoring methods used to evaluate model performance were chosen for their simplicity and effectiveness in comparing different parameter sets. These methods provide clear metrics that can guide manual tuning and automated optimization processes. Alternative scoring methods, such as those based on more complex statistical analyses or machine learning algorithms, could offer deeper insights but were not implemented due to their greater complexity, which was deemed as unnecessary for the goals of this project. Nonetheless, the system is designed to be extensible, allowing for the easy integration of additional scoring methods, as needed.

4.3 The model analysis software

The capability of importing, simulating, and scoring a model was demonstrated in section 4.2. In this section, these capabilities are going to be built upon for more complex tools:

- Diagnostics, a simple way to verify that the model is being run appropriately, confirming how well different θs behave, and for a more nuanced exploration of model outputs;
- Parameter estimation to find a good θ fitting the provided data;
- Sensitivity analysis to explore how variations in θ affect data fitting;
- Profile likelihood analysis to explore the viability landscape of the parameters in θ, taking into account the available data.

4.3.1 Creation of dummy data

In order to apply the analysis tools that will be described in the following sections, it was necessary to provide data to the model. For this purpose, dummy data had to be generated and this section explains that process. The generated curves used equations similar to the ones used by Jensen *et al*. [440].

The dummy data was meant to emulate noisy data that could have been collected in a hypothetical scientific experiment. In order to do this, response curves for x species of the model were created as a response to a number of timed stimuli. The data created for this task consists of concentration-time curves that start at 0 concentration, have an exponential

change into a specific value, and an exponential change back to 0. The times of the increase and decrease, the final value of the increase, and the shape of the curves are determined by equations 4.6 to 4.9.

Uncertainty was introduced at the beginning and end of the stimulus response, for the shape of the response curve, and for its amplitude. To create this uncertainty, a certain number of curves were generated with similar starting conditions. After their generation, a mean and standard deviation were computed and used to feed the parameter estimation algorithm.

In order to generate the curves, 5 different distributions (Ds) (equations 4.1 to 4.5) were used:

$$D\alpha \sim TruncatedNormal(\overline{A}, A\sigma, 0, maxA) \tag{4.1}$$

to determine the amplitude of the change,

$$D\beta \sim TruncatedNormal(\overline{start_d}, start_d\sigma, 0, end - start) \tag{4.2}$$

to determine the delay for the start of the first exponential,

$$D\gamma \sim TruncatedNormal(\overline{start_s}, start_s\sigma, 0, inf) \tag{4.3}$$

to determine the shape of the first exponential,

$$D\epsilon \sim TruncatedNormal(\overline{end_d}, end_d\sigma, 0, E_t - end) \tag{4.4}$$

to determine the shape of the second exponential, and

$$D\zeta \sim TruncatedNormal(\overline{end_s}, end_s\sigma, 0, inf) \tag{4.5}$$

to determine the delay for the start of the second exponential.

These are truncated normal distributions, in which the first element represents the mean of the distribution, the second represents the standard deviation, the third represents the lower bound of the truncation, and the fourth represents the upper bound of the truncation.

The time series is constructed by combining two exponential curves. The values start at 0 until the beginning of the first exponential, then they rise according to the exponential equation scaled so that the maximum change possible is $X\alpha_i$ until the start of the second exponential. Here, the second exponential is introduced to reduce the change back to 0. The shape of the first exponential curve is determined by the "rise" equation. $start$ represents the default starting time of the exponential. $X\beta_i$ can alter this starting time from this position

until the default beginning of the second exponential curve. $X\gamma_i$ shapes the curve, with higher values leading to a shallower exponential and lower values to a steeper one. The shape of the second exponential is determined by the "fall" equation and it is constructed in a similar manner. Its starting time is given by end + $X\epsilon_i$ and its shape influenced by $X\zeta_i$. X_i are samples from the distributions shown in equations 4.1 to 4.5.

This is also described in equations (4.6) and (4.7).

$$\begin{aligned} rise &= \left(1 - e^{\frac{-t+start+X\beta_i}{X\gamma_i}}\right) \\ fall &= \left(1 - e^{\frac{-t+end+X\epsilon_i}{X\zeta_i}}\right) \end{aligned} \tag{4.6}$$

$$\begin{cases} y_{t,i} = 0 & 0 \leq t < start + X\beta_i \\ y_{t,i} = X\alpha_i * rise & start + X\beta_i \leq t < end + X\epsilon_i \\ y_{t,i} = X\alpha_i * rise - X\alpha_i * rise * fall & end + X\epsilon_i \leq t < E_t \end{cases} \tag{4.7}$$

After the time series are generated, we can obtain the mean and standard deviation that will be used for parameter estimation as described in equations (4.8) and (4.9). A small value U is used to insert extra uncertainty in the final values used for parameter estimation.

$$\overline{y_t} = \frac{\sum_i^I (y_{t,i})}{I} \tag{4.8}$$

$$y_t\sigma = max\left(U, \sqrt{\frac{\sum_i^I (y_{t,i} - \overline{y_t})^2}{I}}\right) \; U = 0.001 \tag{4.9}$$

This process was used to create dummy data for both example models. For LiRMo, dummy inputs from table 4.5 were used in order to generate the dummy data time series seen in figure 4.7 later used for parameter estimation. For LiREM, the inputs can be seen in table B.3 and the dummy data time series in figure B.6.

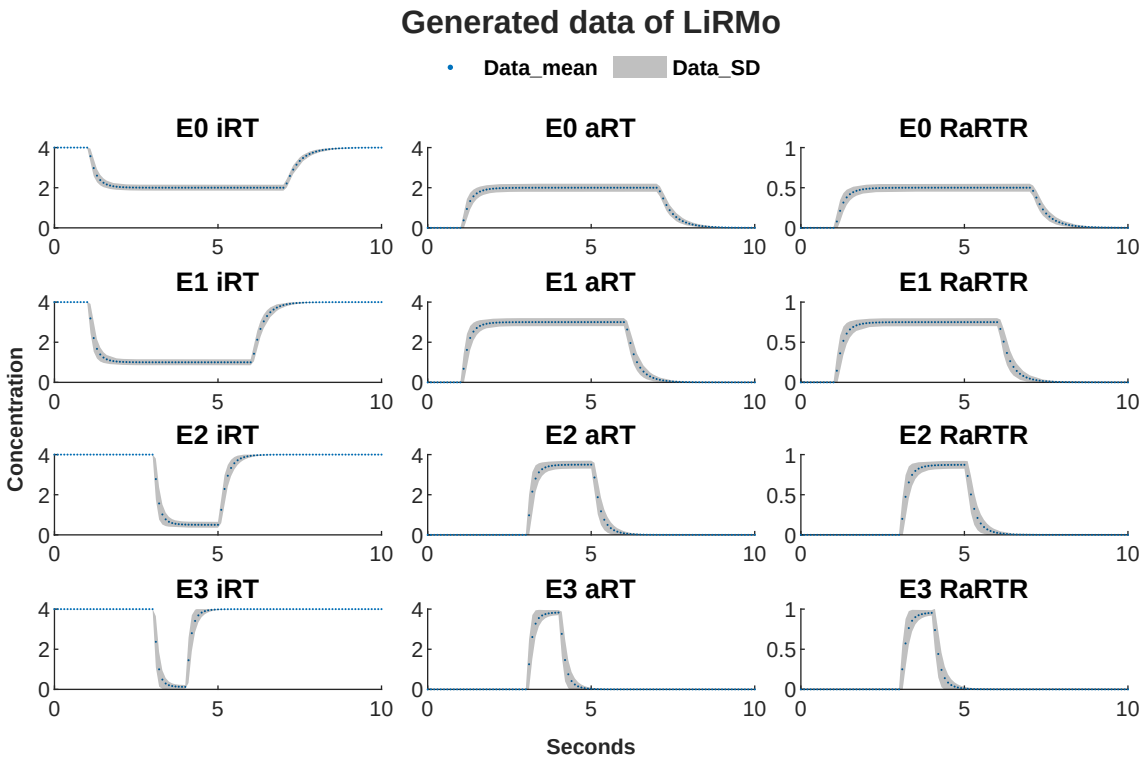


Figure 4.7: LiRMo dummy data. Experiments 0 through 3 are denoted by E_{0-3}. Each experiment has three outputs, inactive receptor total amount (iRT), active receptor total amount (aRT), and ratio of active receptor to total receptor amount (RaRTR). The blue points and the grey area represent the mean and the area encapsulated within one standard deviation of the mean of the generated data, respectively. Together, this data enables the evaluation of the model's performance under varying experimental conditions.

$Name$	$maxA$	$\overline{A}$	$A\sigma$	$start$	$\overline{start_d}$	$start_d\sigma$	$\overline{start_s}$	$start_s\sigma$	end	$\overline{end_d}$	$end_d\sigma$	$\overline{end_s}$	$end_s\sigma$	E_t	N
E0 iRT	-4	-2.000	0.15	1.00	0.00	0.08	0.20	0.08	7.00	0.00	0.08	0.40	0.08	10	100
E0 aRT	4	2.000	0.20	1.00	0.00	0.10	0.20	0.10	7.00	0.00	0.10	0.40	0.10	10	100
E0 RaRTR	1	0.500	0.05	1.00	0.00	0.12	0.20	0.12	7.00	0.00	0.12	0.40	0.12	10	100
E1 iRT	-4	-3.000	0.15	1.00	0.00	0.08	0.15	0.08	6.00	0.00	0.08	0.30	0.08	10	100
E1 aRT	4	3.000	0.20	1.00	0.00	0.10	0.15	0.10	6.00	0.00	0.10	0.30	0.10	10	100
E1 RaRTR	1	0.750	0.05	1.00	0.00	0.12	0.15	0.12	6.00	0.00	0.12	0.30	0.12	10	100
E2 iRT	-4	-3.500	0.15	3.00	0.00	0.08	0.10	0.08	5.00	0.00	0.08	0.20	0.08	10	100
E2 aRT	4	3.500	0.20	3.00	0.00	0.10	0.10	0.10	5.00	0.00	0.10	0.20	0.10	10	100
E2 RaRTR	1	0.875	0.05	3.00	0.00	0.12	0.10	0.12	5.00	0.00	0.12	0.20	0.12	10	100
E3 iRT	-4	-4.000	0.15	3.00	0.00	0.08	0.05	0.08	4.00	0.00	0.08	0.10	0.08	10	100
E3 aRT	4	4.000	0.20	3.00	0.00	0.10	0.05	0.10	4.00	0.00	0.10	0.10	0.10	10	100
E3 RaRTR	1	1.000	0.05	3.00	0.00	0.12	0.05	0.12	4.00	0.00	0.12	0.10	0.12	10	100

Table 4.5: LiRMo dummy data generating parameters. Parameters used for 4 experiments (E_{0-3}) for the three outputs, iRT, aRT, and RaRTR. These parameters determine the characteristics of the generated data, including the maximum amplitude change ($maxA$), mean amplitude change ($\overline{A}$), standard deviation of the amplitude change($A\sigma$), default start and end times($start$,end), mean delays at the start and end times($\overline{start_d}$,$\overline{end_d}$) and their corresponding standard deviations($start_d\sigma$,$end_d\sigma$). Additionally, the mean of the shape parameters of the start and end times($\overline{start_s}$,$\overline{end_s}$) and their standard deviations are also listed($start_s\sigma$,$end_s\sigma$). The table also specifies the total duration of each experiment(E_t) and the number of data points collected (N).

4.3.2 Diagnostics

The process of running all the experiments from the SBtab with the parameters given in the settings file was automated in order to provide an easy way to inspect the models.

This diagnostic tool provides various useful plots:

- Scores of all outputs for all experiments for each test θ. This is useful to visualize which θ is a better overall fit to the data and also to verify if some θs fit some parts of the model better than others. Example in figure 4.9 using LiRMo with the best θs obtained in section 4.3.4;
- Inputs to the model, allowing a visual inspection of the input data. Examples are shown in figures B.3 and B.7 for LiRMo and LiREM, respectively;
- Outputs of the model given test θs, which allow the user to assess if the outputs being measured are good fits to the provided data. Examples are shown in figure B.4 for LiRMo and in figures B.8 and B.9 for LiREM;
- Plots of each experiment with all relevant inputs and outputs given test θs. These have all the same data as the input and output plots but organized in a different way, allowing the user to analyse specific experiments individually. Examples can be seen in figures 4.8 and B.5 for LiRMo.

4.3.3 MATLAB Optimization Toolbox

The MATLAB Optimization Toolbox was used as the basis for parameter estimation. A wrapper function was built around a few MATLAB built-in functions in order to minimize the score discussed in section 3.4. These wrapper functions are modular and can be extended in the future. The MATLAB functions used can generally be divided into local and global optimization solvers. Two different local optimization solvers and four different global ones were adapted to be used in this code. The parameter estimation presented in the results sections was performed either with the *Genetic algorithm* (section 4.3.3.4) or *Particle swarm* (section 4.3.3.5) optimization solvers unless otherwise stated.

All the MATLAB solvers used take as input the objective function, an initial θ, a lower and upper bound for the parameters, and an *optimoptions* MATLAB object, with a few additional optimization options specific to each of the solvers. Some of them also receive a starting θ, or the number of parameters as initial inputs.

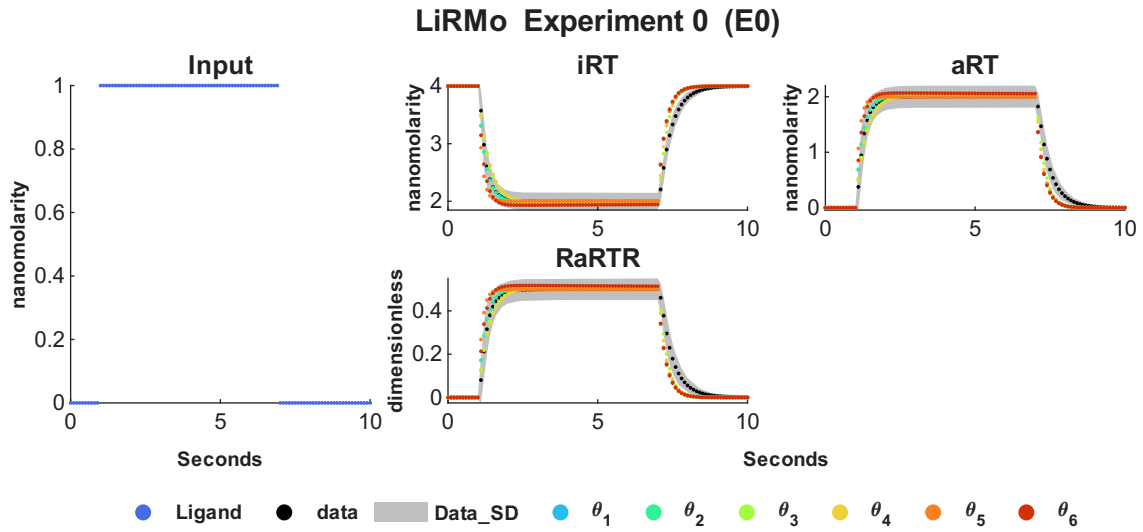


Figure 4.8: LiRMo E_0 diagnostics plot. Shows the input, output data with standard deviation, and θ_{1-6} outputs for LiRMo E_0. Input: 6-second impulse of Ligand (1×10 nMol/L) from 1 to 7 seconds. Outputs: inactive receptor total amount (iRT), active receptor total amount (aRT), and ratio of active receptor to total receptor amount (RaRTR). θ_{1-6} represent optimization results (figure 4.10), using: 1 – Genetic algorithm, 2 – fmincon, 3 – Simulated annealing, 4 – Pattern search, 5 – Particle swarm, 6 – Surrogate optimization. Output data identical to figure 4.7.

The following values for *optimoptions* are common between the solvers *Pattern search*, *Genetic algorithm*, *Particle swarm* and *Surrogate optimization*:

- MaxTime – Maximum run time for the algorithm;
- UseParallel – Option to run the solver in parallel to take advantage of multiple processing cores.

4.3.3.1 fmincon

This solver is based on the simulated annealing algorithm described in section 3.5.1. The following values were modified from the default MATLAB *optimoptions*:

- MaxIterations – Maximum number of times the optimizer can iterate to find a minimum;
- OptimalityTolerance – Minimum value for the first-order optimality [441], solver terminates if this threshold is crossed;
- StepTolerance – Minimum value that each index of the parameter array can change from one iteration of the algorithm to the next; if fmincon needs to take a smaller step to find a minimum, the iterations are halted;

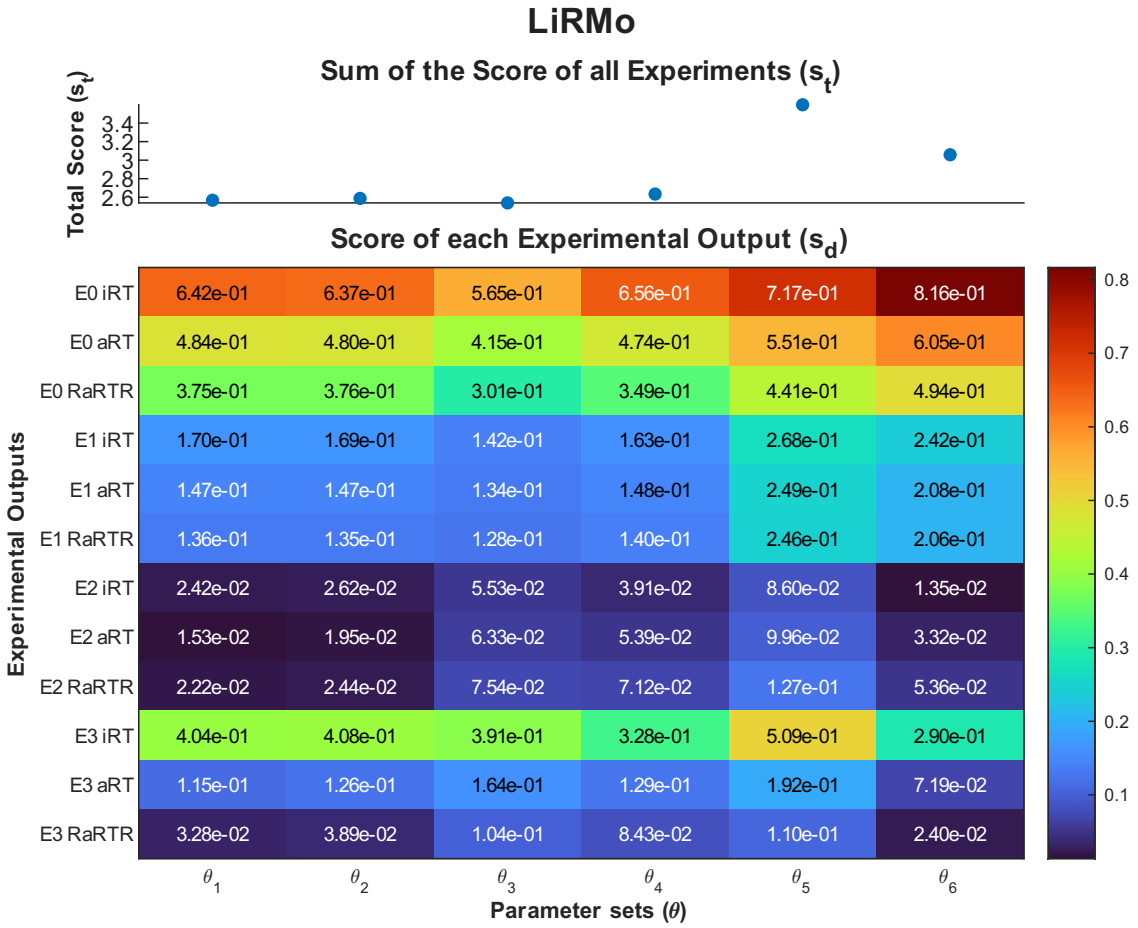


Figure 4.9: LiRMo diagnostic scores plot. Fitting scores of LiRMo given output data from figure 4.7 and inputs shown in figure B.3. θ_{1-6} are the results of optimizing the model to this data as shown in figure 4.10, each corresponding to an optimization algorithm as follows: 1 – Genetic algorithm, 2 – fmincon, 3 – Simulated annealing, 4 – Pattern search, 5 – Particle swarm, and 6 – Surrogate optimization. Lower scores represent a better fit to the model.

- FiniteDifferenceType – Choice between estimating the gradients of the objective function for one direction of the parameter array or for both directions of the parameter array;
- MaxFunctionEvaluations – Maximum number of evaluations of the objective function.

These options were selected for modification after trial and error utilization of the solver. For further reading on how to use this solver, please refer to the online MATLAB documentation [442].

4.3.3.2 Simulated annealing

This solver is based on the simulated annealing algorithm described in section 3.5.2. The following values were modified from the default MATLAB *optimoptions*:

- MaxTime – Maximum run time for the algorithm;
- InitialTemperature – Initial temperature parameter for the algorithm; temperature determines the step in parameter space after each iteration and the probability of accepting a new θ for the next iteration;
- ReannealInterval – Number of θ that must be accepted before re-annealing.

These options were selected for modification after trial and error utilization of the solver. For further reading on how to use this solver, please refer to the online MATLAB documentation [443].

4.3.3.3 Pattern search

This solver is based on the pattern search algorithm described in section 3.5.3. The following values were modified from the default MATLAB *optimoptions*:

- UseCompletePoll – Used to decide if all θs from the pool should be evaluated. If it is set to false, it takes the first θ with a better objective function than the last θ and uses it for the next iteration. If it is true, it evaluates all the θs from the pool and uses the best one as the staring point of the next iteration;
- MaxMeshSize – Maximum size of the mesh;
- MaxFunctionEvaluations – Maximum number of evaluations of the objective function.

These options were selected for modification after trial and error utilization of the solver. For further reading on how to use this solver please, consult the online MATLAB documentation [444].

4.3.3.4 Genetic algorithm

This solver is based on the genetic algorithm described in section 3.5.4. The following values were modified from the default MATLAB *optimoptions*:

- MaxGenerations – Maximum number of the generations the solver is allowed to iterate over;
- PopulationSize – Size of the population used by the solver;
- MutationFcn – Function used create the *mutated child* for the next generation of the solver. *mutationadaptfeasible* was chosen because it is the only built-in function that guarantees that the population does not go outside of bounds.

These options were selected for modification after trial and error utilization of the solver. For further reading on how to use this solver, please consult the online MATLAB documentation [445, 446].

4.3.3.5 Particle swarm

This solver is based on the particle swarm algorithm described in section 3.5.5. The following values were modified from the default MATLAB *optimoptions*:

- MaxIterations – Maximum number of iterations for the solver;
- SwarmSize – Number of θs used in each iteration by the solver.

These options were selected for modification after trial and error utilization of the solver. For further reading on how to use this solver, please consult the online MATLAB documentation [447].

4.3.3.6 Surrogate optimization

This solver is based on the surrogate optimization algorithm described in section 3.5.6. The following values were modified from the default MATLAB *optimoptions*:

- MaxFunctionEvaluations – Maximum number of evaluations of the objective function;
- MinSampleDistance – Minimum distance between successive evaluations of the minimum of the surrogate function.

siisThese options were selected for modification after trial and error utilization of the solver. For further reading on how to use this solver, please consult the online MATLAB documentation [448].

4.3.4 Parameter estimation

In this subsection, the behaviour of parameter estimation software is demonstrated using the example models. The software was designed to match model output to experimental data by tweaking the parameters of the reactions of the model.

In this example, the inputs to the model are not optimized with this software so it is necessary to ensure that they are reasonable in order to achieve the desired output. In a real experiment, these would try to mimic real inputs to a biological system. The hand-crafted inputs to the model can be seen in figures B.3 and B.7 for LiRMo and LiREM respectively. This

tweak is the only manual customization done and, for all others, the algorithms automatically perform all the optimization.

Six optimization algorithms were used, as detailed in section 3.5 and implemented via built-in MATLAB functions described in section 4.3.3.

To ensure fairness in comparison, similar computing time was allocated to each algorithm on an Intel i9-10980XE with 18 physical and 36 virtual cores. The genetic algorithm and particle swarm, capable of utilizing all available cores, were run in multi-core mode. Conversely, fmincon, simulated annealing, pattern search, and surrogate optimization were executed in multi-start mode, with each optimization using a virtual core, due to their inability to efficiently utilize all of the multiple cores.

For the optimization process, an allowable range of log10(-5) to log10(5) was set for all seven parameters. The upper limit ensures reactions are not faster than typically expected in a biological system, while the lower limit prevents them from being too slow to be significant within the time-scales of the model. If data exists that restricts certain parts of the parameter space, these boundaries could be adjusted accordingly.

The parameter estimation code produces the following plots:

- The best-optimized θs for each algorithm. Examples for LiRMo and LiREM are respectively shown in figures 4.10 and B.10;
- The scores of each optimized θ for all algorithms under all starting conditions. Examples with time normalized to ensure equal allocation are shown in figures 4.11 and B.11 for LiRMo and LiREM respectively. These figures provide a comparative perspective on the ability of each algorithm to estimate optimal θs. It is noteworthy that the genetic algorithm and particle swarm were generally found to be the most effective for optimization, although the results are model-specific. In LiRMo all of the algorithms show a similar performance and in LiREM pattern search, simulated annealing, and genetic algorithm are slightly better than the other ones tested;
- All the diagnostic plots are mentioned in section 4.3.2 and were used to illustrate the performance of the different algorithms.

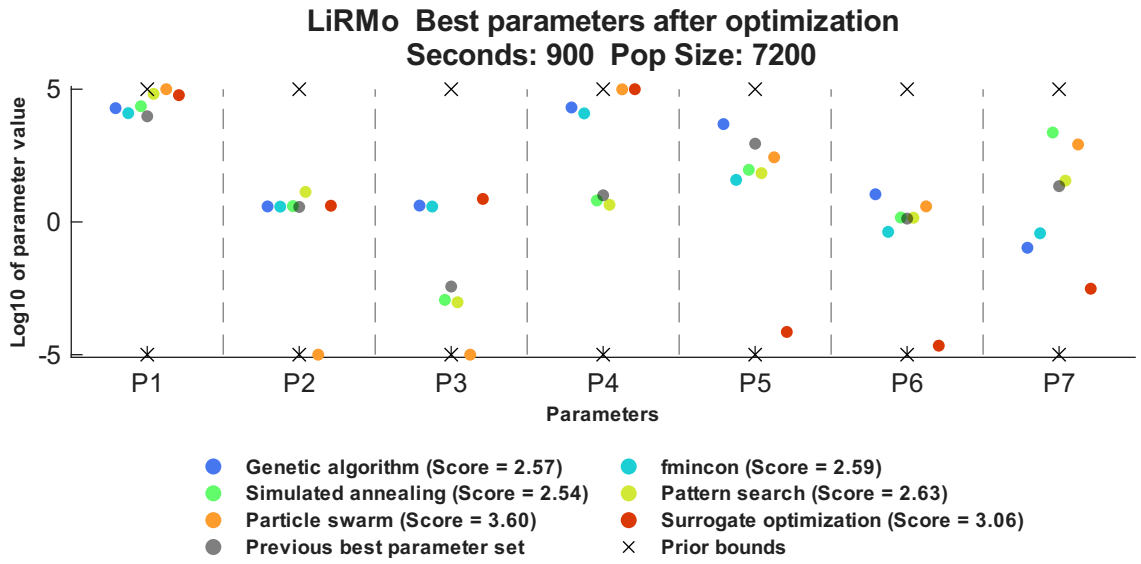


Figure 4.10: LiRMo optimized θ. The figure illustrates the best θ derived for each of the six parameter estimation algorithms used.

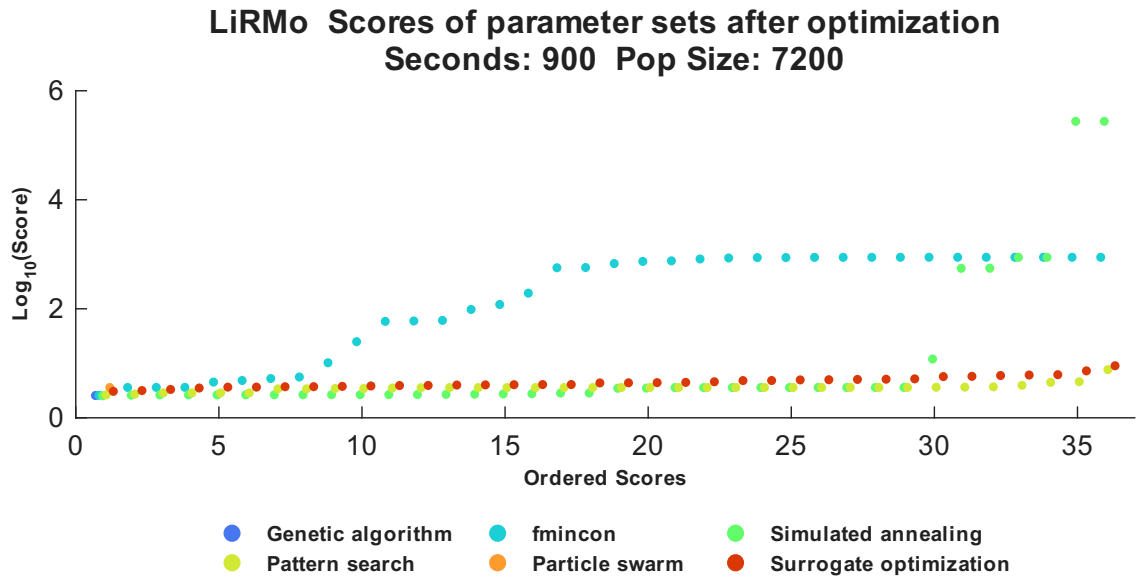


Figure 4.11: LiRMo optimized θ comparative scores. This figure displays the score achieved by each optimized θ, normalized to account for the same computational time allocated to each algorithm. The scores provide an insight into the relative effectiveness of different algorithms in estimating optimal θs.

4.3.5 Sensitivity analysis

Sensitivity analysis software was implemented in order to better understand the uncertainties inherent to matching parameters to data in biological models. In this section, the behaviour of this software will be explained using the two example models described in sections 4.1.1 and 4.1.2.

The uncertainties were calculated taking into account the data generated in section 4.3.1,

shown in tables 4.5 and B.3, and relevant inputs to the model shown in figures B.3 and B.7. Where relevant, the best parameter fit used was the one obtained with the processes explained in section 4.3.4.

4.3.5.1 Local sensitivity analysis

Local sensitivity analysis (LSA) can be used to explore the immediate surroundings of an optimized θ. If the system is simple enough, it can provide relevant information to determine which parameters have a relevant influence in matching the data, and also to help understand if there are correlations between parameters. The code implements the process explained in section 3.6.1. It outputs one plot showing the values of mean (μ) and standard deviation (σ) for each of the analysed model parameters. Examples of this plot can be seen in figures 4.12 and 4.13 for LiRMo and LiREM respectively.

For LiRMo there are 7 parameters. From these, it would be expected that just a subset would have a disproportionate influence in its ability to match the data (since near an optimal value, only one or two parameters in a closed loop will be rate-limiting). Figure 4.12 supports this, as we can see that P_{1-4} are the ones showing a disproportionately amount of μ and σ, and are probably the most influential parameters near this optimized θ. Given the interpretations from section 3.6.1, it can also be concluded that these four parameters have non-linear effects and interactions with each other. Conversely, P_{5-7} have very low values of μ and σ, suggesting that small variations of these parameters near the optimized θ do not significantly influence the model score.

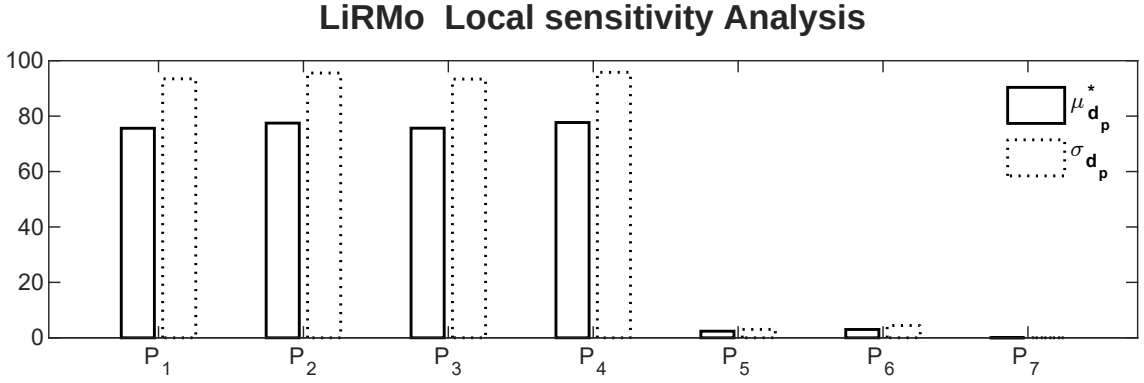


Figure 4.12: LiRMo LSA. σ_{d_p} and mean absolute effect ($\mu^*_{d_p}$) are calculated for P_{1-7} as shown in equations (3.15) and (3.16), respectively, with LiRMo s_t as $f(\theta)$. Interpretations for the values of σ_{d_p} and $\mu^*_{d_p}$ are explained in section 3.6.1.

For LiREM there is a larger number of 12 parameters. As previously mentioned, only a portion of these contribute to the model ability to match the data given this optimal θ. P_{1-4} are important as demonstrated earlier, but now P_{9-10} are also influential. As shown before,

the high values of μ and σ indicate that these six parameters have non-linear effects and interactions with each other. $P_{5–8,11,12}$ are not important near this optimal θ.

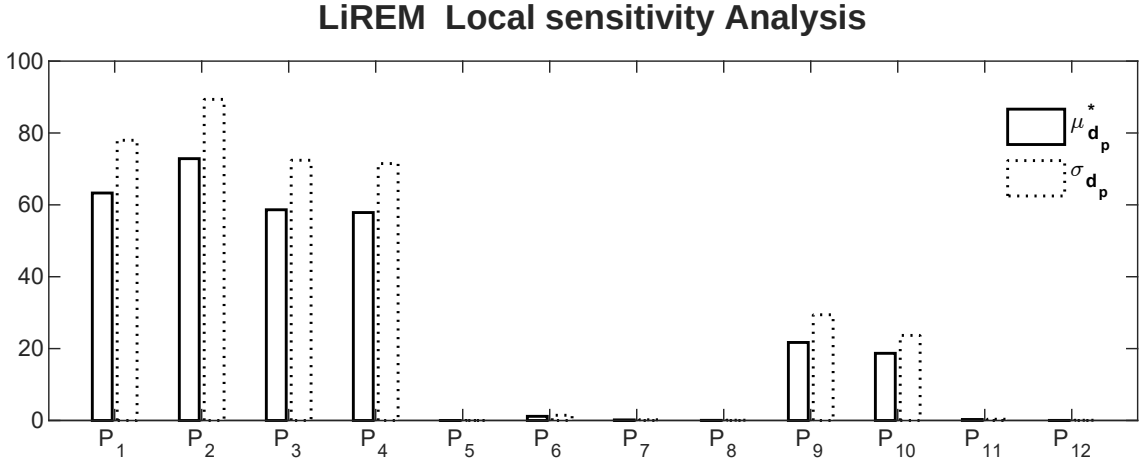


Figure 4.13: LiREM LSA. σ_{d_p} and $\mu^*_{d_p}$ are calculated for $P_{1–12}$ as shown in equations (3.15) and (3.16), respectively, with LiREM s_t as $f(\theta)$. Interpretations for the values of σ_{d_p} and $\mu^*_{d_p}$ are explained in section 3.6.1.

One possible interpretation of which parameters are revealed as important is that some parameters are rate-limiting and so small fluctuations in their values lead to a large influence on the model score.

4.3.5.2 Global sensitivity analysis

Although LSA is useful to explore the immediate vicinity of a certain θ, due to the large dimensional size and complexity inherent to biological models, global sensitivity analysis (GSA) approaches are necessary. This method allows the user to get a broader sense of which parameters influence the match of a certain piece of data, and can be used to explore near certain optimized θs or a large range of θs.

Given the increased complexity of this analysis (compared to LSA), it is important to take into account the amount of samples that will be needed to effectively explore a certain parameter volume and to be mindful of computational feasibility.

The code used implements the process described in section 3.6.2. It outputs the following plots:

- First order sensitivity index (S_{Fi}) for each experimental output (Eo). Examples of this plot can be seen in figure B.12 for LiRMo and LiREM. These plots show that parameters $P_{3,4}$ exert significant influence on the model fit to both iRT and aRT data. In contrast, parameters $P_{1,2,5–7}$ show less influence, and parameters $P_{8–12}$ are largely non-influential. Regarding iE and aE data, the first seven parameters demonstrate reduced importance, while parameters $P_{8,9,12}$ are notably influential;

- Total order sensitivity index (S_{Ti}) for each Eo. Examples of this plot can be seen in figure B.13 for LiRMo and LiREM. These plots demonstrate a similar pattern to the first order sensitivities but with slightly increased influence across all parameters. This enhancement suggests the presence of non-trivial higher order interactions among most parameters, a phenomenon anticipated due to the cyclical nature of the model structure;

- S_{Fi} for each E. Examples of this plot can be seen in figure B.14 for LiRMo and LiREM. The plots indicate that parameters $P_{3,4}$ consistently exhibit significant influence across all experiments, highlighting their direct influence in the models' responses to the various experimental conditions. In contrast, parameters $P_{1,2,5–7}$ in LiRMo and $P_{1,2,5–12}$ in LiREM display substantially lower impact on the experimental outcomes;

- S_{Ti} for each E. Examples of this plot can be seen in figure B.15 for LiRMo and LiREM. As previously mentioned, these plots demonstrate a similar pattern to the first order sensitivities but with slightly increased influence across all parameters;

- S_{Fi} for final values of each Eo. Examples of this plot can be seen in figure B.16 for LiRMo and LiREM. For these plots, a careful interpretation is required because by measuring the final value of the experimental output, any input given to the model will have less influence and, in effect, the measurement will correspond to an approximation to the steady state. Indeed these specific plots are made for models where a steady state analysis is important. Given these considerations, it can nonetheless be observed that $P_{3,4}$, and to a lesser extent $P_{5,6}$ are the most important for iRT and aRTs, and that $P_{8,12}$, with $P_{9–11}$ to a lesser extent, are the important ones for iE and aE;

- S_{Ti} for final values of each Eo. Examples of this plot can be seen in figure B.17 for LiRMo and LiREM. As before, these plots demonstrate a similar pattern to the first order sensitivities but with slightly increased influence across all parameters;

- S_{Fi} and S_{Ti} for all E. Examples of this plot can be seen in figure B.18 for LiRMo. In these plots, for both LiRMo and LiREM, $P_{3,4}$ show larger sensitivity, suggesting that these are the parameters with the greater influence on model score. For both models, Total order sensitivities are larger than first order sensitivities suggesting large interplay between different parameters;

- S_{Fi} and S_{Ti} for all E. Examples of this plot can be seen in figure 4.14 for LiRMo and LiREM. The various parameter sensitivities are very similar to the ones on the previous

plot, and the various quartiles of the box plots are relatively small, suggesting that the sample size chosen was adequate.

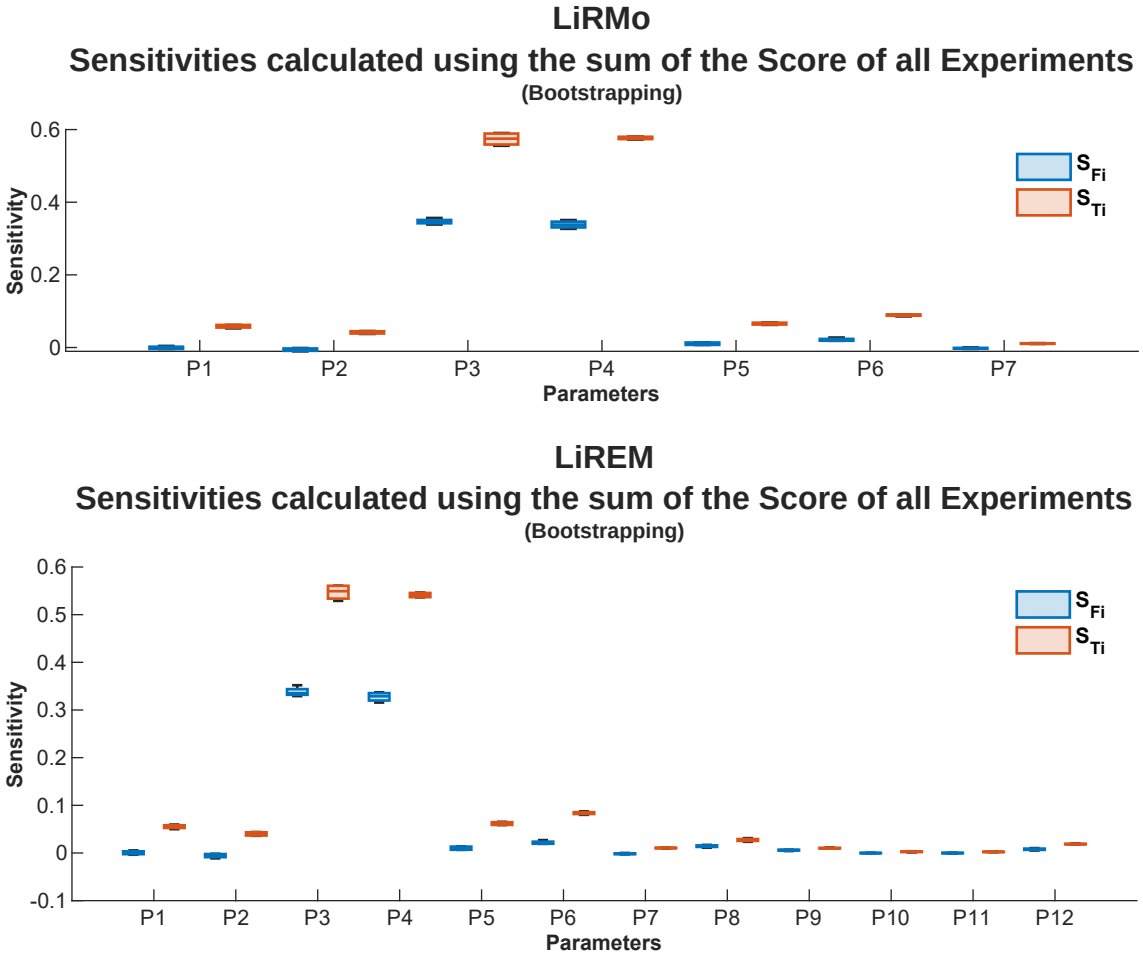


Figure 4.14: LiRMo and LiREM GSA S_{Fi} and S_{Ti} for all Es. S_{Fi} and S_{Ti} are calculated for all model parameters (P_{1-7} for LiRMo and P_{1-12} for LiREM) given each model s_t, as shown in section 3.6.2.1. A sample size of 10000 θ sampled as in section 3.6.2.2 was used. Bootstrapping was performed as shown in section 3.7. The box plots have a darker line for the median of the sampled sensitivities, the top and bottom of the box correspond to the 75% and 25% percentile, respectively, and the top and bottom of the whisker correspond to the non outlier maximum and minimum of the sample, respectively [449]. Higher values indicate greater sensitivity of a specific parameter given a certain experimental output.

4.3.6 Profile likelihood analysis

Profile likelihood analysis (PLA) is used to delineate feasible parameter boundaries by rigorously evaluating the likelihood of different parameter values within models. This analytical approach is important for creating plausible parameter ranges and assessing their identifiability, which is crucial for the reliable application of complex models.

The optimized θs of the example models LiRMo and LiREM, obtained as detailed in

section 4.3.4, were the starting point of the PLA. The PLA was conducted for each parameter selected for analysis (in this case, all model parameters were chosen, but the software allows for the selection of a subset). The methodology described in section 3.7 was followed, resulting in plots that illustrate the feasibility landscape of the parameters. These plots highlight ranges intersecting the χ^2 critical threshold at the 95% confidence level, indicating statistically plausible parameter values consistent with the experimental data. It is important to acknowledge that while individual parameter ranges may be feasible, their combinations might not be, due to complex interactions between parameters. A combined example, including the outputs of two different runs for LiRMo and LiREM is presented in figure 4.15.

In this figure, the PLA reveals distinct patterns for the LiRMo and LiREM models. For LiRMo, $P_{1,2–4,6}$ have one side of their range crossing the inverse cumulative distribution function (icdf)(χ^2, 0.95) threshold, indicating confidence intervals bound on only one side, while P_5 is non-identifiable, encompassing the entire prior parameter bounds. In LiREM, $P_{1,3,4,6,8–12}$ exhibit similar behaviour, with their confidence ranges bound on one side, and $P_{5,7}$ remaining non-identifiable. Notably, P_2 shows a significant improvement in identifiability in LiREM, crossing the icdf(χ^2, 0.95) threshold twice, thereby having a fully defined confidence region.

This analysis, although using artificially generated data, demonstrates the software capability to accurately estimate parameter confidence ranges across models of varying complexity. This lays a solid foundation for further uses of PLA later in this dissertation, where PLA is applied to increasingly complex models.

4.3.7 Discussion

This section discusses the methodologies and results of the model analysis software, emphasizing the strategic generation and employment of dummy data, the critical role of diagnostic plots, the rationale behind utilizing MATLAB Optimization Toolbox, and the insights obtained from sensitivity analyses and PLA. Through a thorough evaluation, these methods are highlighted for their pivotal roles in bolstering model robustness, reliability, and accuracy.

The dummy data generated for this dissertation was crafted to closely emulate real experimental conditions typically encountered in biological research. This was achieved by simulating natural biological response curves to timed stimuli, reflecting diverse experimental scenarios. The inclusion of multiple response amplitudes and varied input durations was instrumental in capturing a wide range of biological responses, akin to Hill equation be-

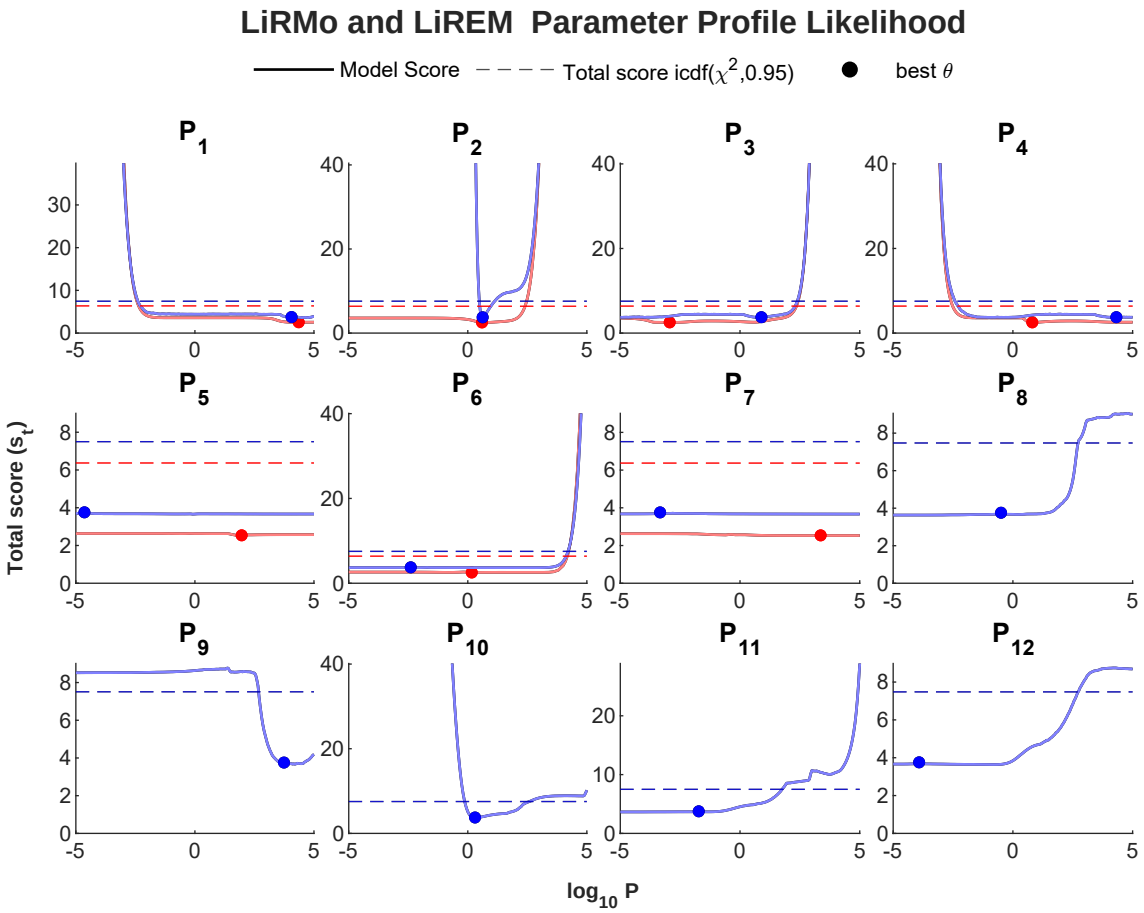


Figure 4.15: LiRMo and LiREM PLA for all Es. Red and blue colours depict the profile likelihoods for LiRMo and LiREM, respectively. Each subplot shows the curves for a single parameter. The x-axis represents the log-transformed parameter values (log10 P) and the y-axis depicts the total score (s_t). The confidence regions are delimited by the intersection of the continuous line and the dashed lines, with the region falling below the horizontal dashed threshold. Note that this region can be discontinuous. The continuous lines represent $-2PL\left(\theta_j\right)$, which correspond to the scores being measured as shown in equation (3.25). The horizontal dashed lines represent the critical threshold, calculated as $\chi^2_{1;0.95} - 2LL(\hat{\theta})$, derived from the maximum log-likelihood value ($\hat{\theta}$) obtained with the model score, adjusted by the 95th percentile of the χ^2 distribution, reflecting the 95% confidence level and 1 degree of freedom as shown in equations (3.26) and (3.27). The points are $-2LL\left(\hat{\theta}\right)$, which represents the best log-likelihood score obtained during parameter estimation, as shown in equation (3.24).

haviour in laboratory settings. Such a comprehensive approach ensures a robust platform for testing the capabilities of the model analysis software, enabling rigorous validation of model predictions against expected outcomes.

Diagnostic plots play a critical role in the model analysis process, offering both visual and quantitative assessments of model performance. The use of score comparison plots for different θ allow the determination of which ones most accurately align with specific experimental results, instead of relying on a global score. Time series plots further aid in dissecting the model performance across different data segments, highlighting any discrep-

ancies that may arise from the model structural or parametric configurations. These plots enable us to identify whether discrepancies, such as poor fits, arise from the model inability to accurately represent rapid increases and decreases in experimental outputs, or from its failure to adequately fit steady states or baseline levels. Visual inspections facilitated by these plots are essential for verifying the correctness and biological relevance of the data inputs and outputs, serving as a fundamental step in the iterative refinement of both model and experimental design.

The choice to employ MATLAB Optimization Toolbox was driven by its robustness, extensive capabilities, and seamless integration with the custom-developed code for this dissertation. The decision to utilize existing, well-validated tools instead of developing custom algorithms allowed for a more focused effort on enhancing other aspects of the model analysis software, ensuring a comprehensive and reliable modelling framework.

LSA was used to identify parameters that have a substantial influence near an optimized θ, highlighting their potential impact on model outputs. Examination of results from these example models reveals a significant correlation between LSA and PLA. Parameters identified as highly sensitive often approach the confidence threshold in PLA, validating that both methods provide meaningful results.

GSA expanded upon LSA by examining the effects of parameters throughout the entire parameter space. This approach is crucial for pinpointing parameters that significantly influence the outputs of models, particularly those involved in essential biological reactions or pathways. When examining the results from sample models, it became evident that parameters orchestrating critical biological processes, such as receptor activation or effector functions, tended to dominate the sensitivity analysis. This dominance is attributed to their direct control over output variables. For instance, parameters $P_{3,4}$, which are involved in the reversible reaction "$iR \rightleftharpoons aR$", produce a measurable output and thus exhibit high sensitivities. In contrast, parameters like $P_{5–7}$, participating in the reactions "$iR + L \rightleftharpoons iRL$" and "$kf_2 \times iRL - kr_2 \times aRL$", which necessitate multiple steps to yield the measured output, demonstrate lower sensitivities due to their indirect roles in activating the system.

PLA is an essential tool for refining parameter estimates, enhancing the precision and reliability of model predictions. By establishing statistically robust parameter bounds, PLA can enable more targeted and effective subsequent optimizations. For instance, as illustrated in figure 4.15, the confidence regions of LiREM fell within those of LiRMo for the parameters they share. This overlap implies that the initial parameter estimates for LiREM could be effectively derived from the LiRMo confidence regions, thereby simplifying the parameter

estimation while maintaining its accuracy. Such a strategy can prove particularly valuable in more complex models where interdependencies among parameters can complicate the optimization process.

The methodologies discussed here underscore the importance of a rigorous analytical framework in developing robust and accurate computational models. They are adequately suited to produce models that are finely tuned to align with empirical data, ensuring their applicability in real-world biological research and their potential to provide meaningful insights into complex biological systems.

4.4 Timing model

Previous sections of this thesis explored example models; building upon them, this section focuses on a previously established and validated model [2], designated in this thesis as Timing Model-Original (TM-O). It incorporates dual inputs (striatal calcium and dopamine), and a comprehensive intracellular signalling cascade, detailed in section 4.4.1, addressing the timing order constraints of input integration in relation to D32 and A21 activity.

TM-O includes the autophosphorylation of CaMKII via manually-crafted functions that limited software compatibility. These functions have been updated with ordinary differential equations (ODEs) to enhance model versatility, resulting in the modified version, Timing Model-Modified (TM-M). This section will specifically focus on the enhancements in TM-M, demonstrating the software's robustness and efficacy in handling complex model modifications.

Initial results from TM-M, which are now published [1], lay the groundwork for this section. This section will focus on the methodologies employed and present additional analyses that build upon the published work.

4.4.1 Model overview

This section provides a comprehensive overview of TM-O and the modifications employed in TM-M for its analysis in this thesis. The modifications present in TM-M are visualized in figure 4.16, which illustrates the intricate pathways and interactions within the model.

4.4.1.1 Calcium cascade

The Ca^{2+} signalling cascade starts with an influx of Ca^{2+} and its binding to CaM, activating it [238]. This conversion is modelled as a two-step process, with each step simulating

calcium binding to CaM domains based on the Adair-Klotz [450] double binding-site model. CaM can bind its target proteins in both the partially and the fully Ca^{2+}-bound state. This is represented by a set of cyclical reactions similar to those shown in figure 4.1. Such cycles require that certain parameters be expressed as a function of other, independent, parameters in order to satisfy thermodynamic constraints, as previously described by Nair *et al*. [179].

CaMKII is an important target of CaM in this model. When activated, it undergoes a translocation between the cytoplasmic compartment and the post-synaptic density (PSD). In TM-O, this is represented by the presence of two different pools of CaMKII with corresponding translocation reactions tuned to guarantee that more of the active CaMKII resides in the post-synaptic density (PSD) and more of the inactive CaMKII resides in the cytoplasm.

4.4.1.2 Autophosphorylation of CaMKII

The autophosphorylation of CaMKII represents a pivotal process in the model, enhancing its affinity to CaM and promoting its translocation to the PSD compartment, where it is subject to dephosphorylation by PP1. In TM-O this autophosphorylation was modelled according to a previous work [451] and included some manually-tuned functions, which made the model incompatible with software systems that are only capable of processing ODE.

In TM-M the process of CaMKII autophosphorylation was modified to improve software compatibility and replaced by eight reactions consisting of four repeated reactions for CaMKII in the two relevant compartments. These reactions are the reversible binding of two CaMKII species followed by the irreversible phosphorylation of one of them, and the reversible binding of one phosphorylated and one unphosphorylated CaMKII followed by the irreversible phosphorylation of the second CaMKII, as can be seen in the schematic in figure 4.16B. The parameters used in the model file can be linked to the parameters used in figure 4.16 and the analysis through table 4.6.

The active CaMKII phosphorylates its substrates, which are subsequently dephosphorylated by PP1. To ensure that these effects are represented, TM-O employs a generic CaMKII/PP1 substrate species localized within the PSD compartment.

4.4.1.3 Dopamine cascade

The dopamine signalling cascade starts with the activation of the D1R, followed by a series of events that engage AC5, elevate cAMP levels, and activate PKA [175, 182–184]. Activated PKA phosphorylates D32 at Thr-34 [244], which in turn is essential for modulating

the activity of PP1 [244].

TM-O demonstrated that the impact of dopamine signalling is critically dependent on the timing of calcium and dopamine inputs. TM-O suggests that an effective downstream response within the signalling pathway is contingent upon the dopamine signal promptly following the calcium input, within a narrowly defined temporal window, thus enforcing strict temporal regulation [2].

This timing constraint is enforced in TM-O by the influence of A21 and its input-order constraint, markedly attenuating the downstream response when dopamine precedes calcium. This incorporation of A21 emphasizes the cascade's sensitivity to the timing and sequence of neurotransmitter signals.

4.4.1.4 Hooks to original model

In TM-M, certain species were identified as key elements for constraining the new reactions, ensuring its response aligns with the response of TM-O. These species are CaM, phosphorylated Substrate (pSubstrate), PP1, and D32, and they are clearly indicated in figure 4.16A, where they are marked in orange for easy reference.

To validate TM-M, especially in terms of maintaining the input order constraint, a set of ten experimental scenarios were chosen. These scenarios, detailed in figure 4.16C, were instrumental in generating data from TM-O reflecting various temporal arrangements of dopamine and calcium inputs. This included configurations where dopamine input precedes, is simultaneous with, or follows the calcium input, as well a scenario featuring exclusively calcium input. The selection of these specific scenarios was guided by the roles and dynamics of the aforementioned species, ensuring a thorough and relevant evaluation of TM-M performance under varied biochemical conditions.

4.4.2 Model analysis

The analysis involved a series of 10 custom-designed experiments, as schematized in figure 4.16C, each employing data from TM-O to inform and constrain TM-M. These experiments introduced Ca^{2+} into the model at a fixed time of 4 seconds, and varied the timing of DA input with a Δt relative to the Ca^{2+} input, ranging from 0 to 8 seconds (E_{0-8}). An additional experiment was conducted without any DA input (E_9). The inputs administrated in the various experiments are depicted in figure 4.17.

The analyses applied to the two example models were also applied to TM-M. However, in contrast to previous efforts, the focus here was exclusively on the new reactions introduced.

Analysis/Figure	Model	Description
F-actin rich cytoplasmic compartment		
P_1	kf_R130	Two unphosphorilaterd CAMKII come together
P_2	kr_R130	Two unphosphorilaterd CAMKII separate
P_3	kf_R131	Two CAMKII separate with one of them phosphorilated
P_4	kf_R132	pCAMKII binds to a unphosphorilated CAMKII
P_5	kr_R132	Two CAMKII separate with one of them phosphorilated
P_6	kf_R133	Two CAMKII separate with both of them phosphorilated
Near-synaptic/PSD compartment		
P_1	kf_R134	Two unphosphorilaterd CAMKII come together
P_2	kr_R134	Two unphosphorilaterd CAMKII separate
P_3	kf_R135	Two CAMKII separate with one of them phosphorilated
P_4	kf_R136	pCAMKII binds to a unphosphorilated CAMKII
P_5	kr_R136	Two CAMKII separate with one of them phosphorilated
P_6	kf_R137	Two CAMKII separate with both of them phosphorilated

Table 4.6: TM-M parameters. The table shows the mapping between different parameter notations: Analysis/Figure: P_{1-6} represent the 6 parameters used in model analysis and in figure 4.16 Model: kf_X and kr_X correspond to the rate constants in the SBtab model file, where X is the reaction number (1–137 in the SBtab 130–137 shown in the table). Description: Briefly explains the biological meaning of each parameter. f indicates forward reactions, while r denotes reverse reactions.

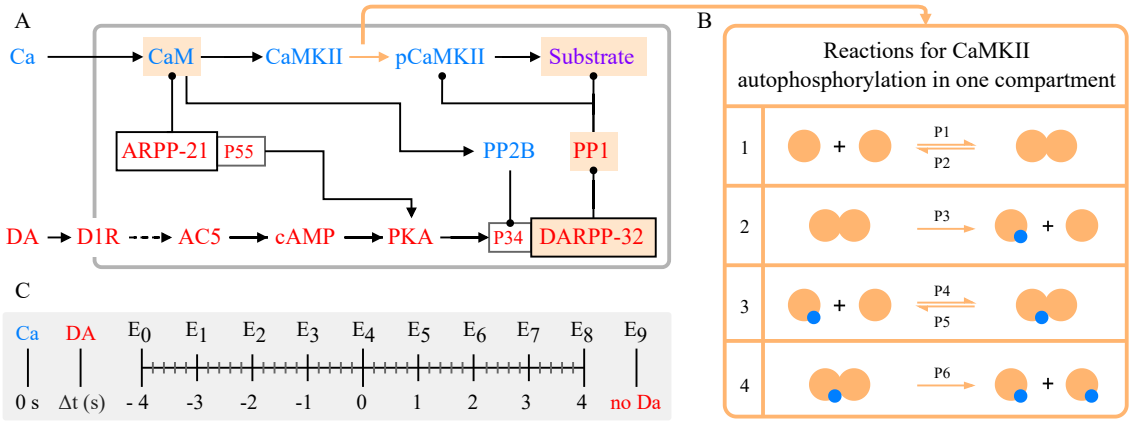


Figure 4.16: TM-O and TM-M schematic overview and experimental timing configurations. A – Signalling pathways and molecular interactions of the TM-O. B – CaMKII autophosphorylation reactions added to TM-M, with CaMKII and phosphate groups denoted by yellow and blue circles, respectively. C – Schematic depicting the relative timing of Calcium and Dopamine inputs across various scenarios used for model validation. Abbreviations: calcium ion (Ca^{2+}), calcium-calmodulin-modulated protein (CaM), calcium-calmodulin-dependent protein kinase II (CaMKII), protein-phosphatase 2B, also know as calcineurin or PPP3 (PP2B), dopamine (DA), D1 dopamine receptor (D1R), adenylyl cyclase type 5 (AC5), cyclic adenosine monophosphate (cAMP), protein kinase A (PKA), protein-phosphatase 1 (PP1), cAMP-regulated phosphoprotein 21 kDa (usually ARPP-21) (A21), dopamine and cAMP regulated phosphoprotein 32 kDa (usually DARPP-32) (D32), experiment (E), parameter (P). In A 'P/p' denotes phosphorylation while in B 'P' denotes parameter. Figure reproduced with permission from Santos *et al*. 2022 [1].

The intent was to keep the core structure of the model intact while refining the CaMKII component. These analyses are explored in sections sections 4.4.3, 4.4.4 and 4.4.5

4.4.3 Parameter estimation

This subsection demonstrates the software's capability to fit TM-M to the data generated by TM-O across experimental scenarios defined in (section 4.4.2). As discussed in section 4.3.4, various optimization algorithms implemented using built-in MATLAB functions from the MATLAB Optimization Toolbox were employed. A series of plots were also similarly generated to validate the effectiveness of the parameter estimation process:

- Figures C.4 and C.5 illustrate the performance of different algorithms;
- Figure 4.18 displays scores for all experimental inputs under the optimal θ obtained by each algorithm;
- Figures 4.19, C.1, C.2 and C.3 compare model outputs to TM-O data, highlighting how closely TM-M aligns with the observed measurements.

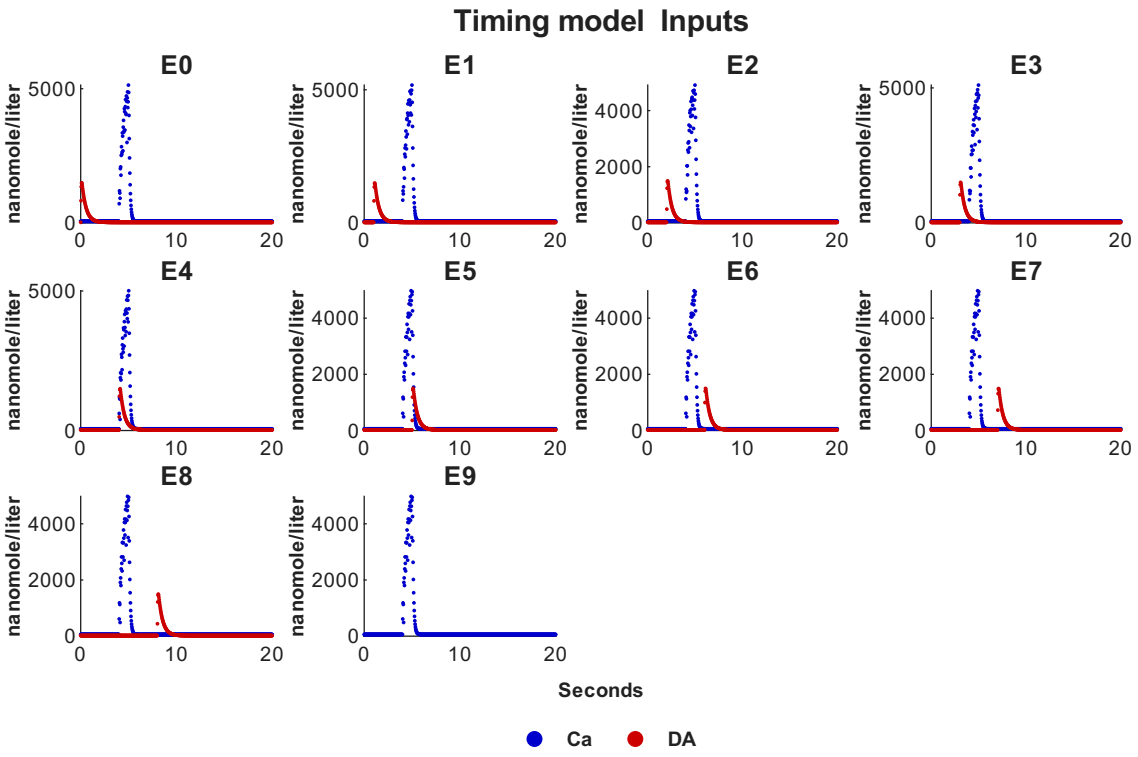


Figure 4.17: TM-M inputs $E_{0–9}$. Experimental input profiles for TM-M from E_0 to E_9, displaying the administration of Ca^{2+} at 4 seconds and DA at 1-second intervals from 0 to 8 seconds. Each graph represents the concentration of inputs over time, with blue lines indicating Ca^{2+} and orange lines indicating DA. $E_{0–8}$ demonstrate the sequential introduction of DA, while E_9 shows a solitary Ca^{2+} input with no accompanying DA. This arrangement allows for the examination of the temporal interaction effects on intracellular signalling pathways within the model.

Figure C.4 shows that despite increased model complexity, all tested optimization algorithms successfully identified appropriate θ values. However, the required time for good optimizations increased in tandem with model complexity, from 900 seconds per algorithm in section 4.3.4 to 7200 here (using an Intel i9-10980XE). This indicates that although the software can manage complex models, computational limitations may arise as optimization durations extend beyond practical limits. Figure C.5 suggests a convergence towards an optimal θ, indicating the potential presence and location of a global minimum within the parameter space.

Figure 4.18 reveals that the best θ values achieved nearly perfect fits for PP1, CaM, and D32 in all but E_3. The primary challenge was fitting the pSubstrate data, especially in experiments $E_{4\text{-}9}$ where DA was either introduced after Ca^{2+} or not present.

Further analysis using figures 4.19, C.1, C.2 and C.3 confirm the observations from figure 4.18. These plots indicate that the model matches the data well for most outputs except for pSubstrate, where discrepancies related to Ca^{2+} introduction were observed. Specifically, the model overestimated pSubstrate production during Ca^{2+} input in experiments $E_{0–3}$

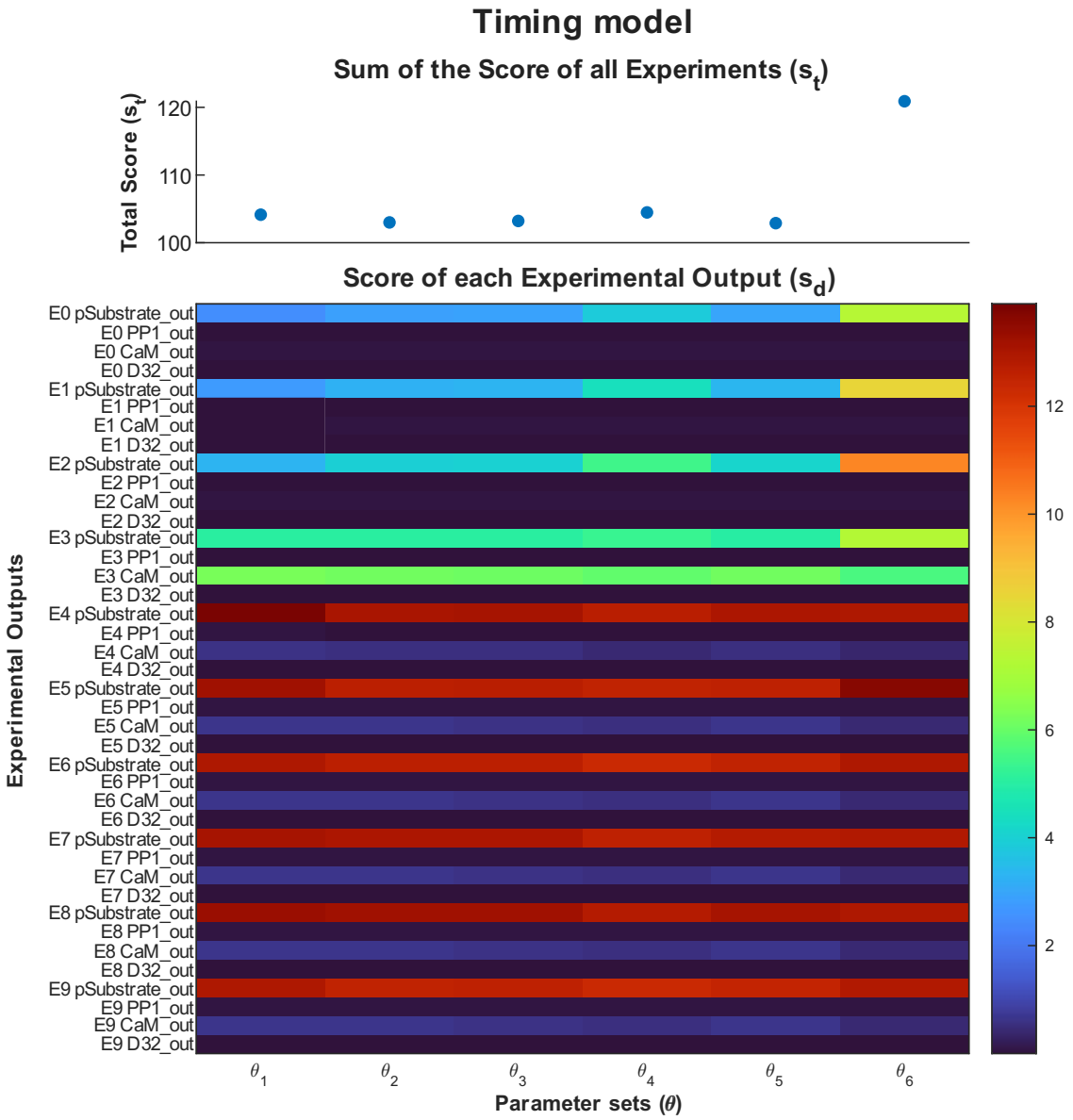


Figure 4.18: TM-M diagnostic scores plot. This heatmap synthesizes the performance of six parameter sets θ_1 to θ_6, derived from different optimization algorithms, across ten experimental conditions ($E_{0–9}$). Each row represents the output variables (pSubstrate, PP1, CaM, and D32) for a given experiment, while the color gradient reflects the scoring metric, with cooler colors indicating lower scores (better fits) and warmer colors denoting higher scores. The chart above the heatmap aggregates the total score for each parameter set across all experiments, providing a quick visual comparison of the overall algorithm effectiveness.

and underestimated it in experiments $E_{5–9}$. This dual behaviour could explain the challenges in achieving better overall fits. The model appears to be in a sort of equilibrium where improvements in fitting the data for $E_{0–3}$ inadvertently worsen the fits for $E_{5–9}$, and vice versa, indicating a fundamental limitation in the current model structure or parametrization that could be addressed in future work.

In addition to the global optimization approach, parameter estimation was also performed

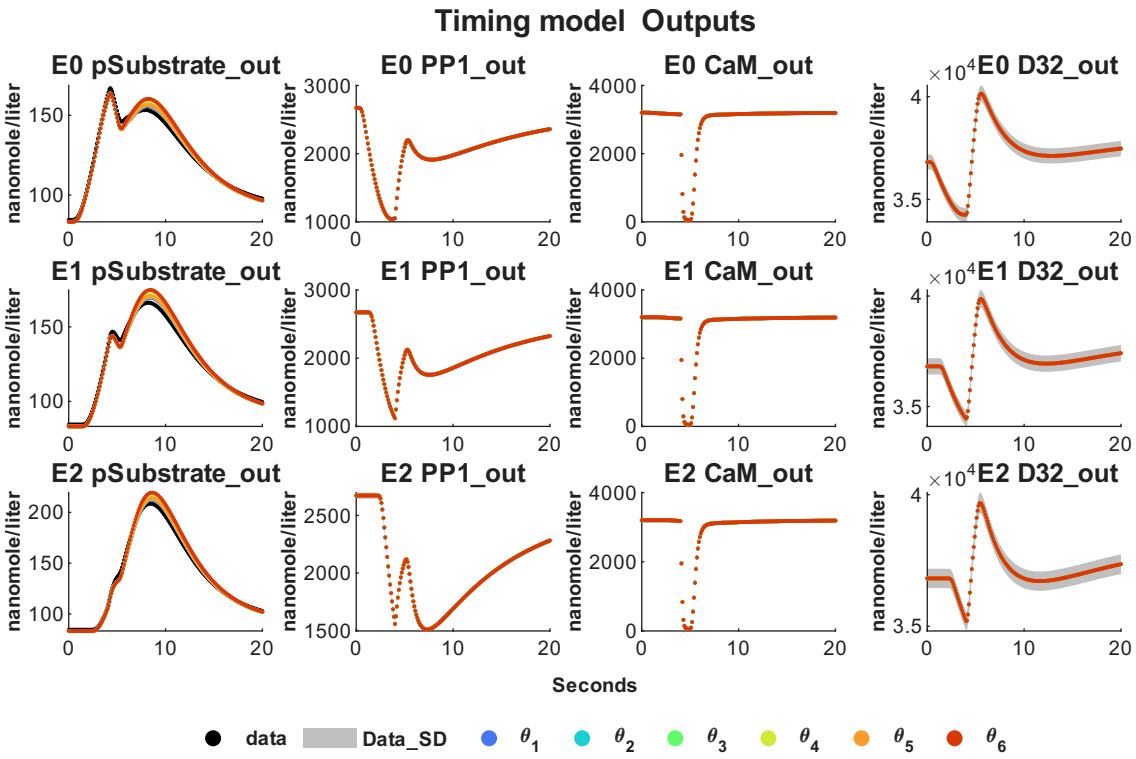


Figure 4.19: TM-M outputs E_{0-2}. The figures display the temporal dynamics of key molecular components: pSubstrate, PP1, CaM, and D32 (respectively pSubstrate_out, PP1_out, CaM_out, D32_out) for TM-M. Each plot contrasts multiple parameter sets (θ_{1-6}) against the observed data and its standard deviation (Data_SD), reflecting the ability of the model to capture the biochemical behaviour in E_{0-2}.

individually for each experiment, resulting in ten distinct parameter sets corresponding to experiments E_{0-9}. These individual estimations served as starting points for the profile likelihood analysis detailed in section 4.4.5.

This approach of global and per-experiment parameter estimation provides a comprehensive understanding of the model dynamics under various conditions, offering a robust framework for future model evaluations and enhancements.

4.4.4 Sensitivity analysis

Sensitivity analysis was applied to TM-M as previously described for the example models. The uncertainties were calculated based on data and inputs presented in section 4.4.2. Local sensitivity analysis used the best parameter fits obtained in section 4.4.3. Global sensitivity analysis used the entire prior parameter range (10^{-8} to 10^{2}).

4.4.4.1 Local sensitivity analysis

In order to provide a deeper insight into the workings of TM-M, the experiments were divided into two groups for local sensitivity analysis. The first set contains the experiments

where dopamine is introduced before calcium ($E_{0–3}$) and the second set includes the experiments where dopamine is introduced after calcium ($E_{5–8}$).

In the first group, as seen in figure 4.20, both parameters P_1 and P_4 exhibit significant μ and σ suggesting that, when dopamine follows calcium, both these parameters are important for the score of the model near the optimal parameter set. Meanwhile, Parameters $P_{2,3,5,6}$ showed negligible mean and standard deviation, reflecting a consistent and minimal impact on the model dynamics throughout the dopamine first scenario.

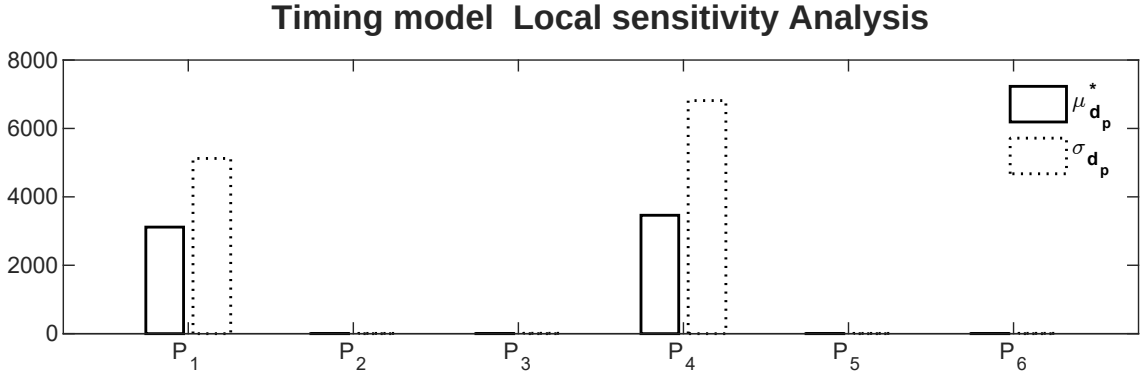


Figure 4.20: TM-M LSA $E_{0–3}$ (DA then Ca^{2+}). The graph displays the μ and σ of Parameters $P_{1–6}$. $P_{1,4}$ show a notable mean and standard deviation. $P_{2,3,5,6}$ exhibit negligible mean and standard deviation, suggesting minimal impact on model outcomes near the best θ.

In the second group, as seen in figure 4.21, only parameter P_1 exhibits significant μ and σ, suggesting a strong and variable impact on the timing model's output. As previously shown, the rest of the parameters do not influence the score of the model near the optimal parameter set.

The behaviour of $P_{1,4}$ suggests that they are well constrained by the data, as minor changes near their optimum value affect the score of that value in a significant manner, while $P_{2,3,5,6}$ are not so well constrained near the optimum score, suggesting that their values could be altered slightly without perturbing the results. This can either suggest that these parameters and associated reactions are not as important for the system in this study or that there is not enough data to properly constrain them.

4.4.4.2 Global sensitivity analysis

As previously described, global sensitivity analysis was performed to get a better understanding of model parameter influences on the measured scores. Various different plots were produced as in the previous GSA in 4.3.5.2 but in this section only S_{Fi} and S_{Ti} for each Eo are going to be discussed as they are the most granular of the results relating variabilities of the parameters with the scores.

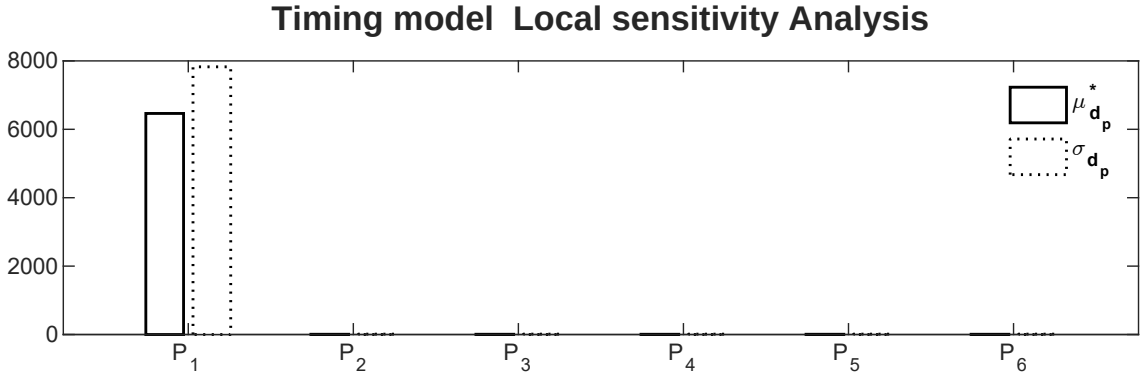


Figure 4.21: TM-M LSA $E_{5–8}$ (Ca^{2+} then DA). The graph displays the μ and σ of Parameters $P_{1–6}$. P_1 shows significant μ and σ values indicating its substantial influence on the model fits near the best θ from parameter estimation. Conversely, $P_{2–6}$ exhibit negligible mean and standard deviation, suggesting minimal impact on model outcomes near the best θ.

Ten experimental conditions were used to calculate the GSA, as detailed in section 4.4.2.

Key observations of the results for S_{Fi} for each Eo, as shown in figure 4.22, are as follows:

- P_1: Showed a dynamic sensitivity pattern in E_0, starting very low for pSubstrate, increasing to low for PP1, and again for medium-low for CaM, and finally decreasing to very low for D32. This pattern, with a general trend of decreasing sensitivity from $E_{0–8}$, highlights the diminishing influence of P_1 in later experiments. E_9, without DA, follows a similar but overall lower pattern, emphasizing the dependency of P_1 on DA;

- P_2: Exhibited negligible sensitivity for pSubstrate and PP1, low for CaM, and very low for D32 in E_0. This pattern, consistently repeated across $E_{0–9}$, suggests P_2 has a minimal impact on the model, especially for pSubstrate and PP1;

- P_3: Demonstrated low sensitivity across all outputs, with a hierarchy of pSubstrate > PP1 > CaM, and D32 approximately equal to PP1. This pattern remained consistent through $E_{1–8}$, slightly altering in E_9 where PP1 and D32 exceed pSubstrate. P_3 contributes modestly and consistently across experiments;

- P_4: Revealed a sensitivity order in E_0 of pSubstrate > PP1 > CaM > D32, with pSubstrate and PP1 classified as low, CaM very low, and D32 negligible. This pattern persisted throughout all experiments, indicating limited but discernible influence of P_4 in the phosphorylation process;

- P_5: In E_0, CaM exhibited very low sensitivity, with pSubstrate, PP1, and D32 being negligible. This pattern, which remained stable in subsequent experiments, suggests

P_5 plays a minor role in the model, with its most significant impact being on CaM interactions;

- P_6: Displayed a pattern in E_0 of pSubstrate > PP1 > CaM > D32, with medium sensitivity for pSubstrate and PP1, CaM being very low, and D32 negligible. This pattern, largely consistent across all experiments, indicates that P_6 is a more influential parameter, especially in the context of pSubstrate availability and PP1 interactions.

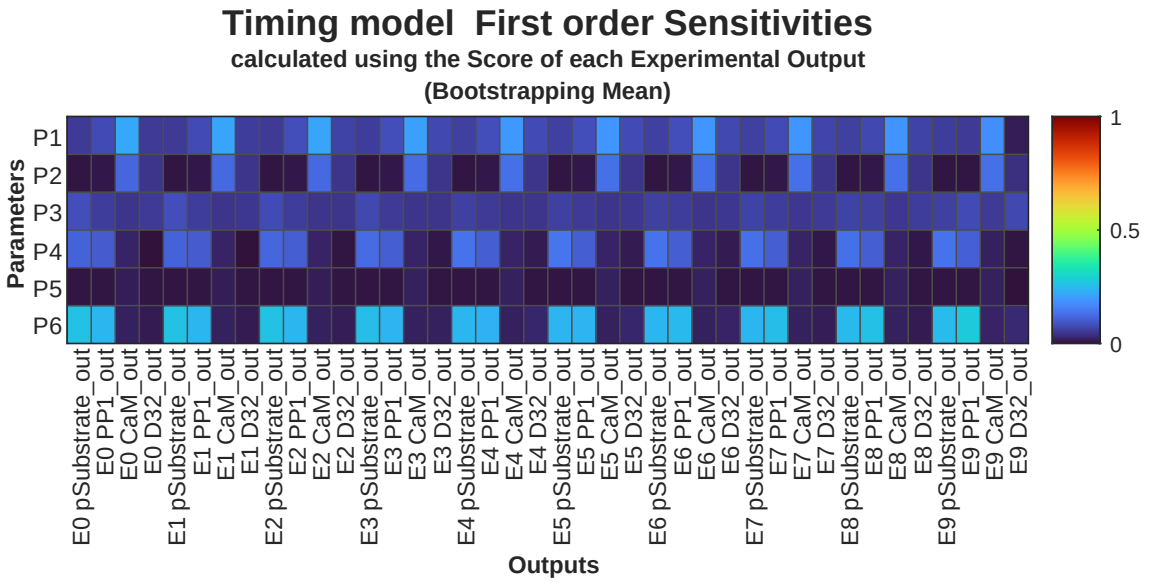


Figure 4.22: TM-M GSA S_{Fi} for each Eo. Each row represents a parameter (P_{1-6}), while each column corresponds to a specific output under different input sequences of DA and Ca^{2+}. The color gradient reflects the bootstrapped mean sensitivity value, ranging from 0 (no influence) to 1 (maximum influence), elucidating the varying impacts of the parameters on model predictions across experimental conditions.

A similar analysis was performed for S_{Fi} for each Eo as shown in figure 4.23, and the most important observations are as follows:

- P_1: Exhibited a diverse range of sensitivities in E_0, with pSubstrate being low, PP1 low-medium, CaM medium-high, and D32 very high. This pattern, with a decreasing trend in sensitivity for PP1, CaM, and D32 from E_{0-9} while maintaining similar levels for pSubstrate, suggests a nuanced role of P_1 across different experimental conditions. In E9, both pSubstrate and PP1 are low, CaM medium, and D32 remains high;

- P_2: Demonstrated a sensitivity gradient in E_0 from very low for pSubstrate to high for D32, implying a broader impact than indicated by first-order sensitivities. This pattern of pSubstrate < PP1 < CaM < D32 persists with minor variations in all experiments up to E_8. In E_9, PP1's sensitivity aligns closer to the very low level of pSubstrate;

- P_3: In E_0, both pSubstrate and PP1 showed medium-low sensitivity, with CaM being medium and D32 high. This pattern, indicating a generally more substantial influence than first-order sensitivities, remains consistent with minor variations through E_{1-8}, affirming the significant role of P_3 in the model;

- P_4: Presented a consistent sensitivity pattern across all experiments, with medium levels for pSubstrate and PP1 and low for CaM and D32. This uniform pattern across different conditions reveals that P_4 has a stable yet moderate influence within the model dynamics;

- P_5: Revealed a low impact on pSubstrate and PP1, while CaM and D32 were medium-low in E_0. This pattern, which remains largely unchanged across all experiments, suggests that P_5 role is minor, with a slightly more pronounced impact on CaM and D32;

- P_6: Showed a pattern in E_0 where pSubstrate had medium-high sensitivity, PP1 medium, CaM medium-low, and D32 low. Consistent across all experiments, this pattern indicates significant cumulative effect of P_6 on the model, particularly regarding the influence of pSubstrate availability and PP1 interactions.

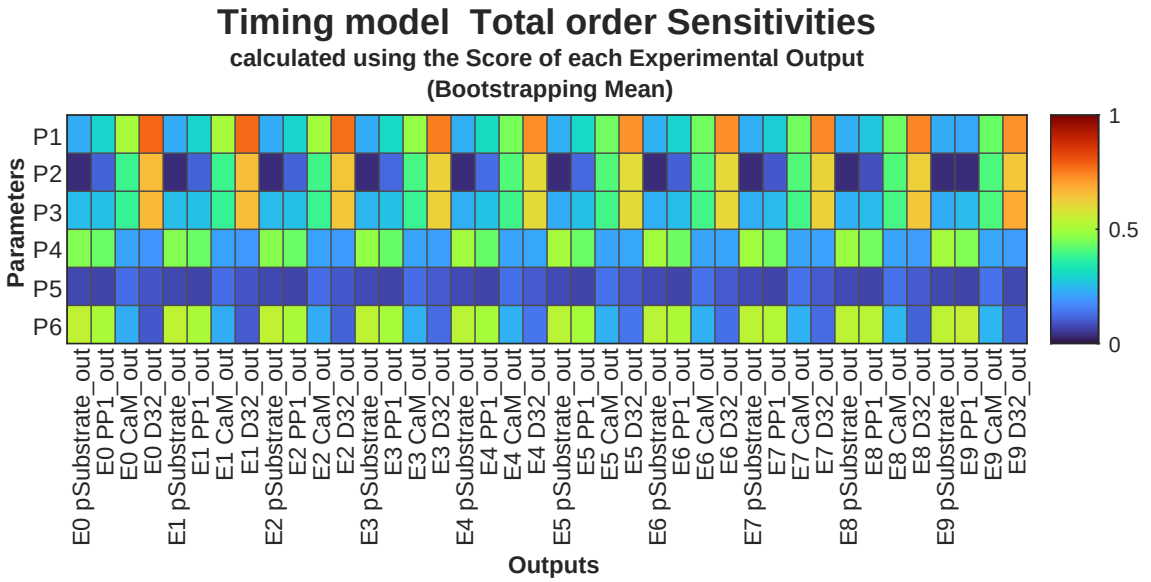


Figure 4.23: TM-M GSA S_{Ti} for each Eo. Each row represents a parameter (P_{1-6}), while each column corresponds to a specific output under different input sequences of DA and Ca^{2+}. The color gradient reflects the bootstrapped mean sensitivity value, ranging from 0 (no influence) to 1 (maximum influence), elucidating the varying impacts of the parameters on model predictions across experimental conditions.

Contrary to the effects seen in the local sensitivity analysis, here all parameters are seen as contributing to the overall sensitivity scores. This suggests that the optimization is taking

the model into a parameter set region that is robust to parameter changes for all parameters except $P_{1,4}$.

These insights contribute to a more detailed understanding of the model dynamics. They help clarify how individual parameters and their interactions influence the model behaviour, enhancing our interpretation of its predictive capabilities. While these findings offer valuable perspectives for this specific model, they also provide a useful reference point for similar studies in biochemical modelling. This analysis sets the stage for upcoming sections of the thesis, where the insights gained will be instrumental in developing and refining a more complex model. By building on the understanding of parameter dynamics and interactions in the timing model, it paves the way for a deeper exploration of biochemical modelling within the broader scope of this work.

4.4.5 Profile likelihood analysis

In this section, contrasting to the composite PLA used in the example models, the PLA is applied using the score of each experiment individually. This nuanced approach dissects the impact of each individual experiment on the parameter space, unveiling how specific experimental conditions, ranging from experiment $E_{0–9}$, influence the likelihood of different parameter values. By isolating the experiments, we can pinpoint the contributions and constraints each one imposes on the model parameters.

As show in figure 4.24, this analysis demonstrated that there are three identifiable ($P_{1,3,4}$) and three non-identifiable parameters ($P_{2,5,6}$) in all experiments. For the identifiable parameters, two distinct feasibility regions are seen for two groups of experiments $E_{0–3}$ (experiments where dopamine is applied before calcium) and $E_{4–9}$ (experiments where dopamine is at the same time, after calcium, or absent). Furthermore, the optimal value in $E_{0–4}$ for P_1 and in $E_{0–1}$ for P4 are in a region where $E_{4–9}$ start showing worse scores and, therefore, show why fitting the model perfectly to all these experiments is fundamentally impossible given this model configuration and data.

The results from this PLA provide a detailed view of the parameter dynamics in the Timing Model, highlighting the regions of the parameter space that are most compatible or incompatible with the experimental data. These findings contribute to a better understanding of the model behaviour under various experimental scenarios and lay the groundwork for further analysis in the Discussion section.

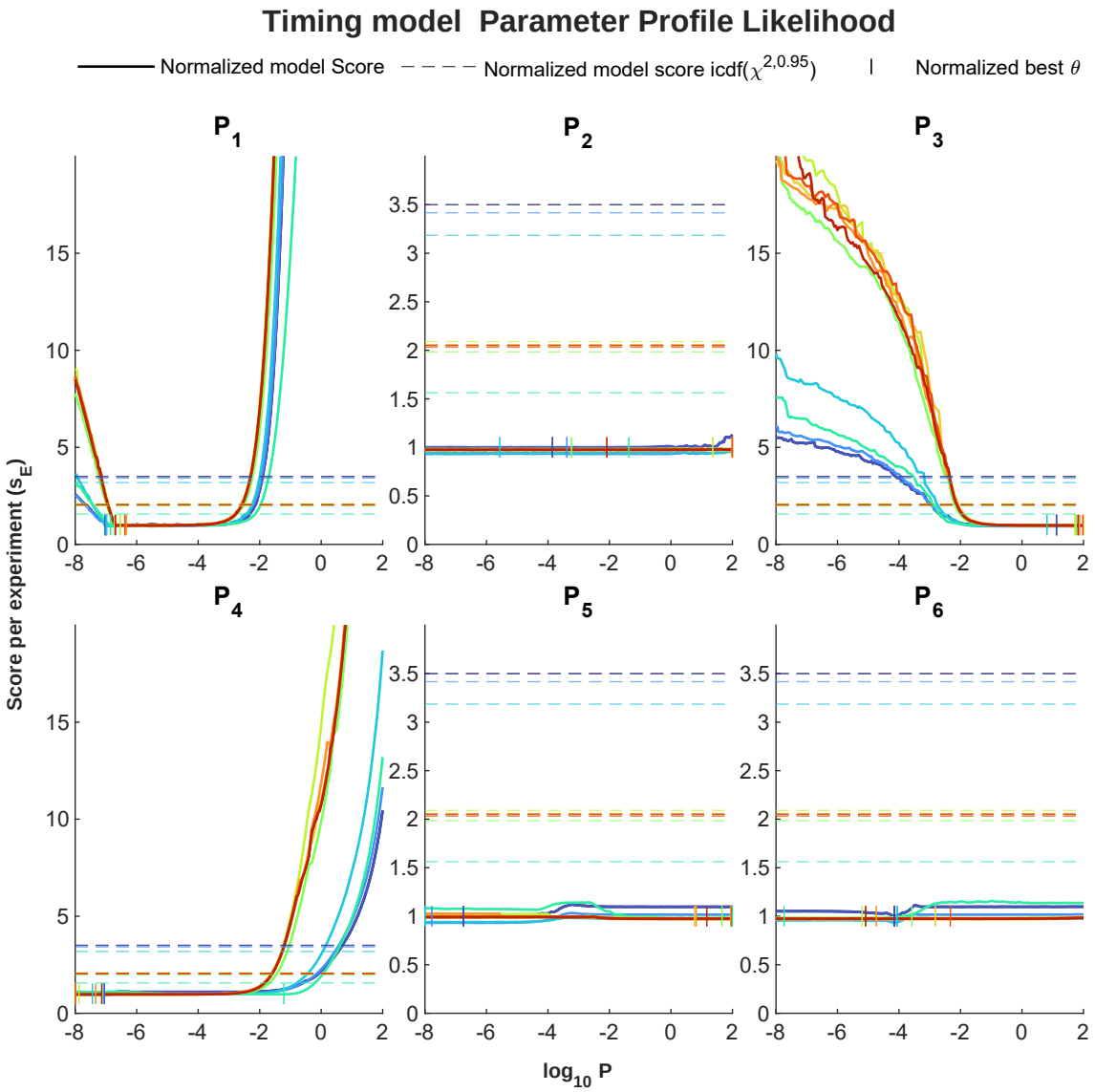


Figure 4.24: TM-M PLA for each E. Each of the ten experiments, E_{0-9}, is depicted by different colors: Dark Indigo for E_0, Blue for E_1, Light Blue for E_2, Cyan for E_3, Green for E_4, Light Green for E_5, Yellow for E_6, Orange for E_7, Red for E_8, and Dark Red for E_9. Each subplot shows the curves for a single parameter across all experiments. The x-axis represents the log-transformed parameter values (log10 P) and the y-axis depicts the score per experiment (s_E). The confidence regions are delimited by the intersection of the continuous line the dashed lines, with the region falling below the horizontal dashed threshold. Note that this region can be discontinuous. The continuous lines represent $-2PL\left(\theta_j\right)$, which correspond to the scores being measured as shown in equation (3.25). The horizontal dashed lines represent the critical threshold, calculated as $\chi^2_{1;0.95} - 2LL(\hat{\theta})$, derived from the maximum log-likelihood value ($\hat{\theta}$) obtained with the model score, adjusted by the 95th percentile of the χ^2 distribution, reflecting the 95% confidence level and 1 degree of freedom as shown in equations (3.26) and (3.27). Vertical dashes are $-2LL\left(\hat{\theta}\right)$, which represents the best log-likelihood score obtained during parameter estimation, as shown in equation (3.24). P_1, P_3, and P_4 show intersecting curves with the threshold, indicating identifiable parameter ranges, while P_2, P_5, and P_6 exhibit non-intersecting, flatter curves, suggesting non-identifiability given the data used.

4.4.6 Discussion

The modified timing model (TM-M) demonstrates close alignment with the original timing model (TM-O) predictions. This congruence underscores the validity of the approach in representing complex intracellular signalling cascades as ODEs. However, discrepancies between TM-M outputs and TM-O data were observed, particularly in pSubstrate production during calcium input. These discrepancies, observed as overestimation in experiments $E_{0–3}$ and underestimation in $E_{5–9}$, suggest a fundamental limitation in the current model structure or parametrization, highlighting areas where the model could be further refined.

The transition from hand-tuned functions to ordinary differential equations (ODEs) for modelling CaMKII autophosphorylation has significantly improved the model's usability and accuracy. This methodological enhancement has not only streamlined the simulation process but also increased computational feasibility and ease of integration with various software tools, broadening the model's applicability and enabling more comprehensive hypothesis testing.

The parameter estimation process revealed the software's capability to fit TM-M to the data generated by TM-O across various experimental scenarios. The optimization algorithms successfully identified appropriate θ values, despite increased model complexity. However, the extended computational time required for good optimizations (from 900 seconds to 7200 seconds per algorithm) indicates potential scalability challenges for more complex models. The convergence towards an optimal θ as shown in figures C.4 and C.5 suggests the presence of a global minimum within the parameter space, providing confidence in the estimation process. The primary challenge in parameter estimation was fitting the pSubstrate data. This difficulty points to a potential area for model refinement or additional experimental data collection to better constrain θ.

As referenced before in section 4.4.1.2, figure 4.16, and table 4.6 the model defines six parameters to describe the association, dissociation and phosphorylation dynamics of CaMKII. P_1 represents the formation of a complex between two unphosphorylated CaMKII units. P_2 describes the dissociation of that same complex. P_3 represents the dissociation of the complex and the auto-phosphorylation of one its members. P_4 captures the binding of a phosphorylated CaMKII to an unphosphorylated CaMKII forming a different complex. P_5 describes the separation of this complex. P_6 represents the auto-phosphorylation of the second CaMKII in this complex.

Sensitivity analysis provided valuable insights into the model's behaviour. Local sensitivity analysis revealed that parameters P_1 and P_4 exhibit significant influence on the model's

output when dopamine precedes calcium ($E_{0\text{-}3}$), while only P_1 shows substantial impact when calcium precedes dopamine ($E_{5\text{-}8}$). This suggests that P_1 plays a crucial role across various experimental conditions, while P_4's influence is more context-dependent. The minimal impact of P_2, P_3, P_5, and P_6 near the optimal parameter set indicates that these parameters are either less critical for the system under study or not well constrained by the available data, indicating again a potential area for improvement in future model refinements. Global sensitivity analysis, on the other hand, showed that all parameters contributed to the overall sensitivity scores. This discrepancy suggests that the optimization process leads the model to a parameter set region that is robust to changes in most parameters except P_1 and P_4. The analysis also revealed complex interactions and dependencies among parameters. This emphasizes the importance of considering both individual and interdependent effects of changing or adding parameters when considering any further modifications to the model.

Profile likelihood analysis (PLA), applied to individual experiments, unveiled three identifiable (P_1, P_3, P_4) and three non-identifiable (P_2, P_5, P_6) parameters across all experiments. This analysis revealed distinct feasibility regions for parameters in experiments where dopamine precedes calcium ($E_{0–3}$) versus those where dopamine is simultaneous with, follows, or is absent relative to calcium input ($E_{4–9}$). The optimal values for P_1 in $E_{0–4}$ and P_4 in $E_{0–1}$ lie in regions where $E_{4–9}$ show worse scores, elucidating why achieving a perfect fit across all experiments is fundamentally challenging given the current model configuration and available data.

Future work should focus on addressing the limitations identified in this study. Refinement of ODE parameters based on observed discrepancies could enhance the model's biological accuracy. This might involve incorporating more complex kinetics or additional signalling pathways to better reflect the intricate processes of synaptic plasticity. Validation of the model with a broader range of experimental data, particularly in vivo measurements of calcium and dopamine dynamics, would further strengthen its predictive power.

The successful application of the software to TM-M demonstrates its robustness and adaptability in handling complex biological systems, further showing the software is a valuable tool for advancing computational neuroscience and exploring complex biological phenomena.

4.5 G-PIPK model

In this chapter, the G-PIPK model will serve as the final test for the developed software and will be studied in detail. It comprises various biochemical components, GPCR and G-

protein (G), PLC (P), PI/PIP/PIP2 (PIP), IP3 (IP), and Kv7.2/3 with its associated current (K) (G-PIPK), and it is the model with the largest number of unconstrained parameters among those investigated in this dissertation. Though its scale may be smaller than the previous model, because it has fewer total parameters, it is engineered for detailed analysis of all its parameters. This increase in free parameters renders it particularly challenging for study by computational methods.

The following sections will provide an overview of the G-PIPK model to elucidate its structure and functionalities. Following this, the model will be bound to existing empirical data using the parameter estimation software. Subsequently, sensitivity analysis and profile likelihood analysis will be shown. These latter analyses are intended not only to refine the fit of the estimated parameters but also to identify areas within the model that may benefit from future refinement and iteration.

At the end of this chapter, the G-PIPK model will be employed to predict the concentrations of specific molecular entities under conditions differing from those on which it was trained. This exercise aims to validate the model utility as a predictive tool, thereby highlighting its potential applications in furthering our understanding of biological systems.

4.5.1 Model overview

This model represents a GPCR signalling pathway, culminating in the dissociation of PIP_2 into IP_3 and DAG by PLC, as well as a current response given by the binding of PIP_2 to Kv7.2/3. The pathway is particularly relevant for neuronal signalling, specifically in neurons that co-express mGluR1/5 and M1R (D1 and D2 striatal neurons), and it serves as a mediator between extracellular signalling and intracellular responses [97, 107–110, 113, 116, 117, 214, 215].

The model captures the intricacy of a comprehensive biochemical system, ranging from ligand binding to the activation of downstream effectors, and it is regulated by a series of reactions labelled R_{1-35}. For more details, refer to figure 4.25 for a visual representation of the model or tables D.1 and D.2 for more details on each reaction parameters and their correspondence between the diagram, SBtab file and analysis.

This model combines multiple biological elements, rendering it computationally complex and valuable for the study of biological system models. In the following subsections, a systematic explanation of the model and its constituent elements will be provided. These elements have been organized to illustrate the sequence from ligand arrival to downstream signalling, thereby improving overall comprehension.

Assumptions related to each section are detailed next but there are some model limitations that do not pertain to any specific section, which should also be taken into account:

- The data is not perfectly up to date and only data from two other research groups was used [116, 440, 452, 453], noting that the most recent article used as a reference was published in 2010. Therefore the model behaviour and its insights could be improved by using more recent and more diverse data;
- A measure of current was used but the model has no measure of voltage. The current is estimated by a mathematical equation from the concentration of Kv7.2/3_PIP_2. A more realistic measure of current and voltage could be included in the future.

These limitations were not addressed because the aim of this thesis is to develop a testing platform for this software, rather than to push the state of the art of models depicting this specific biochemical pathway. Nonetheless, it could be used as a starting point for further analysis in the future.

4.5.1.1 Ligand-GPCR binding

This section contains the upstream molecular interactions of the model which are responsible for initiating downstream signalling, specifically the binding of a ligand to GPCRs. The ligands present in the model include Glu, DHPG, oxotremorine M (OxoM), and ACh, the first two are shown to bind to mGluR1/5 and the latter to M1R receptors.

The molecular dynamics between ligands and GPCRs are coded into reactions R_{1-4} (yellow section in figure 4.25). These reactions represent the processes of ligand binding and subsequent unbinding from the receptor when the receptor itself is connected to different protein assemblies. Empirical data, specifically measurements detailing the binding affinity of glutamate to mGluR-5 and OxoM to M1R, have been documented in the referenced literature [116, 440, 452, 453].

Assumptions:

- The ligand binding rates to the receptor are the same independently of the presence of other proteins in the receptor complex, while the unbinding rates differ for each of them in order to obey thermodynamic constrains. Mathematically represented as: $R_{1f} = R_{2f} = R_{3f} = R_{4f}$;
- The binding affinity of DHPG to mGluR1/5 is analogous to that of OxoM to M1R, and similarly, the binding affinity of glutamate to mGluR1/5 mirrors that of ACh to M1R.

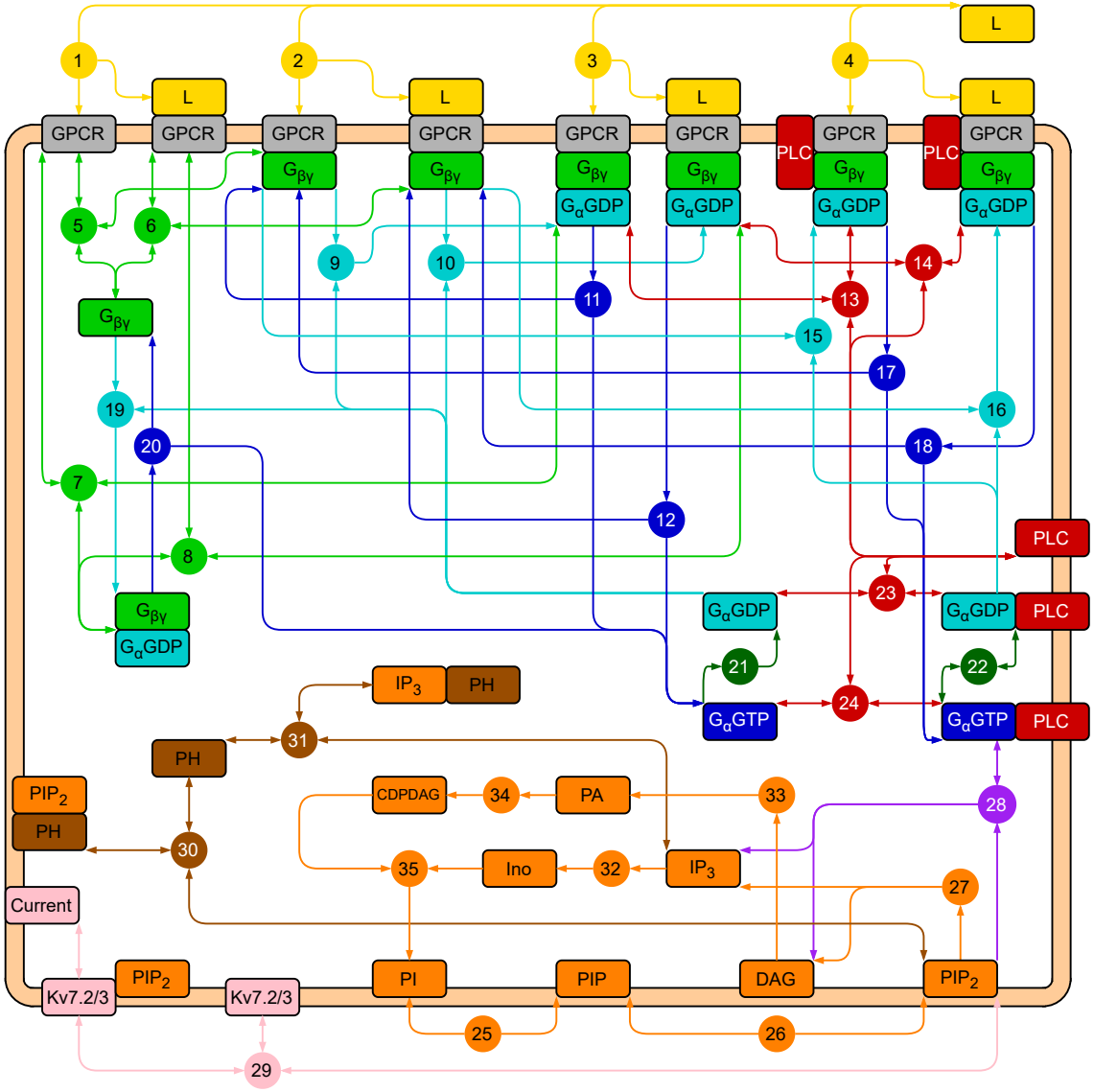


Figure 4.25: G-PIPK network diagram of the GPCR signalling pathway. This diagram presents the intricate cascade of reactions within the G-PIPK model, outlining the progression from ligand-receptor interactions to varied downstream effects. The network illustrates the sequence of reactions labelled R_{1-36}, with emphasis on the key components such as GPCR, G-protein, PLC, and the PI cycle. The color-coding is specifically mapped to model sections: gold for ligand species and ligand-GPCR binding (section 4.5.1.1); grey for GPCR species; light green for $G_{\beta\gamma}$ species and their binding/unbinding to GPCR (section 4.5.1.2); light and dark blue respectively for G-protein α domain (G_α)guanosine diphosphate (GDP) and $G_\alpha$guanosine-5'-triphosphate (GTP) species and their interactions with $G_{\beta\gamma}$ (section 4.5.1.3); dark green for G_αGTP to G_αGDP conversion (section 4.5.1.4); red for G_α-PLC signalling (section 4.5.1.5); purple for the hydrolysis of phosphatidylinositols by PLC (section 4.5.1.6); orange for the phosphatidylinositol cycle (section 4.5.1.7); pink for PIP_2-Dependent Kv7.2/3 membrane current (section 4.5.1.8); brown for real-time monitoring with pleckstrin homology (PH) probes for IP_3 and PIP_2 (section 4.5.1.9).

4.5.1.2 GPCR-G-protein coupling

In this section, the subsequent molecular interactions following ligand binding are delineated, specifically the coupling of GPCRs to $G_{\beta\gamma}$. Upon ligand interaction, GPCRs undergo a conformational change, transitioning from an inactive to an active state and facilitating their interaction with intracellular $G_{\beta\gamma}$ units. This pivotal point translates signals from the extracellular medium into intracellular cascades [113, 116, 117, 215].

The described interaction process is present in the model as reactions 5-8. These reactions are assumed to be sufficient to capture the foundational dynamics and kinetics of this coupling.

Assumptions:

- The $G_{\beta\gamma}$ domain associates with GPCR with consistent affinity, irrespective of the presence or absence of the G_{α} domain. Mathematically expressed as: $R_5 = R_7$ and $R_6 = R_8$;
- The coupling mechanisms for mGluR1/5, and M1R are assumed to be identical, allowing for generalizations in the context of GPCR-G-protein dynamics.

4.5.1.3 G_{α} interaction with $G_{\beta\gamma}$

This section contains the reactions that control the dynamics of G_{α} (GDP or GTP) connection to and separation from $G_{\beta\gamma}$, a critical aspect of G-protein activation and signalling. This interaction can occur when the G-protein complex is by itself or connected to the GPCR. Once GPCRs are activated and coupled with G-proteins, the G_{α} subunit undergoes conformational changes that lead to its separation from the $G_{\beta\gamma}$ complex as G_{α}GTP. This separation is essential for enabling the G_{α} and $G_{\beta\gamma}$ subunits to interact with their respective downstream targets, setting off a cascade of cellular responses. In this model only the downstream signalling of the G_{α}GTP is considered [454].

The model represents G_{α}GDP in both bound and unbound states with $G_{\beta\gamma}$, while G_{α}GTP is present only in the unbound state. The G-protein activation cycle is captured by simulating G_{α} binding to $G_{\beta\gamma}$ as G_{α}GDP and unbinding as G_{α}GTP.

The reactions included in this section are R_{9-12} and R_{15-22}, encompassing all feasible possibilities for G_{α} (GDP and GTP) interaction with $G_{\beta\gamma}$ when it is either free or coupled to other species.

Assumptions:

- For G_{α}GDP interaction with $G_{\beta\gamma}$:

- Both reactions where it binds to $G_{\beta\gamma}$ as part of a receptor-ligand complex, have a consistent rate constant represented as $R_{10} = R_{16}$;
- The reactions where it binds to $G_{\beta\gamma}$ either attached to the receptor without a ligand or entirely free from the receptor, also exhibit a consistent rate constant denoted by $R_9 = R_{15} = R_{19}$;

- For G_αGTP interaction with $G_{\beta\gamma}$:
  - Same as for G_αGDP but for separation, in summary, $R_{12} = R_{18}$ and $R_{11} = R_{17} = R_{20}$.

4.5.1.4 G_αGTP to G_αGDP conversion

This section incorporates the transformation of the G_α subunit from its GTP-bound form to its GDP-bound form. This enzymatic conversion is responsible for terminating the G-protein signalling process and restoring the G-protein to its inactive state, preparing it for another cycle [454]. The conversion process often involves intrinsic GTPase activity or is facilitated by GTPase-activating proteins (GAPs) [454]. The G_αGTP to G_αGDP conversion is mapped to reactions $R_{21,22f}$.

Assumption:

- The reaction rate of the decay of free G_αGTP is distinct from G_αGTP bound to PLC.

4.5.1.5 G_α interaction with PLC

This section pertains to the connection of G_α with PLC and the subsequent activation of PLC. In this model, PLC binds to G_αGTP when it is free or bound to a GPCR. PLC is only considered to be activated when it is bound to free G_αGTP; in this conformation it is capable of participating in the next reactions in the cascade contained in the following section [215]. Apart from this, PLC can also promote the conversion of G_αGDP to G_αGTP even without GPCR.

The molecular kinetics of the G_α PLC interaction are encapsulated in reactions $R_{13,14}$ (bound G_α), $R_{23,24}$ (free G_α) and R_{22r} (G_αGDP to G_αGTP).

Assumption:

- The rates of PLC binding to and unbinding from G_αGDP are assumed to be constant and independent of the binding partner of G_αGDP (GPCR-$G_{\beta\gamma}$ or Ligand-GPCR-$G_{\beta\gamma}$). Mathematically: $R_{13} = R_{14} = R_{23}$.

4.5.1.6 Hydrolysis of PIP_2 by PLC

The reactions of hydrolysis of PIP_2 by activated PLC are included in this section. PLC hydrolyses membrane-bound PIP_2 molecules into two potent secondary messengers, diacylglycerol (DAG) and inositol 1,4,5-trisphosphate (IP_3) [216, 217]. DAG remains in the plasma membrane while IP_3 diffuses into the cytosol, and both of these molecules are responsible for further downstream cascade events that are not present in this model. The model represents this process by reaction R_{28} [455].

4.5.1.7 PI cycle

In this section, the biochemical modifications of PIs are delineated, specifically focusing on the phosphorylation and dephosphorylation cycles that yield various phosphatidylinositol species [456]. Initially, PI undergoes phosphorylation to become PIP. Further phosphorylation of PIP results in the formation of PIP_2 [456]. These phosphorylation reactions are mirrored by their corresponding dephosphorylation processes, whereby PIP_2 is converted back to PIP, and subsequently, PIP is dephosphorylated to regenerate PI [456]. Moreover, PIP_2 is enzymatically cleaved to produce IP_3 and DAG in the presence or absence of PLC [456]. The compound DAG undergoes successive enzymatic conversions, first to PA, then to CDPDAG, while IP_3 is metabolised to Ino [456]. Ino combines with CDPDAG to synthesize PI, thus completing the cycle [456]. These biochemical reactions are encoded within the model as $R_{25–27}$ and $R_{32–35}$.

Assumptions:

- The hydrolysis of PIP_2 to DAG and IP_3, is presumed to occur in the absence of PLC;
- The multiple enzymatic reactions converting IP_3 to Ino are consolidated into a singular reaction within the model for simplicity and coherence.

4.5.1.8 PIP_2-dependent Kv7.2/3 membrane current

This section elaborates on the interaction between PIP_2 and Kv7.2/3 ion channels (represented in the model as KCNQ derived from its gene name), a point connecting the intracellular biochemical processes and the electrophysiological properties of the cell membrane. The Kv7.2/3 channel, which has a high sensitivity to PIP_2 concentrations, serves as the modulator of membrane current in this model. This mechanistic feature is incorporated into the model as reaction R_{29}. As an output point of the model, this section also serves as a

source of model data that can be more easily compared to available experimental data. This model section was derived from [440].

Assumptions:

- The affinity of PIP_2 to Kv7.2/3 is directly proportional to alterations in membrane currents;
- The distribution of PIP_2 concentrations across the membrane is postulated to be uniform in regions where Kv7.2/3 channels are situated.

4.5.1.9 Real-time monitoring: PH probes for IP_3 and PIP_2

This final section delineates the application of advanced monitoring techniques in the model for tracking intracellular concentrations of IP_3 and PIP_2. Specifically, PH probes serve as the technological medium for real-time quantification of these signalling molecules. The real-time tracking is implemented in the model as reactions $R_{30,31}$. Empirical evidence regarding the specificity and efficiency of these probes in tracking IP_3 and PIP_2 is provided in the cited literature. This section is invaluable for comparing model behaviour to experimental data of real cells as it contains literal probes whose concentration can be measured in the model and observed in the lab. This model section was derived from [440].

Assumptions:

- The PH probes are assumed to possess high specificity and sensitivity towards IP_3 and PIP_2, with negligible interference from other intracellular constituents;
- A uniform distribution of PH probes within the cellular matrix is assumed, ensuring the reliability and consistency of real-time measurements.

4.5.2 Experimental data for model calibration and validation

The calibration and validation of the G-PIPK model are underpinned by a comprehensive set of experimental data, carefully selected to encompass the full range of biological phenomena represented in the model. This subsection provides an overview of these datasets, highlighting their crucial role in parameter estimation and model validation. The data used spans multiple levels of the signalling cascade, from initial ligand-receptor interactions to downstream effector responses, allowing for a thorough constraint of the model across various operational regimes.

The experimental data used in this study can be broadly categorized into two types: kinetic data and Hill curve data. Kinetic data elucidates the temporal dynamics of the system,

indicating how rapidly the measured outputs respond to and recover from given inputs. Hill curve data, on the other hand, characterizes the dose-response relationship, measuring how the amplitude of the system's response changes with increasing input concentrations. Together, these data types provide a robust foundation for constraining the model across various operational regimes. table 4.7 summarizes the key features of all experiments used in this study.

Experiment	Input	Output	Data	Constrained aspect
$E_{0,1}$	Glu/DHPG	mGluR binding	Kinetic/ Hill	Ligand-GPCR binding
$E_{2,3}$	OxoM/ACh	M1R binding	Kinetic	Ligand-GPCR binding
$E_{4–7}$	OxoM/ACh	G-protein coupling	Kinetic/ Hill	G-protein activation
$E_{8–11}$	OxoM/ACh	G_{α}-PLC interaction	Kinetic/ Hill	Effector coupling
$E_{12–15}$	OxoM/ACh	PIP_2 hydrolysis/ Kv7.2/3 current (+PH)	Kinetic/ Hill	Downstream signalling/ membrane response
$E_{16–19}$	OxoM/ACh	Kv7.2/3 current	Kinetic/ Hill	Membrane response
$E_{20,21}$	OxoM/ACh	Kv7.2/3 current (+PLC)	Kinetic	Membrane response
E_{22}	None	Basal lipid levels	Steady-state	Resting conditions

Table 4.7: G-PIPK experimental design used for calibration and validation. This table summarizes the key aspects of each experiment ($E_{0–22}$) used to constrain and validate the G-PIPK model. It outlines the input ligands, measured outputs, types of data collected (kinetic, Hill curve, or steady-state), and the specific aspects of the signalling pathway constrained by each experiment. This comprehensive set of experiments spans the entire signalling cascade, from initial ligand-receptor binding to downstream membrane responses, providing a robust framework for model parametrization and validation. Figures D.2 to D.11 represent the results of experiments $E_{0\text{-}22}$, respectively.

To ensure comprehensive coverage of the signalling pathway, from initial ligand-receptor interactions to downstream effects, the experiments are organized into three main groups:

1. Ligand-GPCR binding dynamics;

2. G-proteincoupling and interactions;

3. Downstream signalling and membrane response.

The data used in this model was sourced from key publications in the field [116, 440, 453, 457]. To further constrain the model and fill gaps in the available experimental data, artificially generated data was produced based on the real data. Even-numbered experiments utilize real data, while odd-numbered experiments employ artificial data, allowing for a more comprehensive exploration of the model parameter space and behaviour.

The initial concentrations of key molecular species were carefully constrained based on published literature to ensure biological relevance. For experiments $E_{2–11}$, the concentrations of M1R, G-protein, and PLC were derived from Table 2 of Falkenburger *et al*. (2010) [457]. Experiments $E_{12–22}$ utilized a more comprehensive set of initial concentrations, including M1R, G-protein, PLC, PH, PI, and Kv7.2/3, sourced from Table 1 of Falkenburger *et al*. (2010) [453]. In the case of experiments $E_{0,1}$, where no specific data for mGluR concentrations were available in the literature, its concentration was set to match the over-expressed protein levels used in E_2, maintaining consistency across the experimental series.

To facilitate future extrapolation studies investigating the simultaneous effects of glutamate and ACh on the signalling pathway, mGluR and M1R were expressed at equivalent levels across all experiments. This design choice allows for a more straightforward comparison of the two receptor systems and their downstream effects, while also providing a foundation for exploring potential cross-talk between glutamatergic and cholinergic signalling in future iterations of the model.

The following subsections provide detailed descriptions of each experimental set, including the specific aspects of the model that they constrain or validate. This thorough examination of the experimental data establishes a high degree of confidence in the model's ability to accurately represent the intricate dynamics of the G-PIPK signalling pathway across a wide range of conditions.

4.5.2.1 Ligand-GPCR binding dynamics

The foundation of the G-PIPK signalling cascade lies in the initial interaction between ligands and their corresponding GPCRs. Experiments $E_{0–3}$ focus on these crucial first steps of the signalling process, providing essential data for calibrating the model's representation of receptor activation.

- $E_{0,1}$: These experiments examine the kinetics of ligand (glutamate (E_0) or DHPG (E_1)) binding to GPCR (mGluR1/5). Each experiment consists of 1-second ligand impulses at increasing concentrations (2, 5, 1×10, 2×10^2, 2×10^3, 2×10^4 μM for glutamate, and 100-fold lower for DHPG), separated by 1-second intervals. The outputs measured to refine the Ligand-GPCR binding module are: the total amount of MGluR bound to Glu (MGluR_Glu_T), the normalized version of MGluR_Glu_T (MGluR_Glu_T_Norm), the total amount of MGluR bound to DHPG (MGluR_DHPG_T), and the normalized version of MGluR_DHPG_T (MGluR_DHPG_T_Norm). Hill curve and kinetics data are derived from figures 1 and 2G/H of Marcaggi *et al*. [116]. The model is initialized with 1494.5 nM of mGluR and M1R, reflecting over-expression conditions similar to Jensen *et al*. [440], with all other species set to 0. Figure D.2 shows the inputs, experimental data with standard deviation, and model outputs for two optimized θ;

- $E_{2,3}$: These experiments examine the kinetics of ligand (OxoM (E_2) or ACh (E_3)) binding to GPCR (M1R). Each experiment consists of 10-second pulses of ligand, OxoM (E_2) at a concentration of 10 μM or ACh (E_3) at a concentration of 1000 μM, starting at 1 second and ending at 11 seconds. The outputs measured to refine the Ligand-GPCR binding module are: the total amount of M1R bound to ACh (M1R_ACh_T), the normalized version of M1R_ACh_T (M1R_ACh_T_Norm), the total amount of OxoM bound to M1R (M1R_OxoM_T), and the normalized version of M1R_OxoM_T (M1R_OxoM_T_Norm). Data are sourced from figure 1 (interactions) and table 1 (kinetics) of Jensen *et al*. [440]. The model is initialized with 1494.5 nM of mGluR and M1R, reflecting over-expression conditions similar to Jensen *et al*. [440], and 5 nM of PLC with all other species set to 0. Figure D.3 shows the inputs, experimental data with standard deviation, and model outputs for two optimized θ;

Together, these experiments provide crucial constraints on the model's ligand-receptor interaction parameters, enabling accurate representation of the initial steps in the signalling cascade. The use of both glutamatergic and cholinergic ligands allows for comparative analysis of these two important neurotransmitter systems within the same model framework.

4.5.2.2 G-protein coupling and interactions

Following ligand binding, the next critical step in the signalling cascade involves the coupling of activated GPCRs with G-proteins and the subsequent interactions between G-protein subunits. Experiments $E_{4–11}$ probe these molecular events, providing data on the dynamics

of G-protein activation, subunit dissociation, and interactions with downstream effectors like PLC.

- $E_{4–7}$: This set of experiments analyses the coupling dynamics between GPCR and G-protein, as well as the formation of G-protein trimers, using pulses of OxoM and ACh. In $E_{4,5}$, a 10-second impulse of OxoM (10 μM) and ACh (1000 μM), respectively, is applied to measure a fast kinetic response. $E_{6,7}$ use multiple impulses at varying concentrations (1, 1×10^1, 1×10^2, 1×10^3, 1×10^4, 5×10^4 nM for OxoM, and concentrations 100 times larger for ACh) to represent a Hill curve. The outputs measured include, the total amount of M1R bound to $G_{\beta\gamma}$ (M1R_Gbc_T), the normalized version of M1R_Gbc_T (M1R_Gbc_T_Norm), the total amount of G-protein trimer concentration (Gabc_T) and the normalized version of Gabc_T (Gabc_T_Norm), with data sourced from Jensen *et al*. [440]. The model is initialized with 1494.5 nM of mGluR and M1R, 1495.5 nM of G-protein, and 5 nM of PLC, with all other species set to 0. Figures D.4 and D.5 depict the inputs, experimental data and its standard deviation, and experimental outputs given two optimized θ, for these four experiments;

- $E_{8–11}$: These experiments analyse the interaction between G_α and PLC using pulses of OxoM and ACh. In $E_{8,9}$, a 10-second impulse of OxoM (10 μM) and ACh (1000 μM), respectively, is applied to measure a fast kinetic response. Experiments 10 (E_{10}) and 11 (E_{11}) use multiple impulses at varying concentrations (1×10^1, 1×10^2, 1×10^3, 1×10^4, 5×10^4 nM for OxoM, and concentrations 100 times larger for ACh) to assess the influence of different input amounts on the Hill curve. The outputs measured include the total amount of G_α bound to PLC (Ga_PLC_T) and the normalized version of Ga_PLC_T (Ga_PLC_T_Norm), with data sourced from Jensen *et al*. [440]. The model is initialized with 1494.5 nM of mGluR, M1R, and PLC, and G-protein, with all other species set to 0. Figure D.6 depicts the inputs, experimental data and its standard deviation, and experimental outputs given two optimized θ, for these four experiments.

The data from these experiments, sourced from Jensen *et al*. [440], provide essential constraints on the model's representation of G-protein cycle dynamics and effector coupling. By examining both fast kinetic responses and dose-dependent behaviours, these experiments enable the model to capture both the rapid signalling events and the more gradual accumulation of active signalling complexes.

4.5.2.3 Downstream signalling and membrane response

The final set of experiments $E_{12–20}$ focuses on the downstream effects of G-protein activation, particularly the modulation of PIP_2 levels and its impact on membrane currents. These experiments are essential for calibrating the model's representation of the complex interplay between signalling molecules and cellular responses, ultimately linking receptor activation to physiological outcomes.

- $E_{12–15}$: These experiments analyse the hydrolysis of PIP_2 by PLC and its subsequent reconstitution given OxoM and ACh pulses. In $E_{12,13}$, a 40-second impulse of OxoM (10 μM) and ACh (1000 μM), respectively, is applied to measure fast kinetic responses. $E_{14,15}$) use multiple impulses at varying increasing concentrations (1, 1×10^1, 1×10^2, 1×10^3, 1×10^4, 5×10^4 nM for OxoM, and concentrations 100 times larger for ACh) over 1000 seconds to assess dose-response relationships. The outputs measured include the total amount of PH probes bound to PIP_2 (PIP2_PH_T), the normalized version of PIP2_PH_T (PIP2_PH_T_Norm), the PIP_2-dependent Kv7.2/3 membrane current (M_current), and the normalized version of M_current (M_current_Norm). M_current is calculated as $(KCNQ + PIP2_KCNQ) \times 0.5 \times 0.055 \times PIP2_KCNQ^2$. Data for PIP2_PH and M_current are sourced from figure 5 (Hill curve and kinetics), figure 7B +PH (kinetics) and table 1 (kinetics) of Jensen *et al*. [440]. These experiments provide crucial constraints for the model behaviour in terms of PIP_2 hydrolysis, reconstitution, and its effects on membrane current in a system with over-expressed PH. The model is initialized with 249.1 nM of mGluR and M1R, 19.9 nM of G-protein, 5 nM of PLC, 74386 nM of PI, 2989 nM of PH, and 2 nM of Kv7.2/3, with all other species set to 0. Figures D.7 and D.8 depict the inputs, experimental data and its standard deviation, and experimental outputs given two optimized θ, for these four experiments.

- $E_{16–19}$: These experiments focus on the Kv7.2/3 membrane current response to agonist stimulation. In $E_{16,17}$, a 20-second impulse of OxoM (10 μM) and ACh (1000 μM), respectively, is applied to measure fast kinetic responses. $E_{18,19}$ use multiple impulses at varying increasing concentrations (1, 1×10^1, 1×10^2, 1×10^3, 1×10^4, 5×10^4 nM for OxoM, and concentrations 100 times larger for ACh) over 1000 seconds to assess dose-response relationships. These experiments measure M_current and M_current_Norm. The data, sourced from figure 6 (Hill curve and kinetics) and table 1 (kinetics) of Jensen *et al*. [440], provide crucial constraints for the model's output under physiological conditions. By examining the kinetic response without PLC or PH

over-expression, $E_{16,17}$ establish a baseline for understanding how changes in PLC (as in $E_{18,19}$) or PH (as in $E_{12,13}$) levels affect signalling dynamics and Kv7.2/3 current modulation. The model is initialized with 249.1 nM of mGluR and M1R, 19.9 nM of G-protein, 5 nM of PLC, 74386 nM of PI, and 2 nM of Kv7.2/3, with all other species set to 0. Figure D.9 depicts the inputs, experimental data and its standard deviation, and experimental outputs given two optimized θ, for these experiments.

- $E_{20,21}$: These experiments focus on the Kv7.2/3 membrane current response to agonist stimulation, OxoM and ACh in E_{20} and E_{21} respectively, under conditions of PLC over-expression. In $E_{20,21}$, a 20-second impulse of OxoM (10 μM) and ACh (1000 μM), respectively, is applied to measure fast kinetic responses. Both experiments measure M_current and M_current_Norm over time. The data, sourced from figure 7A +PLC (kinetics) and table 1 (kinetics) of Jensen *et al*. [440], provide crucial constraints for the model output under conditions of enhanced PLC activity. By examining the response with PLC over-expression, these experiments reveal how increased enzyme levels affect signalling dynamics and Kv7.2/3 current modulation, complementing the baseline established in $E_{16,17}$. The model is initialized with 249.1 nM of mGluR and M1R, 19.9 nM of G-protein, 1494.5 nM of PLC, 74386 nM of PI, and 2 nM of Kv7.2/3, with all other species set to 0. Figure D.10 depicts the inputs, experimental data and its standard deviation, and experimental outputs given two optimized θ, for these two experiments;

- E_{22}: This experiment measures the basal concentrations of PI, PIP, total PIP_2, total IP_3, PH bound PIP_2, PH bound IP_3 and free PH over 1000 seconds without any input. The measured outputs are: the total amount of PIP_2 (PIP2_T), the total amount of IP_3 (IP_3_T), PIP2_PH_T, the total amount of PH probes bound to IP_3 (IP_3_PH_T), and the total amount of PH probes (PH_T). It establishes the steady-state levels of these key signalling lipids and second messengers under resting conditions, providing a baseline for the model. Data from Falkenburger *et al*.[453] offer quantitative insights into basal lipid dynamics, crucial for accurately modelling the system's response to stimulation in other experiments. This baseline data constrains the model behaviour in terms of constitutive lipid metabolism and turnover, enhancing our understanding of the underlying cellular processes. The model is initialized with 249.1 nM of mGluR and M1R, 19.9 nM of G-protein, 5 nM of PLC, 74386 nM of PI, 2989 nM of PH, and 2 nM of Kv7.2/3, with all other species set to 0. Figure D.11 depicts the inputs, experimental data and its standard deviation, and experimental outputs given two optimized θ, for this experiment.

This carefully curated set of experimental data, spanning from initial receptor activation to downstream physiological responses, provides a robust foundation for the development and validation of the G-PIPK model. By constraining multiple aspects of the signalling cascade, these experiments enable the model to capture the complex dynamics of G-protein mediated signalling with high fidelity.

4.5.3 Model analysis

The analysis of the G-PIPK model presented unique challenges due to its high complexity and large number of free parameters. With 35 free parameters, compared to 6 in TM-M, 12 in LiREM, and 7 in LiRMo, finding an optimal parameter set (θ) required a more sophisticated approach than the single parameter estimation routine performed for the previous models.

To address this complexity, three different optimization rounds where performed. This approach aimed to increase the odds of finding good θ to be used for the subsequent model prediction in section 4.5.8. The first consisting of one genetic algorithm run followed by a pattern search run. The second consisting of an ensemble of genetic algorithm runs followed by pattern search runs. And the third also consisting of an ensemble of genetic algorithm runs followed by pattern search runs but with parameter bounds constrained by the bounds obtained by constraining the results of the first PLA below 10^3 for each parameter.

The results of these three parameter estimations can be seen in the following figures: figure 4.26 showing a summary of where the model obtained best fits with each θ and figures D.2 to D.11 showing all the experimental outputs of the model for all experiments with all θ.

Following the optimization process, additional analyses were performed to further characterize the model, as done for the previous examples: a PLA based on the best θ from each parameter estimation. A local sensitivity analysis around the best θ from the first estimation. A global sensitivity analysis using θ bounds obtained by constraining the results of the first PLA below 10^3 for each parameter. The use of refined bounds in the global sensitivity analysis was necessary to increase the proportion of successful simulations per sample size. This adjustment was made because the ODEs of this model often encountered integration tolerance problems in regions far from an optimized θ.

This comprehensive analytical approach aimed to thoroughly explore the parameter space, identify optimal parameter sets, and characterize the model's sensitivity to parameter variations. The following sections will detail the results of each step in this analysis sequence.

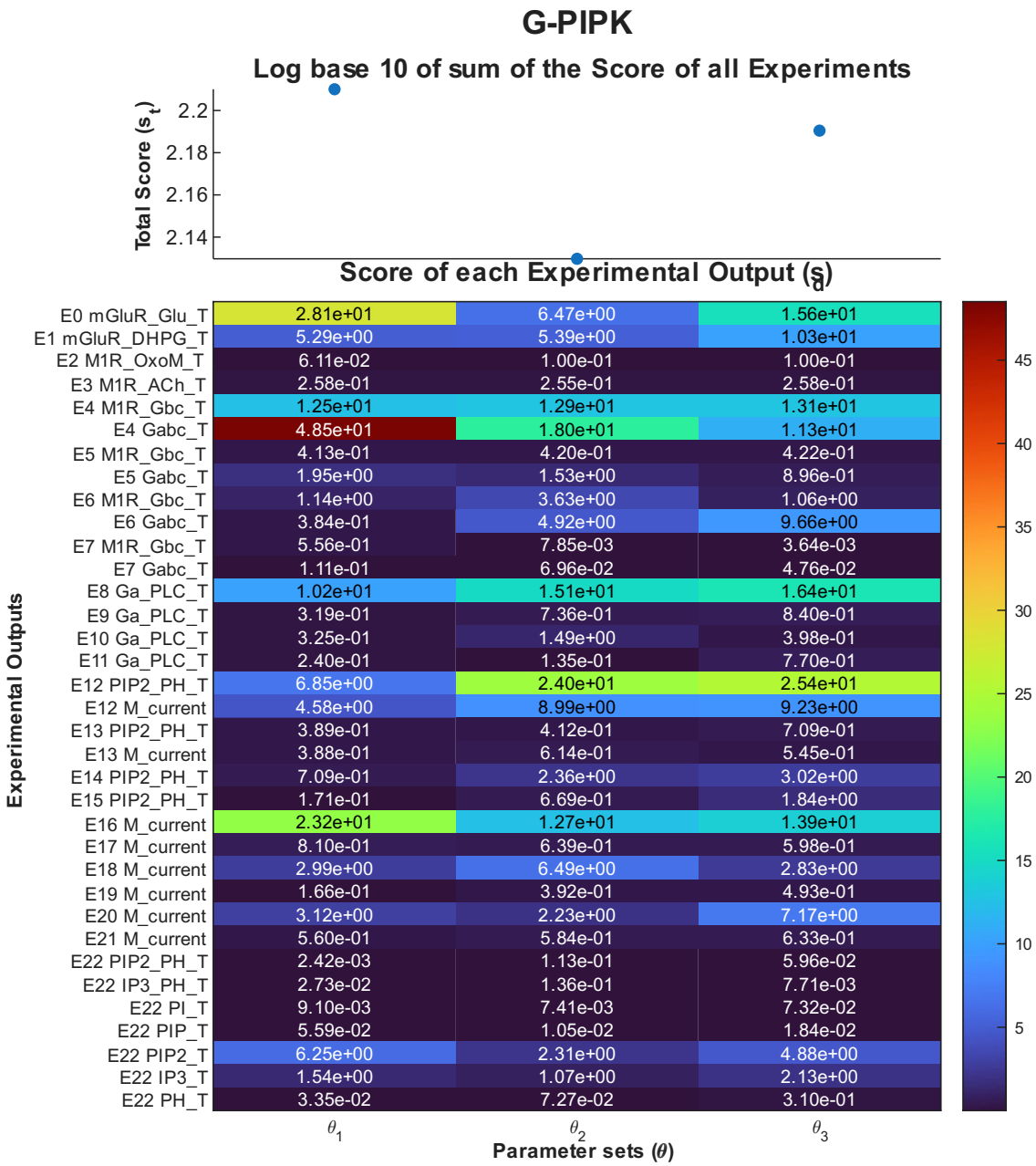


Figure 4.26: G-PIPK diagnostic scores plot. This figure synthesizes the performance of two parameter sets (θ_1 and θ_2), derived from two rounds of parameter estimation, across twenty three experimental conditions ($E_{0–22}$). The top panel displays the total score summed across all experiments for each parameter set, with lower scores indicating better overall model fits. The heatmap below shows individual scores for each experimental output. Each row represents an output variable for a given experiment (e.g., MGluR_Glu_T for E_0, M1R_OxoM_T for E_2, etc.). The color gradient reflects the scoring metric, with cooler colors (blues) indicating lower scores (better fits) and warmer colors (reds) denoting higher scores (poorer fits). Experiments and their respective outputs are as described in section 4.5.2. This visualization allows for quick comparison of model performance across different experimental conditions and between the two rounds of parameter estimation.

4.5.4 Parameter estimation

This subsection demonstrates the capability of the developed software to fit the G-PIPK model to the experimental data defined in section 4.5.2. Given the complexity of the G-PIPK model, with its 35 free parameters, three attempts were made to improve the probability of finding a suitable θ for the model.

For this model, the genetic algorithm was exclusively employed due to its proven effectiveness in handling complex parameter spaces. The first optimization results were obtained by running a single genetic optimization algorithm for 4 days, the second were obtained by expending a similar amount of compute but by dividing it by 216 optimization runs (both of these optimizations used an Intel i9-10980XE processor), and the third following a similar procedure to the second but using parameter bounds informed by the PLA performed on the θ provided by the first optimization results (parameter bounds constrained by the bounds obtained by constraining the results of the first PLA below 10^3 for each parameter).

The first optimization, as shown in figure 4.27, achieved a score of 162.20. The second optimization resulted in an improved score of 134.81, as depicted in figure D.12 (this figure provides a direct comparison between the first and second optimization rounds, showing how parameter estimates changed between the two). The third optimization resulted in an intermediate score of 155.05 as depicted in figure D.13 (this figure also provides a direct comparison between the the first and third optimization rounds, showing how parameter estimates changed between the two, while also showing the constrained parameter bounds).

Notably, some parameters remained nearly identical between all 3 estimations, suggesting they are well-constrained by the available data. This consistency provides confidence in these particular parameter values. However, it's important to note that, as discussed further, none of these should be considered the global optimum of the model.

A detailed examination of the model outputs reveals varying degrees of fit quality across different experimental conditions. The fits can be broadly categorized into three groups:

- Outputs that can be considered well-fitted. These outputs show excellent visual matches and very low scores, often with values below 0.1;
- Outputs that are almost satisfactory. While not perfect, these outputs show good agreement with data in most aspects, with scores typically below or around 5;
- Outputs that need improvement. These outputs show noticeable discrepancies with the experimental data, often in specific regions of the response curves, with scores typically above 10.

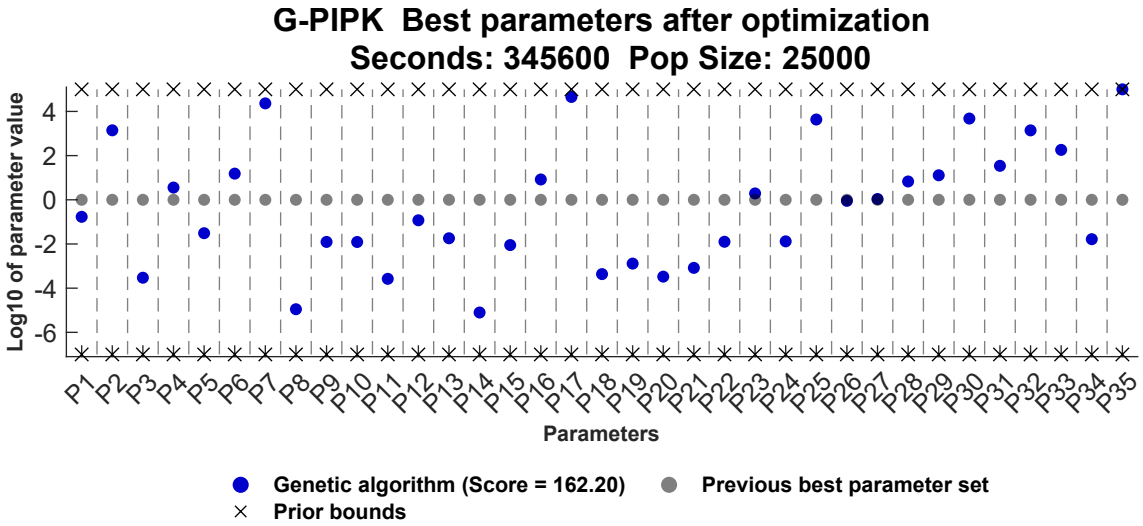


Figure 4.27: G-PIPK first optimized θ. This figure illustrates the results of parameter optimization for the G-PIPK model, showing the results of the first round of optimization. Each vertical line represents one of the model's free parameters (P_{1-35}). Blue dots show the best parameter values (θ) obtained from the first parameter estimation using the genetic algorithm, which achieved a score of 162.20. Grey dots indicate the best parameter values from a previous not shown fast optimization (less than one hour). Black crosses denote the upper and lower bounds for each parameter. The y-axis represents the $\log_{10}$ of parameter values, allowing comparison across different scales. This optimization was computationally intensive, running for 345600 seconds (4 days, run on an Intel i9-10980XE processor) with a population size of 25000 in the genetic algorithm.

Figure 4.26 shows the scores of all the experimental outputs given the three θ. Each θ struggles with different model outputs suggesting either that the obtained θs are not near a global optimum or that the fit of all the outputs at the same time is impossible.

It's worth noting that the visual quality of fits as seen in figures D.2 to D.11 does not always correlate directly with the numerical scores due to variations in the standard deviations of the experimental data. Experiments with data taken directly from articles (E_{0-22}, even-numbered experiments) generally have smaller standard deviations, resulting in higher scores even for visually good fits. Conversely, experiments with generated data to fill gaps (E_{1-21}, odd-numbered experiments) have larger standard deviations, often leading to better scores despite potential visual discrepancies.

The persistence of suboptimal fits for certain experimental outputs, even given three optimization attempts, could be attributed to several factors:

- The need for additional parameter estimation rounds to reach a better minimum;
- Potential limitations in the current model structure that prevent accurate representation of all experimental conditions simultaneously;
- Inconsistencies in the sourced data, particularly between experimentally-derived and

artificially-generated data sets;

- The need to revisit some of the model assumptions, especially those relating to the more poorly fitted outputs.

These results underscore both the power of the developed software in handling complex biochemical models and the challenges inherent to fitting models to diverse experimental datasets. They also highlight the iterative nature of model development and optimization, suggesting avenues for future refinement of the G-PIPK model. Future work could focus on addressing the poorly fitted outputs, potentially through targeted experimentation or refinement of the model structure in specific areas.

4.5.5 Sensitivity analysis

Sensitivity analysis and profile likelihood analysis were used to find regions of the model that need more data or should have the model structure improved.

4.5.5.1 Local sensitivity analysis

The local sensitivity analysis for the G-PIPK model provides insights into the relative importance of different parameters near the optimal parameter set. As shown in figure 4.28, the analysis reveals a diverse range of parameter sensitivities, which can be categorized based on their mean absolute elementary effects ($\mu^*_{d_p}$) and standard deviations of elementary effects (σ_{d_p}).

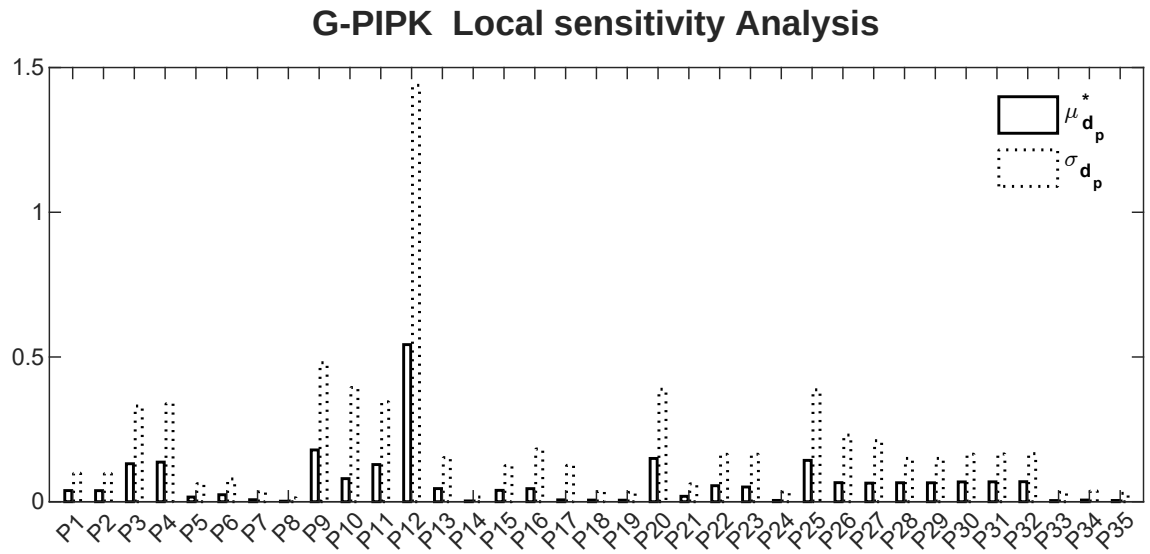


Figure 4.28: G-PIPK LSA. σ_{d_p} and $\mu^*_{d_p}$ are calculated for P_{1-35} as shown in equations (3.15) and (3.16), respectively, with LiRMo s_t as $f(\theta)$. Interpretations for the values of σ_{d_p} and $\mu^*_{d_p}$ are explained in section 3.6.1.

Parameters can be broadly classified into three groups based on their $\mu^*_{d_p}$:

- Large effect: P_{12} stands out with the highest $\mu^*_{d_p}$, indicating it has the most substantial influence on the model's output near the optimal parameter set;
- Medium effect: $P_{3,4,9,10,11,20,26}$ show moderate $\mu^*_{d_p}$ values, suggesting they have a significant, though lesser, impact on the model's behaviour;
- Small effect: $P_{1,2,13,15,16,22,23,26\text{-}32}$ exhibit small but noticeable $\mu^*_{d_p}$ values, indicating a minor influence on the model's output;
- Negligible effect: The remaining parameters show very low $\mu^*_{d_p}$ values, suggesting they have minimal impact on the model's behaviour near the optimal parameter set.

Interestingly, for almost all parameters, the σ_{d_p} is larger than the $\mu^*_{d_p}$. According to the interpretations outlined in section 3.6.1, this suggests that most parameters, including those with small or negligible $\mu^*_{d_p}$, exhibit non-linear effects and/or interactions with other parameters. This is particularly pronounced for P_{12}, which shows both the highest $\mu^*_{d_p}$ and σ_{d_p}, indicating it not only has the strongest influence on the model output but also the most complex interactions or non-linear effects.

The high σ_{d_p} values across most parameters suggest that the G-PIPK has a complex parameter landscape with numerous interdependencies. This complexity highlights the importance of global sensitivity analysis and other advanced techniques to fully understand the behaviour of the model across its entire parameter space.

These results provide valuable insights for future model refinement and experimental design:

- Parameters with high $\mu^*_{d_p}$ and σ_{d_p} (especially P_{12}) should be the focus of further investigation and potentially more precise experimental measurement;
- The widespread high σ_{d_p} values suggest that simplifying assumptions about parameter independence should be avoided in future analyses of this model.

This local sensitivity analysis provides a foundation for understanding the behaviour of the G-PIPK model near its optimal parameter set, but the complex interactions suggested by the results indicate that complementary analysis such as GSA and PLA will be crucial for a comprehensive understanding of the model's behaviour across its entire parameter space.

4.5.5.2 Global sensitivity analysis

The GSA was performed similarly to other models to quantify the influence of each P on model outputs. Sensitivity indices were calculated using the Sobol and Saltelli methods [417, 418], adapted from Halnes *et al.* [419], as shown in section 3.6.2.

Figures 4.29 and 4.30 illustrate the first-order (S_{Fi}) and total-order (S_{Ti}) sensitivity indices for each model parameter across the experimental outputs. These indices assess both direct effects and parameter interactions. The S_{Ti} values are generally higher than the S_{Fi} values, indicating significant parameter interactions within the model. The roles of specific parameters, detailed in Table D.1, are evident across different stages of the signalling pathway.

In $E_{0–11}$, sensitivities are influenced only by $P_{1–19}$, reflecting the model design where later sections (the PI cycle, M-current, and PH probe interactions) do not affect earlier reactions. The key point of separation in this model is the catalysis of PIP_2 into IP_3 and DAG by G_αGTP-PLC, which occurs in a single reaction without affecting upstream components. Since G_αGTP-PLC is not consumed in this process, its levels remain constant, preventing the latter part of the model from influencing the former.

Some clear relations between what reactions some parameters are involved in and their effect on outputs related to those reactions can clearly be observed:

- Influence of ligand types on parameters: $E_{0,\,3,\,5,\,7,\,9,\,11,\,13,\,15,\,17,\,19,\,21}$ (endogenous ligand reactions) are primarily influenced by $P_{1,2}$, which correspond to reactions involving endogenous ligands. In contrast, $E_{1,\,2,\,4,\,6,\,8,\,10,\,12,\,14,\,16,\,18,\,20}$ (exogenous ligand interactions) are significantly affected by $P_{15,16}$, related to exogenous ligand binding dynamics;

- Receptor-G-protein complex dynamics: M1R_Gbc_T is mainly affected by $P_{3–6}$. These parameters are associated with the binding and unbinding reactions of $G_{\beta\gamma}$ or G-protein to the receptor, highlighting their crucial role in receptor-G-protein complex formation;

- G-protein trimer formation: Gabc_T is predominantly influenced by $P_{7–10}$. These parameters govern the association and dissociation of G_α and $G_{\beta\gamma}$ subunits to form or disassemble G-protein trimers, which are essential steps in signal transduction;

- Ga-PLC interaction: Ga_PLC_T is affected by $P_{11–14}$. These parameters correspond to the binding and unbinding reactions between PLC and G_αGTP or G-proteins, emphasizing their role in modulating PLC activation.

- Downstream effects on PIP_2 and M-current: PIP2_PH_T and M_current are influenced by nearly all parameters. This widespread sensitivity reflects the integration of upstream signalling events and the cumulative effect of multiple reactions on these downstream outputs;

- Specific influence of PH probe reactions: $P_{28–32}$, which are related to the binding and unbinding of PH probes from IP_3 or PIP_2, show significant influence only in experi-

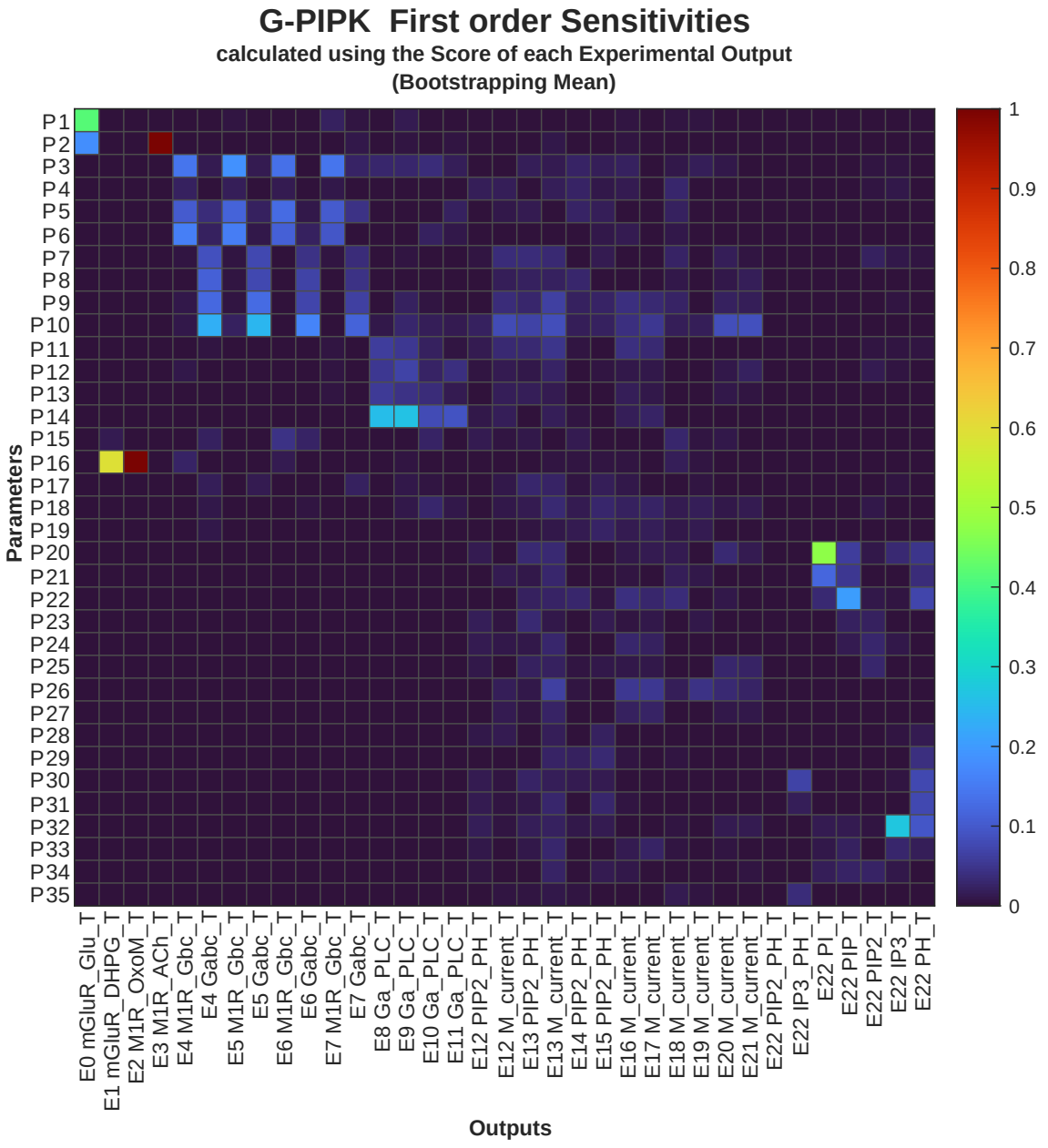


Figure 4.29: G-PIPK GSA S_{Fi} for each Eo. Sensitivity indices are computed for all model parameters ($P_{1–35}$) relative to the s_{Eo} of outputs from $E_{0–22}$ as shown in section 3.6.2.1. A sample size of 10000 θ values was used, sampled as described in section 3.6.2.2. Higher sensitivity values indicate a stronger influence of a parameter on a given experimental output.

ments involving PH probes. Notably, in E_{22}, these parameters have a pronounced effect on PIP2_PH_T, IP_3_PH_T, and PH_T, more so than on other variables.

The hierarchical structure within the signalling pathway is evident, where downstream elements are more influenced by upstream P than vice versa. For example, Gabc_T is more impacted by receptor-G-protein interactions ($P_{3–6}$) than M1R_Gbc_T is by PLC-G-protein interactions ($P_{11–14}$), reinforcing the directional nature of the cascade.

Figure 4.31 presents the first-order (S_{Fi}) and total-order (S_{Ti}) sensitivities for all model

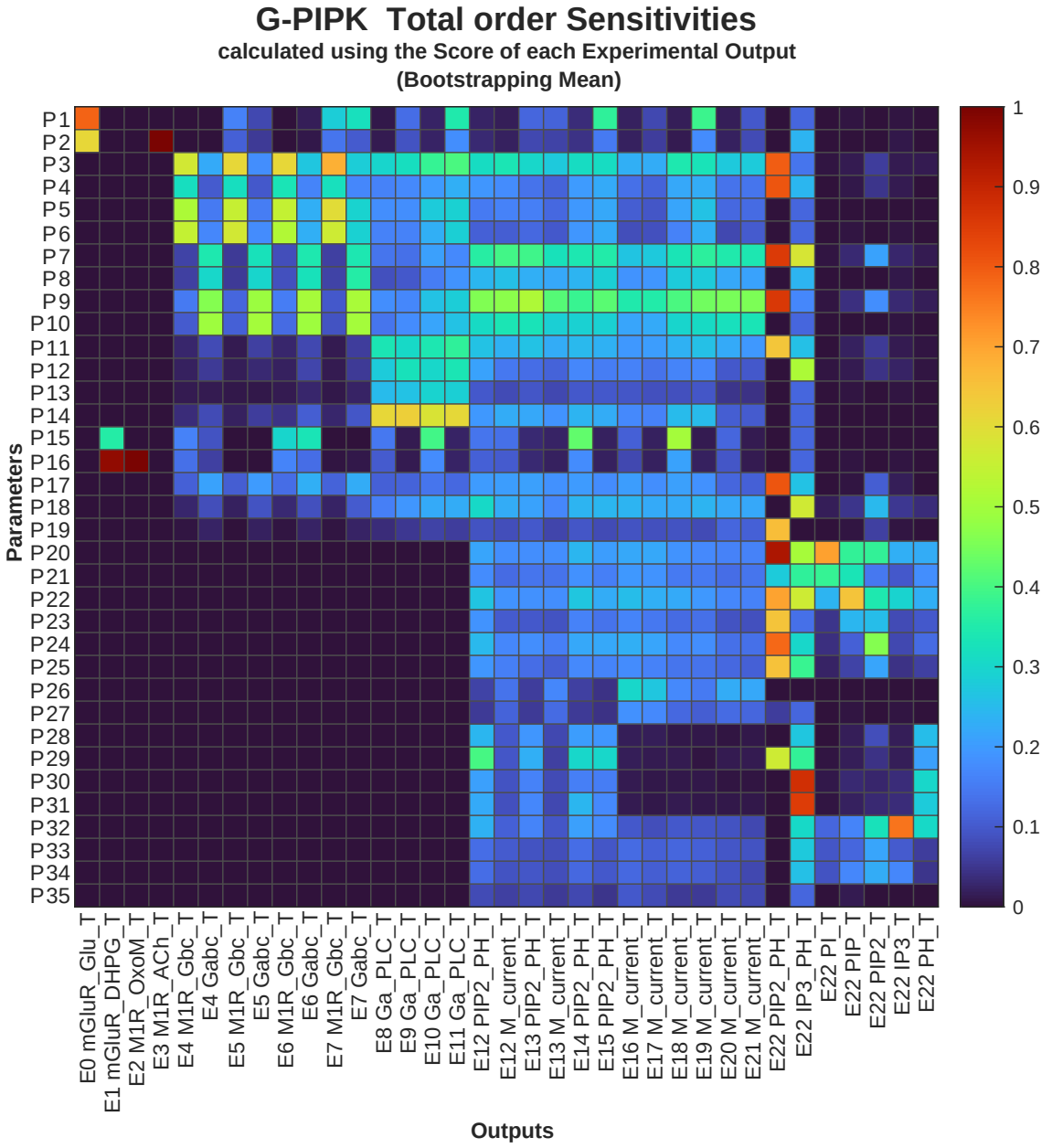


Figure 4.30: G-PIPK GSA S_{Ti} for each Eo. Sensitivity indices are computed for all parameters ($P_{1–35}$) relative to the s_{Eo} of outputs from $E_{0–22}$, as shown in section 3.6.2.1. A sample size of 10000 θ values was used. Higher values indicate greater parameter sensitivity for a given experimental output.

parameters ($P_{1–35}$) concerning the overall model score. Unlike the previous figures, which focused on sensitivities relative to individual experimental outputs, this figure provides insights into the parameters impact on the model as a whole.

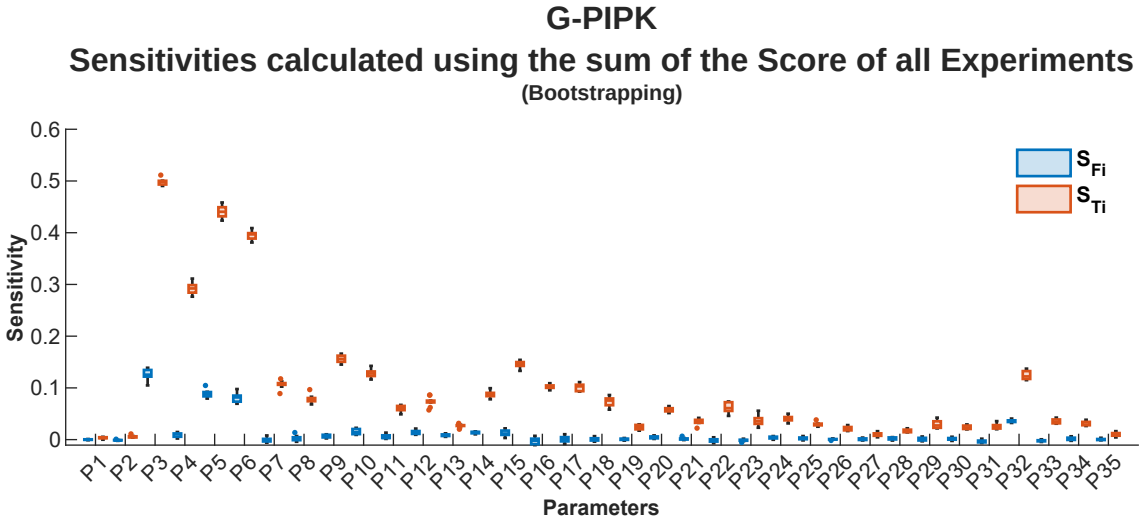


Figure 4.31: G-PIPK GSA S_{Fi} and S_{Ti} for all Es. S_{Fi} and S_{Ti} are calculated for all model parameters ($P_{1–35}$) given the model s_t, as shown in section 3.6.2.1. A sample size of 50000 θ sampled as in section 3.6.2.2 was used. Bootstrapping was performed as shown in section 3.7. The box plots have a darker line for the median of the sampled sensitivities, the top and bottom of the box correspond to the 75% and 25% percentile, respectively, and the top and bottom of the whisker correspond to the non outlier maximum and minimum of the sample, respectively [449]. Higher values indicate greater sensitivity of a specific parameter given a certain experimental output.

As shown, S_{Ti} values are consistently higher than S_{Fi}, suggesting a substantial degree of interaction among model parameters. Parameters P_1 and P_2, related to endogenous ligand binding, have minimal impact on the overall model score due to the limited experiments capturing this interaction with narrow uncertainties (E_0). However, these parameters still significantly affect experiments involving endogenous ligands, as seen in figures 4.29 and 4.30.

Overall, upstream parameters (lower numbered, early in the signalling cascade) generally exhibit a stronger influence on the model score, reinforcing the hierarchical nature of the model structure.

The GSA results provide valuable insights into the influence of model parameters, with several key parameters emerging as significant contributors to the model's output variance. This information can guide further refinement of the model, particularly in terms of parameter tuning and experimental focus.

Detailed results can be seen in figures D.14 to D.16. In figure D.14, notable parameters like P_5 and P_6 show bounded confidence intervals, reflecting their identifiability within the

model's context. In figure D.15, parameters like P_{16}, P_{17}, and P_{21} display identifiable regions with intersections at the confidence threshold, while others, such as P_{14}, suggest broader ranges with less constrained likelihoods. In figure D.16, some parameters, such as P_{26} and P_{29}, showing bounded confidence regions, while others suggest less sensitivity or broader ranges within the parameter space. These results help identify parameter regions that best fit the model, guiding further parameter refinement.

4.5.6 Profile likelihood analysis

The profile likelihood analysis (PLA) was conducted for the G-PIPK model to investigate parameter identifiability and interactions within the model's complex parameter space. Although three promising parameter sets (θs) were identified during the parameter estimation process (as detailed in section 4.5.4), only the first parameter set was utilized for the PLA. This decision was made due to the substantial computational demands of the analysis; performing the PLA for the G-PIPK model requires over one week of continuous computation on a high-performance processor (Intel i9-10980XE). Additional analyses using the other parameter sets were not expected to yield significantly different insights and were thus deemed impractical within the scope of this thesis.

By systematically varying each parameter while maximizing the likelihood over the remaining parameters, the PLA isolates the influence of individual parameters on the overall model fit. The confidence intervals, derived based on the 95% critical threshold of the χ^2 distribution (as described in section 3.7), indicate the range of parameter values consistent with the experimental data.

The PLA results revealed that the first parameter set is not the global minimum. For several parameters—specifically $P_{4–8,\ 10,\ 11,\ 21–24,\ 26,\ 28–32,\ 35}$—alternative parameter values were identified that fit the model as well as, or better than, the initial estimates. This suggests the presence of multiple local minima in the parameter space, complicating the identification of a global optimum.

The PLA results also revealed that some parameters exert a considerably greater influence on the total score than others. Most parameters have scores across their entire range that remain below 10^3, implying that while they significantly affect the model, they do not cause the model to deviate drastically from acceptable performance levels.

A summary of the PLA results is presented in figure 4.32, providing an overview of parameter behaviors during the analysis. This figure illustrates that some parameters lead to significantly worse scores when perturbed, while others have less impact on model per-

formance. Detailed results are provided in figure D.14 to figure D.16, which help identify parameter regions that best fit the model and guide further parameter refinement.

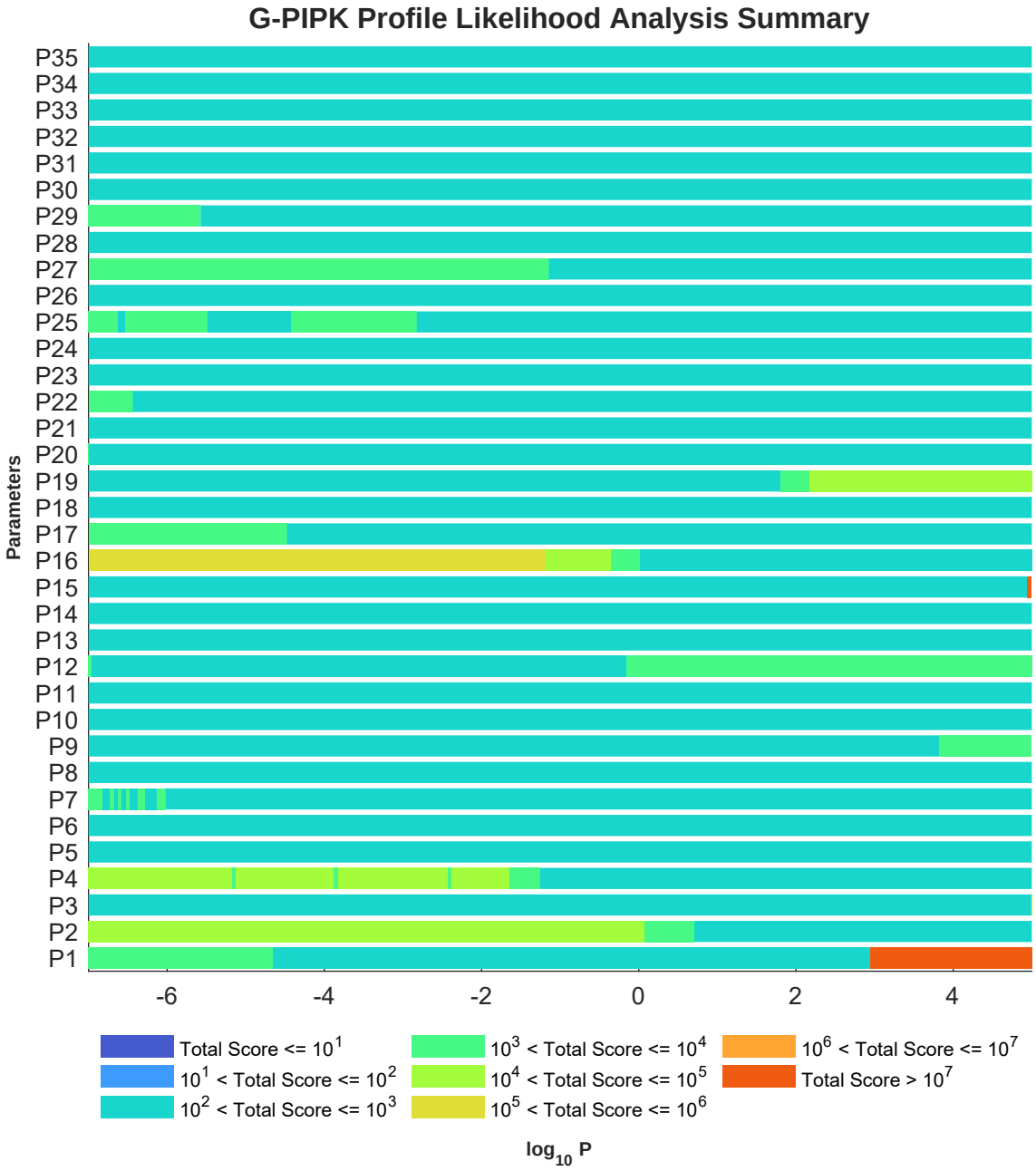


Figure 4.32: Summary of the PLA for the G-PIPK model. The figure illustrates the profile likelihoods for multiple parameters, each plotted as a function of the log-transformed parameter values (*log10 P*). The colors represent different likelihood scores, with colors closer to blue indicating lower scores and better model fits. This figure serves as an overview of the behaviour of parameters during the PLA, showing that while some parameters lead to significantly worse scores when perturbed, others have less impact on the model performance. It is important to note that this image does not depict confidence intervals or identifiable regions. Instead, it summarizes the detailed results from figures D.14, D.15, and D.16, illustrating the varying degrees of parameter influence on the model.

In figure D.14, notable parameters like P_5 and P_6 show bounded confidence intervals,

reflecting their identifiability within the model's context. In figure D.15, parameters like P_{16}, P_{17}, and P_{21} display identifiable regions where their profile likelihoods intersect the confidence threshold, while others, such as P_{14}, suggest broader ranges with less constrained likelihoods. In figure D.16, some parameters, such as P_{26} and P_{29}, show bounded confidence regions, while others indicate less sensitivity or broader ranges within the parameter space. These results aid in identifying parameter regions that best fit the model, guiding further parameter refinement.

Importantly, all parameters have defined bounds in the PLA, indicating that they are all identifiable within the scope of this analysis. However, since the minimum identified is not the true global optimum, the confidence intervals are valid only relative to the current local minimum.

The PLA also highlights alternative parameter sets that fit the model as well as, or better than, the initial parameter estimates. This suggests the presence of local minima in the parameter space, complicating the search for the global optimum. Despite these challenges, the most likely parameter ranges are successfully delineated by the PLA, providing a robust framework for parameter refinement and model improvement.

This detailed analysis underscores the importance of parameter identifiability in complex models, where interactions between parameters can lead to multiple regions of feasible solutions. The G-PIPK model's PLA results form a foundation for further refinement and application in more intricate scenarios later in this thesis.

4.5.7 Volume considerations

The data gathered for this model was recorded using whole cells [116, 440]. It is important to acknowledge that the difference in dimensions between entire cells and neuronal spines might significantly affect reaction dynamics. Here, it is assumed that the radius of an average cell is around 10 μm and that of an average spine is around 0.2 μm and see how this difference would affect the species concentrations. As illustrated in table 4.8, the dendritic spines would have, on average, a radius that is 50 times smaller than an average cell. This would also translate into a 2.500-fold difference in surface area and an enormous 125.000-fold difference in volume. These differences would result in spines having a surface-to-volume ratio that is 50 times higher than that of whole cells.

The surface area and volume were calculated assuming spheric cell and spine and using the formulas for the surface area (Sa) and volume (V) of the sphere: Sa = $4\pi r^2$ and $V = \frac{4}{3}\pi r^3$, respectively. The surface-to-volume ratio was then determined by dividing the surface

area by the volume (Sa/V). These calculations highlight the significantly higher surface area to volume ratio in spines, which has important implications for reaction kinetics and the local concentrations of membrane-bound molecules.

Measurement	Cell	Spine	Unit
radius (r)	10	0.2	μm
Sa	1257	0.5027	μm^2
V	4189	0.0335	μm^3
Sa to V Ratio	0.3	15	μm^{-1}

Table 4.8: Cell and spine typical dimensions. Cell and spine radii are based on typical values from literature [insert appropriate citation]. Sa, V, and Sa to V ratio calculated as described in the text. Note the significantly higher surface area to volume ratio in spines, which has important implications for reaction kinetics and local concentrations of membrane-bound molecules.

The models in this dissertation represent a homogeneous single compartment where all reactions occur and all species are uniformly distributed. To accurately model this system, the surface density of cellular components must be converted to volume concentrations. Due to their higher surface-to-volume ratio, spines exhibit higher concentrations of membrane-bound molecules after this conversion. Consequently, species concentrations differ markedly between average cells and neuronal spines, as shown in table 4.9.

To determine the concentration of molecules within the volume, we begin by converting the surface density to the total number of molecules present. This conversion is essential for transitioning from a surface-based measurement to a volume-based concentration.

$$Sd(\mu m^{-2}) = \frac{N(number\ of\ molecules)}{Sa(\mu m^2)} \tag{4.10}$$

$$N(number\ of\ molecules) = Sd(\mu m^{-2}) \times Sa(\mu m^2) \tag{4.11}$$

The total number of molecules is converted to molar concentration using Avogadro's constant ($N_A = 6.02214 \times 10^{23}$ mol^{-1}). This step translates the absolute number of molecules into a standardized concentration unit.

$$n(mol) = \frac{N(number\ of\ molecules)}{N_A(mol^{-1})} \tag{4.12}$$

The molar concentration is calculated by dividing the number of moles by the volume of the compartment.

$$C(mol/dm^3) = \frac{n(mol)}{V(dm^3)}$$

By making some substitution using equations (4.11) and (4.12) we can get to a formula that relates surface density with an internal volume concentration, providing a clear link between the two:

$$C(mol/dm^3) = \frac{N(number\ of\ molecules)}{V(dm^3) \times N_A(mol^{-1})} = \frac{Sd(\mu m^{-2}) \times Sa(\mu m^2)}{V(dm^3) \times N_A(mol^{-1})}$$

To ensure consistency in units, several unit transformations are necessary. The volume in cubic micrometers is converted to cubic decimetres, and the concentration is adjusted from nanomoles per cubic decimetre to moles per cubic decimetre.

$$V(dm^3) = \frac{V(\mu m^3)}{10^{15}}$$
$$C(mol/dm^3) = \frac{C(nmol/dm^3)}{10^9}$$

Applying these unit transformations, the concentration formula can be refined to account for the changes in units:

$$\frac{C(nmol/dm^3)}{10^9} = \frac{Sd(\mu m^{-2}) \times Sa(\mu m^2)}{\frac{V(\mu m^3)}{10^{15}} \times N_A(mol^{-1})}$$
$$C(nmol/dm^3) = \frac{Sd(\mu m^{-2}) \times Sa(\mu m^2) \times 10^9 \times 10^{15}}{V(\mu m^3) \times N_A(mol^{-1})} = \frac{Sd(\mu m^{-2}) \times Sa(\mu m^2) \times 10^{24}}{V(\mu m^3) \times N_A(mol^{-1})}$$

Finally, the concentration formulas specific to cells and spines are presented, incorporating all previous calculations and constants.

$$C_{\text{Cell}} = \frac{Sd_{\text{Cell}} \times Sa_{\text{Cell}} \times 10^{24}}{V_{\text{Cell}} \times N_A} \tag{4.13}$$
$$C_{\text{Spine}} = \frac{Sd_{\text{Cell}} \times Sa_{\text{Spine}} \times 10^{24}}{V_{\text{Spine}} \times N_A} \tag{4.14}$$

In section 4.5.8, model predictions using the best θ are presented to demonstrate the impact of using spine-appropriate versus cell-appropriate concentrations of species.

These volume differences are crucial when interpreting experimental data from whole cells versus isolated synapses. Accounting for these differences may help explain discrepancies observed between some in vitro and in vivo studies.

Species and condition	$Sd_{Cell}\ (\mu m^{-2})$	$C_{Cell}\ (nM)$	$C_{Spine}\ (nM)$	Reference
R (endogenous)	1	0.5	25	[440]
R (M1R-YFP)	3,000	1,494	74,724	[440]
R (M1R, FRET)	3,000	1,494	74,724	[440]
R (M1R, current)	500	249	12,454	[440]
G-protein (endogenous)	40	19.9	996	[440]
G-protein (overexpressed)	3,000	1,494	74,724	[440]
PLC(endogenous)	3	1.5	75	[440]
PLC (overexpressed)	3,000	1,494	74,724	[440]
PLC (total endogenous)	10	5.0	249	[440]
PI	140,000	69,743	3,487,132	[452]
PIP	5,000	2,491	124,540	[452]
PIP_2	4,000	1,993	99,632	[452]
PH domains (membrane)	3,000	1,494	74,724	[452]
Kv7 channels/KCNQ2/3	4	2.0	100	[452]

Table 4.9: Comparison of species concentrations between average cells and neuronal spines. Sd_{Cell}: surface density in cells; C_{Cell}: concentration in cells; C_{Spine}: concentration in spines. Concentrations calculated as shown in equations (4.13) and (4.14).

4.5.8 Model extrapolations for cell and spines

In this section, the G-PIPK model is extended to explore signalling dynamics in cellular and spine environments. This extension is built upon the parameter estimation results from section 4.5.4 and the volume considerations discussed in section 4.5.7. These extrapolations to distinct compartments aim to investigate potential differences in GPCR signalling and to demonstrate the software capabilities in making extrapolations.

The three best-performing θ obtained from the optimization process were utilized in conjunction with the initial conditions defined in table 4.9 to simulate GPCR signalling under various conditions. It should be noted that due to the complexity of the model and limitations in the used experimental data, some variability in predictions across different θ is observed. Nevertheless, valuable insights into potential differences between cellular and spine signalling dynamics are provided by these simulations.

To demonstrate the model's extrapolation capabilities, six experimental conditions not present in the training data were simulated: basal levels of ACh (100 nM), an ACh peak (10 μM) and an ACh dip (0 M), each of these with or without a glutamate peak. For each

condition, several key variables were analysed to provide a comprehensive view of the signalling cascade: ligand binding to GPCR with MGluR_Glu_T and M1R_ACh_T; GPCR-$G_{\beta\gamma}$ interaction with M1R_Gbc_T and the total amount of MGluR bound to $G_{\beta\gamma}$ (MGluR_Gbc_T); G-protein trimer separation with Gabc_T; G_{α}-PLC binding with Ga_PLC_T; membrane current with M_current; and lipid dynamics with PIP2_T and IP_3_T. It should be noted that M_current is calculated as $(KCNQ + PIP2_KCNQ) \times 0.5 \times 0.055 \times (PIP2_KCNQ^2)$, representing the PIP_2-dependent KCNQ2/3 membrane current.

The model's predictions for basal ACh levels in both cellular and spine environments are illustrated in figure 4.33, while results for the other conditions are presented in figures D.17 to D.21. These figures illustrate the model's predictions for ACh peaks, dips, and their combinations with glutamate peaks, providing a comprehensive view of the signalling dynamics under various stimuli conditions.

Despite high variability between different parameter sets, faster reaction kinetics and higher concentrations of signalling components in the spine environment compared to the cellular environment are consistently observed in the extrapolation simulations. The higher concentration of signalling components is expected based on the higher concentration of initial species due to higher surface-to-volume ratio as discussed in section 4.5.7, but the increase of speed of reactions can be considered an interesting novel result.

The main observations are accelerated GPCR activation in spine, faster G-protein cycling, larger and faster fluctuations in PIP_2 and IP_3 levels, and faster changes in M_current. This suggests a more dynamic regulation of membrane excitability in spine compartments. While preliminary, a foundation for understanding how cellular compartment size may influence GPCR signalling dynamics is provided by these findings, highlighting the importance of considering spatial factors in interpreting experimental data and developing mechanistic models of neuronal signalling.

The potential impact of compartment-specific dynamics on synaptic plasticity and neuronal information processing is underscored by the observed differences between cell and spine simulations. However, it is crucial that the limitations of these extrapolations are acknowledged, given the current constraints of the model. Future work should be focused on refining these predictions through additional experimental validation, particularly in spine-like compartments. The variability observed between different θ may be reduced by improved constraints on model parameters and incorporation of additional signalling components.

While limitations in the current extrapolations are recognized, the potential of the G-PIPK model to provide insights into compartment-specific signalling dynamics is demonstrated.

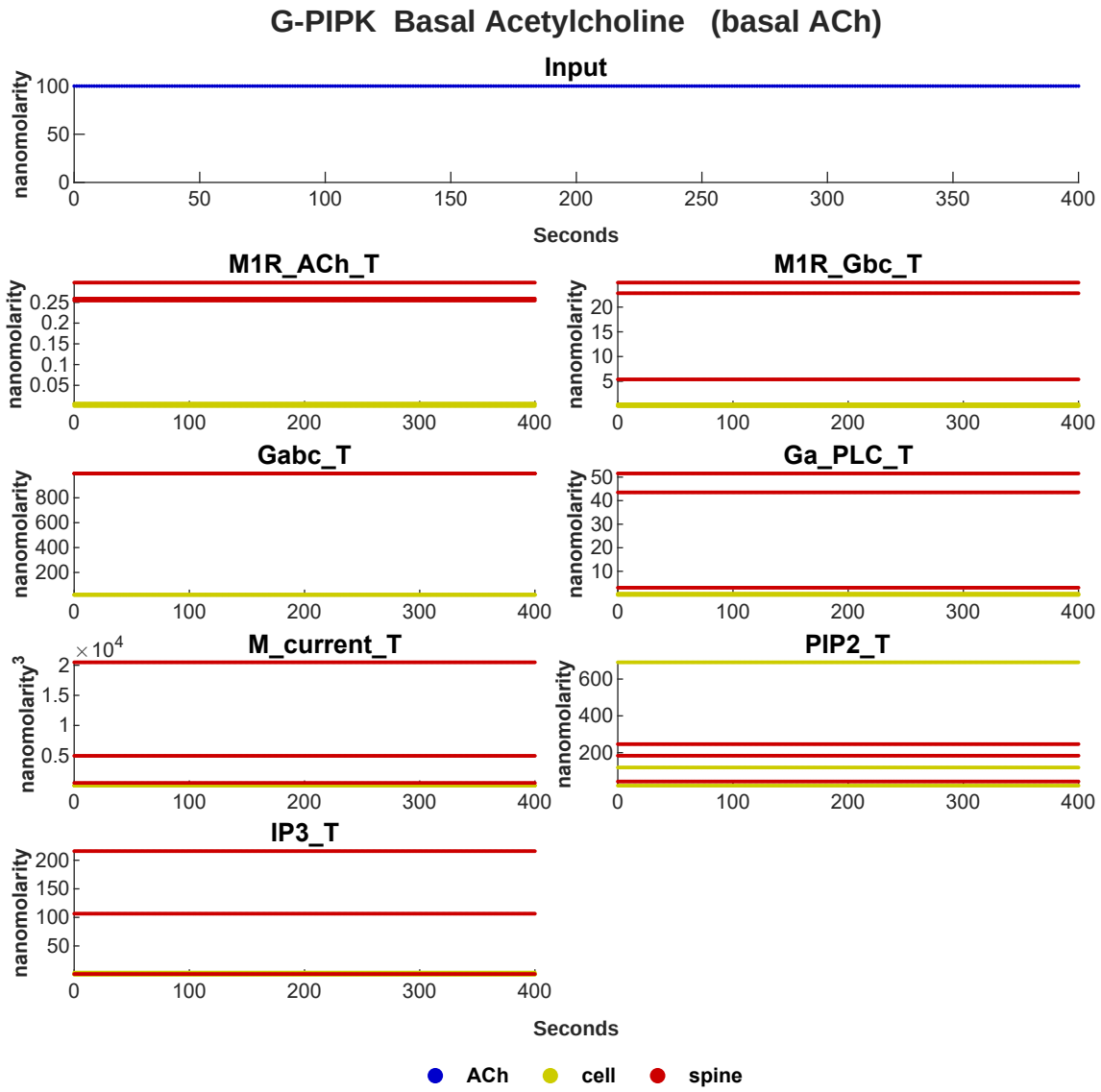


Figure 4.33: G-PIPK prediction with basal ACh. The top graph shows the input (100 nM ACh), while the bottom panels depict various outputs from the model: the total amount of M1R bound to ACh (M1R_ACh_T), the total amount of M1R bound to $G_{\beta\gamma}$ (M1R_Gbc_T), the total amount of G-protein trimer concentration (Gabc_T), the total amount of G_{α} bound to PLC (Ga_PLC_T), the PIP_2-dependent Kv7.2/3 membrane current (M_current), the total amount of PIP_2 (PIP2_T) and the total amount of IP_3 (IP_3_T). The simulation was performed using initial conditions for cell and spine, as described in section 4.5.7 and table 4.9, with three optimized θ values.

The groundwork for future studies exploring the functional significance of spine morphology in GPCR-mediated signalling and its implications for neuronal function is laid by these results.

4.5.9 Discussion

The G-PIPK model, with its 35 free parameters, functioned as a final test to the applicability of the developed software. Being the most complex system analysed in this dissertation,

it challenged and validated the effectiveness of the parameter estimation and analysis tools in handling intricate biochemical systems. This discussion focuses on the main challenges encountered while developing the model and applying the analysis software to it.

Previous models had small unconstrained parameter sizes: LiRMo – 7 parameters; LiREM – 12 parameters; TM-M – 6 parameters. G-PIPK with its 35 unconstrained parameters demanded advanced optimization strategies and significantly more computational resources for its analysis, bordering on what is feasible to tackle in a top-of-the-line workstation computer (Intel i9-10980XE was used throughout this study).

The parameter estimation analysis was able to handle this more complex model, successfully dealing with the increased amount of parameters and data and providing reasonable θ for data fitting.

Different optimization strategies where used to try and improve the odds of finding good θ. Genetic algorithm was used either in a large single run or in two ensemble runs. One of the ensemble runs used the default θ bounds and the other used bounds obtained from performing PLA on the θ of the single large run. The optimization process yielded three parameter sets, θ_1, θ_2 and θ_3, with total scores of 162.20, 134.81 and 155.05, respectively. The ensemble approaches were more successful in finding better θs but neither of them could be considered a global optimum. This highlights that despite being capable of optimizing larger models, the software is limited by the computational difficulty of finding global optimums of large parameter spaces.

Some outputs where easier to fit than others and, while overall the parameter estimation process can be considered a success for this model, several points of improvement should be taken into account if further development of the model software is done in the future. For the model these are data limitations and simplifying model assumptions. The data used was sourced from only two research groups with the most recent data from 2010 [116, 440, 452, 453], and artificially generated datasets were used to fill gaps, introducing biases. Several model assumptions were made, such as, consistent ligand binding rates regardless of receptor complex composition, which may oversimplify biological complexities and decrease the capability of the model to fit the available data. For the software these are its susceptibility to getting stuck in local minima in complex parameter spaces. Further work should be done to improve optimization routines maybe going beyond what is offered out-of-the-box in MATLAB.

The sensitivity analyses provided significant insights into the model's behaviour and the relative importance of various parameters. Both local and global sensitivity analyses high-

lighted the substantial impact of upstream parameters on the model's outputs, reflecting the hierarchical organization of the GPCR signalling pathway. Parameters such as P_{12}, which is involved in the binding dynamics of $G_{\alpha}GTP$ to PLC, were identified as having considerable influence on the model's output near the optimal parameter set, underscoring their critical role. The observed standard deviations in the elementary effects during local sensitivity analysis, along with the high total-order sensitivities across most parameters, pointed to significant non-linearities and parameter interactions, revealing the inherent complexity of the model. These non-linear effects and high sensitivities suggest that the influence of individual parameters is not simply additive but arises from intricate interactions that require advanced analytical approaches to be fully understood.

These findings offer valuable guidance for future experimental efforts, indicating specific areas where targeted and precise measurements could enhance parameter estimation and thereby improve model reliability. By focusing on accurate experimental quantification for the most sensitive parameters while also enhancing data acquisition for the less sensitive ones, it is expected that the overall accuracy of the model can be significantly improved. Such a targeted approach is likely to reduce uncertainties and refine the model's predictive power, thereby enabling more reliable simulations of the GPCR signalling pathway.

The PLA revealed that several parameters were not well-identified, with some demonstrating extensive confidence intervals. This lack of identifiability implies that the available data are insufficient to adequately constrain these parameters, resulting in considerable uncertainty in parameter estimation. Additionally, the PLA revealed the existence of lower minima compared to those obtained by the initial parameter estimation, further indicating that the parameter estimation had only found a local minimum within the parameter space.

The findings from the PLA underscore the necessity for targeted experimental data to enhance the identifiability of poorly constrained parameters to improve model accuracy. Experimental designs aimed at reducing uncertainties in parameters with broad confidence intervals could substantially contribute to achieving better-constrained estimates. Conversely, parameters with narrow confidence intervals signify regions where the model is sufficiently constrained, making additional data acquisition less critical. These observations emphasize the importance of refining both the model structure and data quality to enhance the model.

The extrapolation of the model to cellular and dendritic spine environments revealed significant differences in signalling dynamics, primarily driven by variations in surface-to-volume ratios. Dendritic spines exhibited enhanced reaction kinetics and elevated concentrations of signalling molecules, consistent with their relatively higher surface-to-volume ratios. These

insights could have profound implications for elucidating the intricate biochemical processes within neuronal circuits. Specifically, the compartmentalization inherent to neuronal architecture likely exerts a substantial influence on signalling dynamics, thereby modulating synaptic plasticity and, by extension, information processing. These results emphasize the critical need to integrate spatial dimensions and compartment-specific properties into biochemical models to achieve a more accurate representation of cellular processes. Oversimplifications that treat the system as a homogeneous single compartment risk neglecting key spatial heterogeneities and diffusion constraints that are essential to accurately characterize intracellular signalling. Furthermore, notable variability across different parameter sets during extrapolation highlighted potential challenges regarding parameter robustness. Such variability underscores the necessity for refining parameter estimation methods to enhance the reliability of model predictions. Looking ahead, incorporating spatial heterogeneity explicitly and employing more sophisticated parameter estimation strategies will be essential for capturing the complexity of intracellular dynamics within intricate cellular compartments, and represent clear avenues for improvement of the developed software.

The identified limitations of the model underscore several critical areas for future research, particularly concerning data insufficiencies and oversimplified model assumptions. Incorporating recent, diverse experimental data, especially from multiple independent laboratories, would considerably enhance the model's calibration, validation, and overall robustness. The current model lacks several key aspects, such as voltage dynamics, receptor conformational states, and the influence of membrane regions. Addressing these deficiencies by incorporating spatial modelling approaches, including partial differential equations to represent diffusion and localization, could substantially enhance model fidelity and mitigate the inherent limitations of the well-mixed assumption. Additionally, parameter estimation challenges must be addressed by employing more sophisticated techniques, to effectively navigate complex parameter landscapes and reduce convergence to local minima.

Future research should prioritize targeted experimental and computational advancements to improve model reliability. Detailed kinetic data acquisition through advanced methods like single-molecule fluorescence resonance energy transfer (FRET) or live-cell imaging is essential for poorly constrained parameters, which would reduce uncertainties and increase accuracy. Further development of compartmental modelling is also necessary to better capture the impact of cellular morphology on signalling dynamics. Current extrapolations to complex environments, such as cellular and dendritic spines, have demonstrated substantial differences driven by variations in surface-to-volume ratios. Enhanced spatial modelling

would enable a more precise representation of these differences, accurately reflecting the intricate biochemical processes occurring within distinct cellular environments.

Another important avenue for future research is the exploration of pathway cross-talk. Examining interactions between glutamatergic and cholinergic pathways may reveal synergistic or antagonistic effects, thereby providing a more nuanced understanding of neuronal signalling. Clarifying these integrative and competitive relationships is pivotal for elucidating mechanisms underlying synaptic plasticity and neuronal adaptation. By systematically addressing data limitations, refining spatial modelling, and investigating pathway interactions, the model's accuracy and predictive capabilities can be significantly enhanced, facilitating a deeper understanding of the complexities inherent in intracellular signalling networks and neuronal dynamics.

The successful application of the developed software tools to the G-PIPK model highlights their efficacy in managing complex, high-dimensional biochemical systems, thereby affirming their value in advanced systems biology research. The findings from this work not only provide a basis for refining the current model but also pave the way for applying these computational methodologies to other complex biological systems. By incorporating more comprehensive experimental data, optimizing strategies, and refining the software to support expanded analysis capabilities, including spatial data integration, future research can significantly enhance the predictive accuracy of such models. This work contributes to a better understanding of neuronal signalling mechanisms and emphasizes the role of computational tools in biological research, facilitating more detailed analysis of complex cellular processes.

5 Conclusion

This thesis aimed to develop and validate a comprehensive analysis framework designed to address the intricacies inherent to ODE biochemical pathway models, specifically focusing on the basal ganglia. The primary objective was to establish an intuitive software platform capable of managing diverse models, integrating multiple analytical methodologies, and deriving meaningful insights into cellular signalling dynamics. Unlike existing frameworks [4], this platform uniquely integrates FAIR principles [374], ensuring that all models, analyses, and data are shareable and reusable, and that data storage and processing are structured to facilitate straightforward re-utilization. By combining a broad range of integrated analytical tools with minimal prerequisites for technical expertise, the platform enables the exploration of increasingly diverse and complex biological systems, thereby promoting efficient knowledge accumulation. This conclusion encapsulates the principal achievements, contributions, limitations, and broader implications of this work.

The platform's robustness and adaptability were demonstrated through progressive validation using increasingly sophisticated test cases. Key technical achievements include a streamlined mechanism for model import and processing in MATLAB, which requires only a rudimentary familiarity with the software, thereby making it accessible to users without coding experience. The implementation of several optimization algorithms enhanced the platform's analytical capabilities and enabled the application of additional analysis tools: sensitivity analysis, which facilitated the identification of model regions that were more or less affected by parameter changes, and PLA that enabled a deeper understanding of parameter identifiability, highlighting areas of the models that could benefit from targeted future experimentation.

In this thesis, the software was applied to four models of increasing complexity: LiRMo, LiREM, TM-M, and G-PIPK, showcasing its flexibility in addressing different levels of model intricacy. The LiRMo and LiREM models were employed to validate basic software functionalities. They successfully demonstrated that the software could properly import models into MATLAB, perform parameter estimation accurately, and generate meaningful results from sensitivity analysis and PLA tools. Their simplicity allowed for straightforward evaluation of software performance before progressing to more complex models.

The TM-M model, derived from a previously developed framework [2], served as an intermediary test for software evaluation. While it was the largest model in terms of total parameter size, it was not the most challenging regarding the number of parameters evaluated.

This demonstrated the software's capacity to focus on and enhance just a subset of a larger model, highlighting that it can effectively handle extensive models as long as the parameter set under scrutiny remains manageable. Additionally, it demonstrated the software's utility in processing data derived from other models (unlike LiRMo and LiREM, which used artificially generated data) and in providing new, biologically relevant insights into pre-existing models, thereby proving its broader applicability to the research community.

The G-PIPK model, developed to investigate a complex GPCR signalling cascade, represented the most parameter-rich model handled by the software. It demonstrated the platform's ability to facilitate the development of new models effectively. The software successfully managed real biological data, with further analysis using sensitivity analysis and PLA pinpointing areas where additional data were needed and where existing data were sufficient. This model also demonstrated the software's predictive capabilities, evidenced by the differences in model behaviour in response to changes in volume and surface-to-volume ratio-constrained molecules.

Despite these successes, this research has identified several areas requiring further advancement, particularly regarding computational efficiency and experimental data requirements. A primary limitation is the computational constraint encountered when managing large parameter spaces. To address this, future work could leverage high-performance cloud computing platforms or employ GPU-based optimization methods. Another key limitation, specific to the G-PIPK model, is the need for additional experimental data to further constrain the model and improve its predictive accuracy. Collaborations with experimental laboratories could play a critical role in bridging this gap. Enhancements to the software to accommodate spatial components also offer a promising avenue for future refinement.

The developed framework holds significant potential for enhancing precision in drug target identification and for improving the understanding of neurological disease mechanisms through advanced model simulations. These applications underscore the broader impact of this work in both the research and medical fields. This research makes meaningful contributions to the computational neuroscience tool-kit by providing novel methods and tools for model analysis and development. The refined model development workflow benefits future research initiatives, while the improved understanding of cellular signalling dynamics lays a foundation for subsequent developments in biological modelling and analysis. These contributions have the potential to inform experimental studies and advance our comprehension of complex biological systems, positioning this work as an important step forward in the field.

In conclusion, this research presents a versatile and FAIR-compliant analysis frame-

work capable of addressing complex computational neuroscience models. It provides a robust foundation for advancing computational tools and methodologies in neuroscience, with promising implications for future research and practical applications. This framework has the potential to accelerate discoveries and enhance the modelling of complex biological systems, ultimately contributing to the advancement of both basic and translational research in the field.

Bibliography

[1] João PG Santos et al., 2022. “A modular workflow for model building, analysis, and parameter estimation in systems biology and neuroscience”. *Neuroinformatics* 20(1), pp. 241–259.

[2] Anu G Nair, Upinder S Bhalla, and Jeanette Hellgren Kotaleski, 2016. “Role of DARPP-32 and ARPP-21 in the emergence of temporal constraints on striatal calcium and dopamine integration”. *PLoS computational biology* 12(9), e1005080.

[3] João PG Santos et al., 2022. “Model Nair 2016”. *GitHub repository*, URL: `https://github.com/jpgsantos/Model_Nair_2016`, Release tag: 1.0.

[4] Nisha A Viswan et al., 2018. “FindSim: a framework for integrating neuronal data and signaling models”. *Frontiers in neuroinformatics* 12, p. 38.

[5] João PG Santos et al., 2022. “Model Viswan 2018”. *GitHub repository*, URL: `https://github.com/jpgsantos/Model_Viswan_2018`, Release tag: 1.0.

[6] Kazuhiro A Fujita et al., 2010. “Decoupling of receptor and downstream signals in the Akt pathway by its low-pass filter characteristics”. *Science signaling* 3(132), ra56–ra56.

[7] João PG Santos et al., 2022. “Model Fujita 2010”. *GitHub repository*, URL: `https://github.com/jpgsantos/Model_Fujita_2010`, Release tag: 1.0.

[8] João PG Santos, 2024. “Subcellular workflow”. *GitHub repository*, URL: `https://github.com/jpgsantos/Subcellular_workflow`, Release tag: 2.0.

[9] João PG Santos et al., 2022. “Subcellular workflow”. *GitHub repository*, URL: `https://github.com/jpgsantos/Subcellular_workflow`, Release tag: 1.0.

[10] João PG Santos and Jeanette Hellgren Kotaleski, 2024. “Model GPIPK”. *GitHub repository*, URL: `https://github.com/jpgsantos/GPIPK`, Release tag: 1.0.

[11] João PG Santos and Jeanette Hellgren Kotaleski, 2024. “Timing Model Modified”. *GitHub repository*, URL: `https://github.com/jpgsantos/Timing_Model_Modified`, Release tag: 1.0.

[12] Danielle S Bassett and Olaf Sporns, 2017. “Network neuroscience”. *Nature neuroscience* 20(3), pp. 353–364.

[13] Terrence J Sejnowski, Patricia S Churchland, and J Anthony Movshon, 2014. “Putting big data to good use in neuroscience”. *Nature neuroscience* 17(11), pp. 1440–1441.

[14] Danielle S Bassett, Perry Zurn, and Joshua I Gold, 2018. “On the nature and use of models in network neuroscience”. *Nature Reviews Neuroscience* 19(9), pp. 566–578.

[15] Jeanette Hellgren Kotaleski and Kim T Blackwell, 2010. “Modelling the molecular mechanisms of synaptic plasticity using systems biology approaches”. *Nature Reviews Neuroscience* 11(4), pp. 239–251.

[16] Narat J Eungdamrong and Ravi Iyengar, 2004. “Modeling cell signaling networks”. *Biology of the Cell* 96(5), pp. 355–362.

[17] Narat J Eungdamrong and Ravi Iyengar, 2004. “Computational approaches for modeling regulatory cellular networks”. *Trends in Cell Biology* 14(12), pp. 661–669.

[18] José L Laego, Natasha Luquin, and José A Obeso, 2012. “Functional neuroanatomy of the basal ganglia”. *Cold Spring Harbor perspectives in medicine* 2(12), a009621.

[19] Charles R Gerfen and D James Surmeier, 2011. “Modulation of striatal projection systems by dopamine”. *Annual review of neuroscience* 34, pp. 441–466.

[20] Anatol C Kreitzer and Robert C Malenka, 2008. “Striatal plasticity and basal ganglia circuit function”. *Neuron* 60(4), pp. 543–554.

[21] William C. Hall and Leonard White, 2018. “Modulation of Movement by the Basal Ganglia”. In: *Neuroscience*, 6th ed., chap. 18, pp. 407–425.

[22] Mitchell Glickstein, Jack G May III, and Barbara E Mercier, 1985. “Corticopontine projection in the macaque: the distribution of labelled cortical cells after large injections of horseradish peroxidase in the pontine nuclei”. *Journal of Comparative Neurology* 235(3), pp. 343–359.

[23] Janet M Kemp and Thomas Philip Stroud Powell, 1971. “The connexions of the striatum and globus pallidus: synthesis and speculation”. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences* 262(845), pp. 441–457.

[24] Nicholas N Foster et al., 2021. “The mouse cortico–basal ganglia–thalamic network”. *Nature* 598(7879), pp. 188–194.

[25] Silvia Arber and Rui M Costa, 2018. “Connecting neuronal circuits for movement”. *Science* 360(6396), pp. 1403–1404.

[26] Garrett E Alexander, Mahlon R DeLong, and Peter L Strick, 1986. “Parallel organization of functionally segregated circuits linking basal ganglia and cortex”. *Annual review of neuroscience* 9(1), pp. 357–381.

[27] Frank A Middleton and Peter L Strick, 2000. “Basal ganglia output and cognition: evidence from anatomical, behavioral, and clinical studies”. *Brain and cognition* 42(2), pp. 183–200.

[28] Kenji Doya, 2000. “Complementary roles of basal ganglia and cerebellum in learning and motor control”. *Current opinion in neurobiology* 10(6), pp. 732–739.

[29] Andreas Klaus, Joaquim Alves da Silva, and Rui M Costa, 2019. “What, if, and when to move: Basal ganglia circuits and self-paced action initiation”. *Annual review of neuroscience* 42, pp. 459–483.

[30] Nicholas R Wall et al., 2013. “Differential innervation of direct-and indirect-pathway striatal projection neurons”. *Neuron* 79(2), pp. 347–360.

[31] André Parent and Lili-Naz Hazrati, 1995. “Functional anatomy of the basal ganglia. I. The cortico-basal ganglia-thalamo-cortical loop”. *Brain research reviews* 20(1), pp. 91–127.

[32] Garrett E Alexander and Michael D Crutcher, 1990. “Functional architecture of basal ganglia circuits: neural substrates of parallel processing”. *Trends in neurosciences* 13(7), pp. 266–271.

[33] Charles R Gerfen et al., 1990. “D1 and D2 dopamine receptor-regulated gene expression of striatonigral and striatopallidal neurons”. *Science* 250(4986), pp. 1429–1432.

[34] Y Smith et al., 1998. “Microcircuitry of the direct and indirect pathways of the basal ganglia.” *Neuroscience* 86(2), pp. 353–387.

[35] Mahlon R DeLong, 1990. “Primate models of movement disorders of basal ganglia origin”. *Trends in neurosciences* 13(7), pp. 281–285.

[36] Roger L Albin, Anne B Young, and John B Penney, 1989. “The functional anatomy of basal ganglia disorders”. *Trends in neurosciences* 12(10), pp. 366–375.

[37] Nicolas X Tritsch and Bernardo L Sabatini, 2012. “Dopaminergic modulation of synaptic transmission in cortex and striatum”. *Neuron* 76(1), pp. 33–50.

[38] D James Surmeier, Steven M Graves, and Weixing Shen, 2014. “Dopaminergic modulation of striatal networks in health and Parkinson’s disease”. *Current opinion in neurobiology* 29, pp. 109–117.

[39] Richard Gast et al., 2021. “On the role of arkypallidal and prototypical neurons for phase transitions in the external pallidum”. *Journal of neuroscience* 41(31), pp. 6673–6683.

[40] Rodrigo Muñoz-Castañeda et al., 2021. “Cellular anatomy of the mouse primary motor cortex”. *Nature* 598(7879), pp. 159–166.

[41] Alberto Cacciola et al., 2019. “Structural connectivity-based topography of the human globus pallidus: Implications for therapeutic targeting in movement disorders”. *Movement Disorders* 34(7), pp. 987–996.

[42] Michael C Chen et al., 2015. “Identification of a direct GABA ergic pallidocortical pathway in rodents”. *European Journal of Neuroscience* 41(6), pp. 748–759.

[43] Arpiar Saunders et al., 2015. “A direct GABAergic output from the basal ganglia to frontal cortex”. *Nature* 521(7550), pp. 85–89.

[44] Nicolas Mallet et al., 2016. “Arkypallidal cells send a stop signal to striatum”. *Neuron* 89(2), pp. 308–316.

[45] Atsushi Nambu, Hironobu Tokuno, and Masahiko Takada, 2002. “Functional significance of the cortico–subthalamo–pallidal ‘hyperdirect’pathway”. *Neuroscience research* 43(2), pp. 111–117.

[46] Alberto Cacciola et al., 2016. “A direct cortico-nigral pathway as revealed by constrained spherical deconvolution tractography in humans”. *Frontiers in Human Neuroscience* 10, p. 374.

[47] F-M Zhou and Christian R Lee, 2011. “Intrinsic and integrative properties of substantia nigra pars reticulata neurons”. *Neuroscience* 198, pp. 69–94.

[48] Lauren E McElvain et al., 2021. “Specific populations of basal ganglia output neurons target distinct brain stem areas while collateralizing throughout the diencephalon”. *Neuron* 109(10), pp. 1721–1738.

[49] James M Tepper et al., 2010. “Heterogeneity and diversity of striatal GABAergic interneurons”. *Frontiers in neuroanatomy* 4, p. 150.

[50] James M Tepper et al., 2018. “Heterogeneity and diversity of striatal GABAergic interneurons: update 2018”. *Frontiers in neuroanatomy* 12, p. 91.

[51] Aryn H Gittis et al., 2010. “Distinct roles of GABAergic interneurons in the regulation of striatal output pathways”. *Journal of Neuroscience* 30(6), pp. 2223–2234.

[52] Henrike Planert et al., 2010. “Dynamics of synaptic transmission between fast-spiking interneurons and striatal projection neurons of the direct and indirect pathways”. *Journal of Neuroscience* 30(9), pp. 3499–3507.

[53] Susanne N Szydlowski et al., 2013. “Target selectivity of feedforward inhibition by striatal fast-spiking interneurons”. *Journal of Neuroscience* 33(4), pp. 1678–1683.

[54] Johannes Hjorth, Kim T Blackwell, and Jeanette Hellgren Kotaleski, 2009. “Gap junctions between striatal fast-spiking interneurons regulate spiking activity and synchronization as a function of cortical activity”. *Journal of Neuroscience* 29(16), pp. 5276–5286.

[55] Maxime Assous and James M Tepper, 2019. “Excitatory extrinsic afferents to striatal interneurons and interactions with striatal microcircuitry”. *European Journal of Neuroscience* 49(5), pp. 593–603.

[56] Daniel F English et al., 2012. “GABAergic circuits mediate the reinforcement-related signals of striatal cholinergic interneurons”. *Nature neuroscience* 15(1), pp. 123–130.

[57] Maxime Assous et al., 2017. “Differential processing of thalamic information via distinct striatal interneuron circuits”. *Nature communications* 8(1), p. 15860.

[58] Thomas W Faust et al., 2015. “Novel fast adapting interneurons mediate cholinergic-induced fast GABAA inhibitory postsynaptic currents in striatal spiny neurons”. *European Journal of Neuroscience* 42(2), pp. 1764–1774.

[59] Veronique Bernard, Elisabeth Normand, and Bertrand Bloch, 1992. “Phenotypical characterization of the rat striatal neurons expressing muscarinic receptor genes”. *Journal of Neuroscience* 12(9), pp. 3591–3600.

[60] Alexandria E Melendez-Zaidi, Harini Lakshminarasimhah, and Dalton James Surmeier, 2019. “Cholinergic modulation of striatal nitric oxide-producing interneurons”. *European Journal of Neuroscience* 50(11), pp. 3713–3731.

[61] Tibor Koós and James M Tepper, 2002. “Dual cholinergic control of fast-spiking interneurons in the neostriatum”. *Journal of Neuroscience* 22(2), pp. 529–535.

[62] Sarah Threlfell et al., 2012. “Striatal dopamine release is triggered by synchronized activity in cholinergic interneurons”. *Neuron* 75(1), pp. 58–64.

[63] Maya Ketzef and Gilad Silberberg, 2021. “Differential synaptic input to external globus pallidus neuronal subpopulations in vivo”. *Neuron* 109(3), pp. 516–529.

[64] Ann M Graybiel, 2005. “The basal ganglia: learning new tricks and loving it”. *Current opinion in neurobiology* 15(6), pp. 638–644.

[65] Ann M Graybiel, 2008. “Habits, rituals, and the evaluative brain”. *Annu. Rev. Neurosci.* 31, pp. 359–387.

[66] Henry H Yin et al., 2009. “Dynamic reorganization of striatal circuits during the acquisition and consolidation of a skill”. *Nature neuroscience* 12(3), pp. 333–341.

[67] H Kita and ST Kitai, 1988. “Glutamate decarboxylase immunoreactive neurons in rat neostriatum: their morphological types and populations”. *Brain research* 447(2), pp. 346–352.

[68] Guohong Cui et al., 2013. “Concurrent activation of striatal direct and indirect pathways during action initiation”. *Nature* 494(7436), pp. 238–242.

[69] Alexxai V Kravitz, Lynne D Tye, and Anatol C Kreitzer, 2012. “Distinct roles for direct and indirect pathway striatal neurons in reinforcement”. *Nature neuroscience* 15(6), pp. 816–818.

[70] Wolfram Schultz, 2015. “Neuronal reward and decision signals: from theories to data”. *Physiological reviews* 95(3), pp. 853–951.

[71] Yoland Smith et al., 2014. “The thalamostriatal system in normal and diseased states”. *Frontiers in systems neuroscience* 8, p. 5.

[72] H Ehringer and OHRINGERH Hornykiewicz, 1960. “Distribution of noradrenaline and dopamine (3-hydroxytyramine) in the human brain and their behavior in diseases of the extrapyramidal system”. *Klinische Wochenschrift* 38, pp. 1236–1239.

[73] Carles Gaig and Eduardo Tolosa, 2009. “When does Parkinson’s disease begin?” *Movement Disorders* 24(S2), S656–S664.

[74] Kevin J Mastro et al., 2014. “Transgenic mouse lines subdivide external segment of the globus pallidus (GPe) neurons and reveal distinct GPe output pathways”. *Journal of Neuroscience* 34(6), pp. 2087–2099.

[75] Aryn H Gittis et al., 2014. “New roles for the external globus pallidus in basal ganglia circuits and behavior”. *Journal of Neuroscience* 34(46), pp. 15178–15183.

[76] GA Graveland and M DiFiglia, 1985. “The frequency and distribution of medium-sized neurons with indented nuclei in the primate and rodent neostriatum”. *Brain research* 327(1-2), pp. 307–311.

[77] Dennis A Burke, Horacio G Rotstein, and Veronica A Alvarez, 2017. “Striatal local circuitry: a new framework for lateral inhibition”. *Neuron* 96(2), pp. 267–284.

[78] Anatol C Kreitzer and Robert C Malenka, 2007. “Endocannabinoid-mediated rescue of striatal LTD and motor deficits in Parkinson’s disease models”. *Nature* 445(7128), pp. 643–647.

[79] Edén Flores-Barrera et al., 2010. “Different corticostriatal integration in spiny projection neurons from direct and indirect pathways”. *Frontiers in systems neuroscience*, p. 15.

[80] Tracy S Gertler, C Savio Chan, and D James Surmeier, 2008. “Dichotomous anatomical properties of adult striatal medium spiny neurons”. *Journal of Neuroscience* 28(43), pp. 10814–10824.

[81] Michelle Day et al., 2008. “Differential excitability and modulation of striatal medium spiny neuron dendrites”. *Journal of Neuroscience* 28(45), pp. 11603–11614.

[82] Henrike Planert, Thomas K Berger, and Gilad Silberberg, 2013. “Membrane properties of striatal direct and indirect pathway neurons in mouse and rat slices and their modulation by dopamine”. *PloS one* 8(3), e57054.

[83] Kai Du et al., 2017. “Cell-type–specific inhibition of the dendritic plateau potential in striatal spiny projection neurons”. *Proceedings of the National Academy of Sciences* 114(36), E7612–E7621.

[84] Adam G Carter and Bernardo L Sabatini, 2004. “State-dependent calcium signaling in dendritic spines of striatal medium spiny neurons”. *Neuron* 44(3), pp. 483–493.

[85] Anna Tokarska and Gilad Silberberg, 2022. “GABAergic interneurons expressing the α2 nicotinic receptor subunit are functionally integrated in the striatal microcircuit”. *Cell Reports* 39(8), p. 110842.

[86] Yasuo Kawaguchi, 1993. “Physiological, morphological, and histochemical characterization of three classes of interneurons in rat neostriatum”. *Journal of Neuroscience* 13(11), pp. 4908–4923.

[87] MF Chesselet and AM Graybiel, 1986. “Striatal neurons expressing somatostatin-like immunoreactivity: evidence for a peptidergic interneuronal system in the cat”. *Neuroscience* 17(3), pp. 547–571.

[88] SR Vincent et al., 1983. “NADPH-diaphorase: A selective histochemical marker for striatal neurons containing both somatostatin-and avian pancreatic polypeptide (APP)-like immunoreactivities”. *Journal of Comparative Neurology* 217(3), pp. 252–263.

[89] Steven R Vincent and Olle Johansson, 1983. “Striatal neurons containing both somatostatin- and avian pancreatic polypeptide (APP)-like immunoreactivities and NADPH-diaphorase activity: a light and electron microscopic study”. *Journal of Comparative Neurology* 217(3), pp. 264–270.

[90] Bruce Thomas Hope et al., 1991. “Neuronal NADPH diaphorase is a nitric oxide synthase.” *Proceedings of the National Academy of Sciences* 88(7), pp. 2811–2814.

[91] Yoshiyuki Kubota and Yasuo Kawaguchi, 2000. “Dependence of GABAergic synaptic areas on the interneuron type and target size”. *Journal of Neuroscience* 20(1), pp. 375–386.

[92] BD Bennett and JP Bolam, 1993. “Characterization of calretinin-immunoreactive structures in the striatum of the rat”. *Brain research* 609(1-2), pp. 137–148.

[93] Y Kubota and Y Kawaguchi, 1994. “Three classes of GABAergic interneurons in neocortex and neostriatum.” *The Japanese journal of physiology* 44, S145–8.

[94] Osvaldo Ibánez-Sandoval et al., 2010. “Electrophysiological and morphological characteristics and synaptic connectivity of tyrosine hydroxylase-expressing neurons in adult mouse striatum”. *Journal of Neuroscience* 30(20), pp. 6999–7016.

[95] Maxime Assous et al., 2018. “Identification and characterization of a novel spontaneously active bursty GABAergic interneuron in the mouse striatum”. *Journal of Neuroscience* 38(25), pp. 5688–5699.

[96] YASUO Kawaguchi, 1992. “Large aspiny cells in the matrix of the rat neostriatum in vitro: physiological identification, relation to the compartments and excitatory postsynaptic currents”. *Journal of Neurophysiology* 67(6), pp. 1669–1682.

[97] Zhongfeng Wang et al., 2006. “Dopaminergic control of corticostriatal long-term synaptic depression in medium spiny neurons is mediated by cholinergic interneurons”. *Neuron* 50(3), pp. 443–452.

[98] Shana M Augustin, Jessica H Chancey, and David M Lovinger, 2018. “Dual dopaminergic regulation of corticostriatal plasticity by cholinergic interneurons and indirect pathway medium spiny neurons”. *Cell reports* 24(11), pp. 2883–2893.

[99] Charles J Wilson, 2005. “The mechanism of intrinsic amplification of hyperpolarizations and spontaneous bursting in striatal cholinergic interneurons”. *Neuron* 45(4), pp. 575–585.

[100] John NJ Reynolds, Brian I Hyland, and Jeff R Wickens, 2004. “Modulation of an afterhyperpolarization by the substantia nigra induces pauses in the tonic firing of striatal cholinergic interneurons”. *Journal of Neuroscience* 24(44), pp. 9870–9877.

[101] JJ Johannes Hjorth et al., 2020. “The microcircuits of striatum in silico”. *Proceedings of the National Academy of Sciences* 117(17), pp. 9554–9565.

[102] Paolo Calabresi et al., 2007. “Dopamine-mediated regulation of corticostriatal synaptic plasticity”. *Trends in neurosciences* 30(5), pp. 211–219.

[103] Purves, 2017. “Neuroscience 6th Edition”. Chap. 6, pp. 113–144.

[104] Valentino Braitenberg and Almut Schüz, 2013. “Cortex: statistics and geometry of neuronal connectivity”.

[105] Jun Shen et al., 1999. “Determination of the rate of the glutamate/glutamine cycle in the human brain by in vivo 13C NMR”. *Proceedings of the National Academy of Sciences* 96(14), pp. 8235–8240.

[106] David Attwell and Simon B Laughlin, 2001. “An energy budget for signaling in the grey matter of the brain”. *Journal of Cerebral Blood Flow & Metabolism* 21(10), pp. 1133–1145.

[107] BjØrnar Hassel and Raymond Dingledine, 2012. “Glutamate and glutamate receptors”. In: *Basic neurochemistry*, ed., pp. 342–366.

[108] Mary C McKenna, 2007. “The glutamate-glutamine cycle is not stoichiometric: Fates of glutamate in brain”. *Journal of neuroscience research* 85(15), pp. 3347–3358.

[109] Lasse K Bak, Arne Schousboe, and Helle S Waagepetersen, 2006. “The glutamate/GABA-glutamine cycle: aspects of transport, neurotransmitter homeostasis and ammonia transfer”. *Journal of neurochemistry* 98(3), pp. 641–653.

[110] Stephen F Traynelis et al., 2010. “Glutamate receptor ion channels: structure, regulation, and function”. *Pharmacological reviews* 62(3), pp. 405–496.

[111] Colleen M Niswender and P Jeffrey Conn, 2010. “Metabotropic glutamate receptors: physiology, pharmacology, and disease”. *Annual review of pharmacology and toxicology* 50, pp. 295–322.

[112] Melinda M Zeron et al., 2002. “Increased sensitivity to N-methyl-D-aspartate receptor-mediated excitotoxicity in a mouse model of Huntington’s disease”. *Neuron* 33(6), pp. 849–860.

[113] Antoine Koehl et al., 2019. “Structural insights into the activation of metabotropic glutamate receptors”. *Nature* 566(7742), pp. 79–84.

[114] Julie Kniazeff et al., 2011. “Dimers and beyond: The functional puzzles of class C GPCRs”. *Pharmacology & therapeutics* 130(1), pp. 9–25.

[115] Etienne Doumazane et al., 2011. “A new approach to analyze cell surface protein complexes reveals specific heterodimeric metabotropic glutamate receptors”. *The FASEB Journal* 25(1), pp. 66–77.

[116] Païkan Marcaggi et al., 2009. “Optical measurement of mGluR1 conformational changes reveals fast activation, slow deactivation, and sensitization”. *Proceedings of the National Academy of Sciences* 106(27), pp. 11388–11393.

[117] Jean-Philippe Pin and Bernhard Bettler, 2016. “Organization and functions of mGlu and GABAB receptor complexes”. *Nature* 540(7631), pp. 60–68.

[118] Eugene Roberts and Sam Frankel, 1950. “γ-Aminobutyric acid in brain: its formation from glutamic acid.”

[119] David L Martin and Karin Rimvall, 1993. “Regulation of γ-aminobutyric acid synthesis in the brain”. *Journal of neurochemistry* 60(2), pp. 395–407.

[120] LL Iversen and JS Kelly, 1975. “Uptake and metabolism of γ-aminobutyric acid by neurones and glial cells”. *Biochemical pharmacology* 24(9), pp. 933–938.

[121] Xiao-Tao Jin et al., 2011. “Localization and function of GABA transporters GAT-1 and GAT-3 in the basal ganglia”. *Frontiers in systems neuroscience* 5, p. 63.

[122] F Anne Stephenson, 1995. “The GABAA receptors.” *Biochemical Journal* 310(Pt 1), p. 1.

[123] Robert L Macdonald and Richard W Olsen, 1994. “GABAA receptor channels”. *Annual review of neuroscience* 17(1), pp. 569–602.

[124] Mark Farrant and Kai Kaila, 2007. “The cellular, molecular and ionic basis of GABAA receptor signalling”. *Progress in brain research* 160, pp. 59–87.

[125] Tibor Koós and James M Tepper, 1999. “Inhibitory control of neostriatal projection neurons by GABAergic interneurons”. *Nature neuroscience* 2(5), pp. 467–472.

[126] Justin Boyes and J Paul Bolam, 2007. “Localization of GABA receptors in the basal ganglia”. *Progress in brain research* 160, pp. 229–243.

[127] Yehezkel Ben-Ari, 2002. “Excitatory actions of gaba during development: the nature of the nurture”. *Nature Reviews Neuroscience* 3(9), pp. 728–739.

[128] AR Calver, CH Davies, and MN Pangalos, 2002. “GABAB receptors: from monogamy to promiscuity”. *Neurosignals* 11(6), pp. 299–314.

[129] Purves, 2017. “Neuroscience 6th Edition”. Chap. 6, pp. 131–135.

[130] Andreas Heinz and Florian Schlagenhauf, 2010. “Dopaminergic dysfunction in schizophrenia: salience attribution revisited”. *Schizophrenia bulletin* 36(3), pp. 472–485.

[131] Gail Tripp and Jeff Wickens, 2012. “Reinforcement, dopamine and rodent models in drug development for ADHD”. *Neurotherapeutics* 9(3), pp. 622–634.

[132] Masaya Segawa, 2011. “Hereditary progressive dystonia with marked diurnal fluctuation”. *Brain and Development* 33(3), pp. 195–201.

[133] José M Musacchio, 1975. “Enzymes involved in the biosynthesis and degradation of catecholamines”. In: *Biochemistry of biogenic amines*, ed., pp. 1–35.

[134] W von Birkmayer and Oleh Hornykiewicz, 1961. “The L-3, 4-dioxyphenylalanine (DOPA)-effect in Parkinson-akinesia”. *Wiener klinische Wochenschrift* 73, pp. 787–788.

[135] Farrukh A Chaudhry, Robert H Edwards, and Frode Fonnum, 2008. “Vesicular neurotransmitter transporters as targets for endogenous and exogenous toxic substances”. *Annu. Rev. Pharmacol. Toxicol.* 48(1), pp. 277–301.

[136] Hui Zhang and David Sulzer, 2012. “Regulation of striatal dopamine release by presynaptic auto-and heteroreceptors”. *Basal ganglia* 2(1), pp. 5–13.

[137] Jacob Eriksen, Trine Nygaard Jørgensen, and Ulrik Gether, 2010. “Regulation of dopamine transporter function by protein-protein interactions: new discoveries and methodological challenges”. *Journal of neurochemistry* 113(1), pp. 27–41.

[138] Graeme Eisenhofer, Irwin J Kopin, and David S Goldstein, 2004. “Catecholamine metabolism: a contemporary view with implications for physiology and medicine”. *Pharmacological reviews* 56(3), pp. 331–349.

[139] Cristina Missale et al., 1998. “Dopamine receptors: from structure to function”. *Physiological reviews*.

[140] Daniela Vallone, Roberto Picetti, and Emiliana Borrelli, 2000. “Structure and function of dopamine receptors”. *Neuroscience & biobehavioral reviews* 24(1), pp. 125–132.

[141] Kim A Neve, Jeremy K Seamans, and Heather Trantham-Davidson, 2004. “Dopamine receptor signaling”. *Journal of receptors and signal transduction* 24(3), pp. 165–205.

[142] Anita Sidhu and Hyman B Niznik, 2000. “Coupling of dopamine receptor subtypes to multiple and diverse G proteins”. *International Journal of Developmental Neuroscience* 18(7), pp. 669–677.

[143] Henry Hallett Dale, 1914. “The action of certain esters and ethers of choline, and their relation to muscarine”. *Journal of Pharmacology and Experimental Therapeutics* 6(2), pp. 147–190.

[144] Purves, 2017. “Neuroscience 6th Edition”. Chap. 6, pp. 113–120.

[145] Hideki Kawai, Ronit Lazar, and Raju Metherate, 2007. “Nicotinic control of axon excitability regulates thalamocortical transmission”. *Nature neuroscience* 10(9), pp. 1168–1175.

[146] Margaret E Rice and Stephanie J Cragg, 2004. “Nicotine amplifies reward-related dopamine signals in striatum”. *Nature neuroscience* 7(6), pp. 583–584.

[147] Susan Wonnacott, 1997. “Presynaptic nicotinic ACh receptors”. *Trends in neurosciences* 20(2), pp. 92–98.

[148] Hui Zhang and David Sulzer, 2004. “Frequency-dependent modulation of dopamine release by nicotine”. *Nature neuroscience* 7(6), pp. 581–582.

[149] Palmer Taylor and Joan Heller Brown, 1999. “Synthesis, storage and release of acetylcholine”. In: *Basic Neurochemistry: Molecular, Cellular and Medical Aspects. 6th edition*, ed.

[150] Stanley M Parsons et al., 1987. “Acetylcholine transport: fundamental properties and effects of pharmacologic agents”. *Ann NY Acad Sci* 493(1), pp. 220–33.

[151] S Hussain and S Davanger, 2011. “The discovery of the soluble N-ethylmaleimide-sensitive factor attachment protein receptor complex and the molecular regulation of synaptic vesicle transmitter release: the 2010 Kavli Prize in neuroscience”. *Neuroscience* 190, pp. 12–20.

[152] Jing Han, Kristyna Pluhackova, and Rainer A Böckmann, 2017. “The multifaceted role of SNARE proteins in membrane fusion”. *Frontiers in physiology* 8, p. 5.

[153] Daniel M Quinn, 1987. “Acetylcholinesterase: enzyme structure, reaction dynamics, and virtual transition states”. *Chemical reviews* 87(5), pp. 955–979.

[154] MICHAEL J Marks et al., 1986. “Nicotinic binding sites in rat and mouse brain: comparison of acetylcholine, nicotine, and alpha-bungarotoxin.” *Molecular pharmacology* 30(5), pp. 427–436.

[155] John A Dani and Daniel Bertrand, 2007. “Nicotinic acetylcholine receptors and nicotinic cholinergic mechanisms of the central nervous system”. *Annu. Rev. Pharmacol. Toxicol.* 47(1), pp. 699–729.

[156] Manickavasagom Alkondon and Edson X Albuquerque, 2004. “The nicotinic acetylcholine receptor subtypes and their function in the hippocampus and cerebral cortex”. *Progress in brain research* 145, pp. 109–120.

[157] Cecilia Gotti et al., 2009. “Structural and functional diversity of native brain neuronal nicotinic receptors”. *Biochemical pharmacology* 78(7), pp. 703–711.

[158] Roger L Papke, 2014. “Merging old and new perspectives on nicotinic acetylcholine receptors”. *Biochemical pharmacology* 89(1), pp. 1–11.

[159] Robert Fredriksson et al., 2003. “The G-protein-coupled receptors in the human genome form five main families. Phylogenetic analysis, paralogon groups, and fingerprints”. *Molecular pharmacology* 63(6), pp. 1256–1272.

[160] EC Hulme, NJM Birdsall, and NJ Buckley, 1990. “Muscarinic receptor subtypes”. *Annual review of pharmacology and toxicology* 30(1), pp. 633–673.

[161] Andrew C Kruse et al., 2014. “Muscarinic acetylcholine receptors: novel opportunities for drug development”. *Nature reviews Drug discovery* 13(7), pp. 549–560.

[162] Jürgen Wess, Richard M Eglen, and Dinesh Gautam, 2007. “Muscarinic acetylcholine receptors: mutant mice provide new insights for drug development”. *Nature reviews Drug discovery* 6(9), pp. 721–733.

[163] Tim VP Bliss and Terje Lømo, 1973. “Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path”. *The Journal of physiology* 232(2), pp. 331–356.

[164] GL Collingridge, CE Herron, and RA Lester, 1988. “Synaptic activation of N-methyl-D-aspartate receptors in the Schaffer collateral-commissural pathway of rat hippocampus.” *The Journal of physiology* 399(1), pp. 283–300.

[165] Gary Lynch et al., 1983. “Intracellular injections of EGTA block induction of hippocampal long-term potentiation”. *Nature* 305(5936), pp. 719–721.

[166] Manfred Krug, B Lössner, and Tilmann Ott, 1984. “Anisomycin blocks the late phase of long-term potentiation in the dentate gyrus of freely moving rats”. *Brain research bulletin* 13(1), pp. 39–42.

[167] Serena M Dudek and Mark F Bear, 1992. “Homosynaptic long-term depression in area CA1 of hippocampus and effects of N-methyl-D-aspartate receptor blockade.” *Proceedings of the National Academy of Sciences* 89(10), pp. 4363–4367.

[168] Rosel M Mulkey, Caroline E Herron, and Robert C Malenka, 1993. “An essential role for protein phosphatases in hippocampal long-term depression”. *Science* 261(5124), pp. 1051–1055.

[169] Masao Ito and Masanobu Kano, 1982. “Long-lasting depression of parallel fiber-Purkinje cell transmission induced by conjunctive stimulation of parallel fibers and climbing fibers in the cerebellar cortex”. *Neuroscience letters* 33(3), pp. 253–258.

[170] David J Linden et al., 1991. “A long-term depression of AMPA currents in cultured cerebellar Purkinje neurons”. *Neuron* 7(1), pp. 81–89.

[171] Paolo Calabresi et al., 1992. “Long-term synaptic depression in the striatum: physiological and pharmacological characterization”. *Journal of Neuroscience* 12(11), pp. 4224–4233.

[172] Louise Adermark and David M Lovinger, 2007. “Combined activation of L-type Ca2+ channels and synaptic transmission is sufficient to induce striatal long-term depression”. *Journal of Neuroscience* 27(25), pp. 6781–6787.

[173] Vincenza Bagetta et al., 2011. “Dopamine-dependent long-term depression is expressed in striatal spiny neurons of both direct and indirect pathways: implications for Parkinson’s disease”. *Journal of Neuroscience* 31(35), pp. 12513–12522.

[174] Kristen K Ade and David M Lovinger, 2007. “Anandamide regulates postnatal development of long-term synaptic plasticity in the rat dorsolateral striatum”. *Journal of Neuroscience* 27(9), pp. 2403–2409.

[175] Talia N Lerner and Anatol C Kreitzer, 2012. “RGS4 is required for dopaminergic control of striatal LTD and susceptibility to parkinsonian motor deficits”. *Neuron* 73(2), pp. 347–359.

[176] Weixing Shen et al., 2008. “Dichotomous dopaminergic control of striatal synaptic plasticity”. *Science* 321(5890), pp. 848–851.

[177] Paolo Gubellini et al., 2001. “Selective involvement of mGlu1 receptors in corticostriatal LTD”. *Neuropharmacology* 40(7), pp. 839–846.

[178] Paolo Gubellini et al., 2004. “Metabotropic glutamate receptors and striatal synaptic plasticity: implications for neurological diseases”. *Progress in neurobiology* 74(5), pp. 271–300.

[179] Anu G Nair et al., 2014. “Modeling intracellular signaling underlying striatal function in health and disease”. In: *Progress in molecular biology and translational science*, ed., pp. 277–304.

[180] Jean-Martin Beaulieu and Raul R Gainetdinov, 2011. “The physiology, signaling, and pharmacology of dopamine receptors”. *Pharmacological reviews* 63(1), pp. 182–217.

[181] John W Kebabian and Paul Greengard, 1971. “Dopamine-sensitive adenyl cyclase: possible role in synaptic transmission”. *Science* 174(4016), pp. 1346–1349.

[182] Antonello Bonci and F Woodward Hopf, 2005. “The dopamine D2 receptor: new surprises from an old friend”. *Neuron* 47(3), pp. 335–338.

[183] D James Surmeier et al., 2007. “D1 and D2 dopamine-receptor modulation of striatal glutamatergic signaling in striatal medium spiny neurons”. *Trends in neurosciences* 30(5), pp. 228–235.

[184] Paul Greengard, Patrick B Allen, and Angus C Nairn, 1999. “Beyond the dopamine receptor: the DARPP-32/protein phosphatase-1 cascade”. *Neuron* 23(3), pp. 435–447.

[185] Helen S Bateup et al., 2008. “Cell type–specific regulation of DARPP-32 phosphorylation by psychostimulant and antipsychotic drugs”. *Nature neuroscience* 11(8), pp. 932–939.

[186] Anatol C Kreitzer and Robert C Malenka, 2005. “Dopamine modulation of state-dependent endocannabinoid release and long-term depression in the striatum”. *Journal of Neuroscience* 25(45), pp. 10537–10545.

[187] Motokazu Uchigashima et al., 2007. “Subcellular arrangement of molecules for 2-arachidonoyl-glycerol-mediated retrograde signaling and its physiological contribution to synaptic modulation in the striatum”. *Journal of Neuroscience* 27(14), pp. 3663–3676.

[188] Peter M Zygmunt et al., 1999. “Vanilloid receptors on sensory nerves mediate the vasodilator action of anandamide”. *Nature* 400(6743), pp. 452–457.

[189] Miles Herkenham et al., 1991. “Characterization and localization of cannabinoid receptors in rat brain: a quantitative in vitro autoradiographic study”. *Journal of Neuroscience* 11(2), pp. 563–583.

[190] Giovanni Marsicano and Beat Lutz, 1999. “Expression of the cannabinoid receptor CB1 in distinct neuronal subpopulations in the adult mouse forebrain”. *European journal of neuroscience* 11(12), pp. 4213–4225.

[191] Rachel I Wilson and Roger A Nicoll, 2002. “Endocannabinoid signaling in the brain”. *science* 296(5568), pp. 678–682.

[192] Gregory L Gerdeman, Jennifer Ronesi, and David M Lovinger, 2002. “Postsynaptic endocannabinoid release is critical to long-term depression in the striatum”. *Nature neuroscience* 5(5), pp. 446–451.

[193] Alessandra Musella et al., 2009. “TRPV1 channels facilitate glutamate transmission in the striatum”. *Molecular and Cellular Neuroscience* 40(1), pp. 89–97.

[194] Brad A Grueter, Gabor Brasnjo, and Robert C Malenka, 2010. “Postsynaptic TRPV1 triggers cell type–specific long-term depression in the nucleus accumbens”. *Nature neuroscience* 13(12), pp. 1519–1525.

[195] Henry H Yin et al., 2006. “The role of protein synthesis in striatal long-term depression”. *Journal of Neuroscience* 26(46), pp. 11811–11820.

[196] Luisa Speranza et al., 2021. “Dopamine: the neuromodulator of long-term synaptic plasticity, reward and movement control”. *Cells* 10(4), p. 735.

[197] Antonio Pisani et al., 2007. “Re-emergence of striatal cholinergic interneurons in movement disorders”. *Trends in neurosciences* 30(10), pp. 545–553.

[198] Paolo Calabresi et al., 2014. “Direct and indirect pathways of basal ganglia: a critical reappraisal”. *Nature neuroscience* 17(8), pp. 1022–1030.

[199] Jean-Martin Beaulieu, Stefano Espinoza, and Raul R Gainetdinov, 2015. “Dopamine receptors–IUPHAR R eview 13”. *British journal of pharmacology* 172(1), pp. 1–23.

[200] Raul R Gainetdinov et al., 2004. “Desensitization of G protein–coupled receptors and neuronal functions”. *Annu. Rev. Neurosci.* 27(1), pp. 107–144.

[201] Shana M Augustin et al., 2014. “Cyclic AMP and afferent activity govern bidirectional synaptic plasticity in striatopallidal neurons”. *Journal of Neuroscience* 34(19), pp. 6692–6699.

[202] Paolo Calabresi et al., 2006. “A convergent model for cognitive dysfunctions in Parkinson’s disease: the critical dopamine–acetylcholine synaptic balance”. *The Lancet Neurology* 5(11), pp. 974–983.

[203] Stephanie J Cragg, 2006. “Meaningful silences: how dopamine listens to the ACh pause”. *Trends in neurosciences* 29(3), pp. 125–131.

[204] Elodie Fino, Jacques Glowinski, and Laurent Venance, 2005. “Bidirectional activity-dependent plasticity at corticostriatal synapses”. *Journal of Neuroscience* 25(49), pp. 11279–11287.

[205] Cristiano Nazzaro et al., 2012. “SK channel modulation rescues striatal plasticity and control over habit in cannabinoid tolerance”. *Nature neuroscience* 15(2), pp. 284–293.

[206] A Giuffrida et al., 1999. “Dopamine activation of endogenous cannabinoid signaling in dorsal striatum”. *Nature neuroscience* 2(4), pp. 358–363.

[207] Ki-Wug Sung, Sukwoo Choi, and David M Lovinger, 2001. “Activation of group I mGluRs is necessary for induction of long-term depression at striatal synapses”. *Journal of neurophysiology* 86(5), pp. 2405–2412.

[208] Sukwoo Choi and David M Lovinger, 1997. “Decreased probability of neurotransmitter release underlies striatal long-term depression and postnatal development of corticostriatal synapses”. *Proceedings of the National Academy of Sciences* 94(6), pp. 2665–2670.

[209] Talia N Lerner et al., 2010. “Endocannabinoid signaling mediates psychomotor activation by adenosine A2A antagonists”. *Journal of Neuroscience* 30(6), pp. 2160–2164.

[210] Talia N Lerner and Anatol C Kreitzer, 2011. “Neuromodulatory control of striatal plasticity and behavior”. *Current opinion in neurobiology* 21(2), pp. 322–327.

[211] Mazen A Kheirbek et al., 2009. “Adenylyl cyclase type 5 contributes to corticostriatal plasticity and striatum-dependent learning”. *Journal of Neuroscience* 29(39), pp. 12115–12124.

[212] Jongrye Jeon et al., 2010. “A subpopulation of neuronal M4 muscarinic acetylcholine receptors plays a critical role in modulating dopamine-dependent behaviors”. *Journal of Neuroscience* 30(6), pp. 2396–2405.

[213] Barbara Picconi et al., 2011. “Inhibition of phosphodiesterases rescues striatal long-term depression and reduces levodopa-induced dyskinesia”. *Brain* 134(2), pp. 375–387.

[214] Malcolm P Caulfield and Nigel JM Birdsall, 1998. “International Union of Pharmacology. XVII. Classification of muscarinic acetylcholine receptors”. *Pharmacological reviews* 50(2), pp. 279–290.

[215] Gloria H Biddlecome, Gabriel Berstein, and Elliott M Ross, 1996. “Regulation of phospholipase C-β1 by Gq and m1 muscarinic cholinergic receptor: Steady-state balance of receptor-mediated activation and GTPase-activating protein-promoted deactivation”. *Journal of Biological Chemistry* 271(14), pp. 7999–8007.

[216] Daniele Piomelli, 2003. “The molecular logic of endocannabinoid signalling”. *Nature Reviews Neuroscience* 4(11), pp. 873–884.

[217] Michael J Berridge and Robin F Irvine, 1984. “Inositol trisphosphate, a novel second messenger in cellular signal transduction”. *Nature* 312(5992), pp. 315–321.

[218] Kay Ahn, Michele K McKinney, and Benjamin F Cravatt, 2008. “Enzymatic pathways that regulate endocannabinoid signaling in the nervous system”. *Chemical reviews* 108(5), pp. 1687–1707.

[219] Tamas Balla, 2013. “Phosphoinositides: tiny lipids with giant impact on cell regulation”. *Physiological reviews* 93(3), pp. 1019–1137.

[220] Mabel R Hokin and Lowell E Hokin, 1959. “The synthesis of phosphatidic acid from diglyceride and adenosine triphosphate in extracts of brain microsomes”. *Journal of Biological Chemistry* 234(6), pp. 1381–1386.

[221] Bernard W Agranoff, Roy M Bradley, and Roscoe O Brady, 1958. “The enzymatic synthesis of inositol phosphatide”. *Journal of Biological Chemistry* 233(5), pp. 1077–1083.

[222] Hailin Zhang et al., 2003. “PIP2 activates KCNQ channels, and its hydrolysis underlies receptor-mediated inhibition of M currents”. *Neuron* 37(6), pp. 963–975.

[223] Byung-Chang Suh and Bertil Hille, 2002. “Recovery from muscarinic modulation of M current channels requires phosphatidylinositol 4, 5-bisphosphate synthesis”. *Neuron* 35(3), pp. 507–520.

[224] Hong-Sheng Wang et al., 1998. “KCNQ2 and KCNQ3 potassium channel subunits: molecular correlates of the M-channel”. *Science* 282(5395), pp. 1890–1893.

[225] DA Brown and PR Adams, 1980. “Muscarinic suppression of a novel voltage-sensitive K+ current in a vertebrate neurone”. *Nature* 283(5748), pp. 673–676.

[226] DA Brown, 1983. “Slow cholinergic excitation—a mechanism for increasing neuronal excitability”. *Trends in Neurosciences* 6, pp. 302–307.

[227] Charlotte Piette et al., 2020. “Lights on endocannabinoid-mediated synaptic potentiation”. *Frontiers in molecular neuroscience* 13, p. 132.

[228] JC Stoof and JW Kebabian, 1981. “Opposing roles for D-1 and D-2 dopamine receptors in efflux of cyclic AMP from rat neostriatum”. *Nature* 294(5839), pp. 366–368.

[229] David R Sibley and Frederick J Monsma Jr, 1992. “Molecular biology of dopamine receptors”. *Trends in pharmacological sciences* 13, pp. 61–69.

[230] JC Corvol et al., 2001. “Gαolf is necessary for coupling D1 and A2a receptors to adenylyl cyclase in the striatum”. *Journal of neurochemistry* 76(5), pp. 1585–1588.

[231] Anu G Nair et al., 2019. “The high efficacy of muscarinic M4 receptor in D1 medium spiny neurons reverses striatal hyperdopaminergia”. *Neuropharmacology* 146, pp. 74–83.

[232] Omar Gutierrez-Arenas, Olivia Eriksson, and Jeanette Hellgren Kotaleski, 2014. “Segregation and crosstalk of D1 receptor-mediated activation of ERK in striatal medium spiny neurons upon acute administration of psychostimulants”. *PLoS computational biology* 10(1), e1003445.

[233] Carmen Blanco-Aparicio, Josema Torres, and Rafael Pulido, 1999. “A novel regulatory mechanism of MAP kinases activation and nuclear translocation mediated by PKA and the PTP-SL tyrosine phosphatase”. *The Journal of cell biology* 147(6), pp. 1129–1136.

[234] Surojit Paul et al., 2000. “The Dopamine/D1 receptor mediates the phosphorylation and inactivation of the protein tyrosine phosphatase STEP via a PKA-dependent pathway”. *Journal of Neuroscience* 20(15), pp. 5630–5638.

[235] Surojit Paul et al., 2003. “NMDA-mediated activation of the tyrosine phosphatase STEP regulates the duration of ERK signaling”. *Nature neuroscience* 6(1), pp. 34–42.

[236] Emmanuel Valjent et al., 2005. “Regulation of a protein phosphatase cascade allows convergent dopamine and glutamate signals to activate ERK in the striatum”. *Proceedings of the National Academy of Sciences* 102(2), pp. 491–496.

[237] Manju Saxena et al., 1999. “Crosstalk between cAMP-dependent kinase and MAP kinase through a protein tyrosine phosphatase”. *Nature cell biology* 1(5), pp. 305–310.

[238] Wai Yiu Cheung, 1980. “Calmodulin plays a pivotal role in cellular regulation”. *Science* 207(4426), pp. 19–27.

[239] J Michael Bradshaw et al., 2003. “An ultrasensitive Ca2+/calmodulin-dependent protein kinase II-protein phosphatase 1 switch facilitates specificity in postsynaptic calcium signaling”. *Proceedings of the National Academy of Sciences* 100(18), pp. 10512–10517.

[240] Andres Barria et al., 1997. “Regulatory phosphorylation of AMPA-type glutamate receptors by CaM-KII during long-term potentiation”. *Science* 276(5321), pp. 2042–2045.

[241] Per Svenningsson et al., 2004. “DARPP-32: an integrator of neurotransmission”. *Annu. Rev. Pharmacol. Toxicol.* 44(1), pp. 269–296.

[242] Jung-Hyuck Ahn et al., 2007. “The B" /PR72 subunit mediates Ca2+-dependent dephosphorylation of DARPP-32 by protein phosphatase 2A”. *Proceedings of the National Academy of Sciences* 104(23), pp. 9876–9881.

[243] HC Hemmings et al., 1989. “ARPP-21, a cyclic AMP-regulated phosphoprotein (Mr= 21,000) enriched in dopamine-innervated brain regions: Amino acid sequence of the site phosphorylated by cyclic AMP in intact cells and kinetic studies of its phosphorylation in vitro”. *Journal of Biological Chemistry* 264(13), pp. 7726–7733.

[244] Hugh C Hemmings Jr et al., 1984. “DARPP-32, a dopamine-regulated neuronal phosphoprotein, is a potent inhibitor of protein phosphatase-1”. *Nature* 310(5977), pp. 503–505.

[245] Akinori Nishi et al., 1999. “Role of calcineurin and protein phosphatase-2A in the regulation of DARPP-32 dephosphorylation in neostriatal neurons”. *Journal of neurochemistry* 72(5), pp. 2015–2021.

[246] Akinori Nishi et al., 2000. “Amplification of dopaminergic signaling by a positive feedback loop”. *Proceedings of the National Academy of Sciences* 97(23), pp. 12840–12845.

[247] Akinori Nishi, Gretchen L Snyder, and Paul Greengard, 1997. “Bidirectional regulation of DARPP-32 phosphorylation by dopamine”. *Journal of Neuroscience* 17(21), pp. 8147–8155.

[248] MM King and CY Huang, 1984. “The calmodulin-dependent activation and deactivation of the phosphoprotein phosphatase, calcineurin, and the effect of nucleotides, pyrophosphate, and divalent metal ions. Identification of calcineurin as a Zn and Fe metalloenzyme.” *Journal of Biological Chemistry* 259(14), pp. 8847–8856.

[249] Frederic Desdouits et al., 1995. “Dopamine-and cAMP-regulated phosphoprotein DARPP-32: phosphorylation of Ser-137 by casein kinase I inhibits dephosphorylation of Thr-34 by calcineurin.” *Proceedings of the National Academy of Sciences* 92(7), pp. 2682–2685.

[250] Feng Liu et al., 2001. “Regulation of cyclin-dependent kinase 5 and casein kinase 1 by metabotropic glutamate receptors”. *Proceedings of the National Academy of Sciences* 98(20), pp. 11062–11068.

[251] James A Bibb et al., 1999. “Phosphorylation of DARPP-32 by Cdk5 modulates dopamine signalling in neurons”. *Nature* 402(6762), pp. 669–671.

[252] Whittemore G Tingley et al., 1997. “Characterization of protein kinase A and protein kinase C phosphorylation of the N-methyl-D-aspartate receptor NR1 subunit using phosphorylation site-specific antibodies”. *Journal of Biological Chemistry* 272(8), pp. 5157–5166.

[253] Yoshimi Takai et al., 1979. “Unsaturated diacylglycerol as a possible messenger for the activation of calcium-activated, phospholipid-dependent protein kinase system”. *Biochemical and biophysical research communications* 91(4), pp. 1218–1224.

[254] A Kishimoto et al., 1980. “Activation of calcium and phospholipid-dependent protein kinase by diacylglycerol, its possible relation to phosphatidylinositol turnover.” *Journal of Biological Chemistry* 255(6), pp. 2273–2276.

[255] Katherine W Roche et al., 1996. “Characterization of multiple phosphorylation sites on the AMPA receptor GluR1 subunit”. *Neuron* 16(6), pp. 1179–1188.

[256] Jiean Huang et al., 2007. “Inhibition of Gαq-dependent PLC-β1 activity by PKG and PKA is mediated by phosphorylation of RGS4 and GRK2”. *American Journal of Physiology-Cell Physiology* 292(1), pp. C200–C208.

[257] KANG Tsou, Gretchen L Snyder, and Paul Greengard, 1993. “Nitric oxide/cGMP pathway stimulates phosphorylation of DARPP-32, a dopamine-and cAMP-regulated phosphoprotein, in the substantia nigra.” *Proceedings of the National Academy of Sciences* 90(8), pp. 3462–3465.

[258] Hugh C Hemmings Jr, AC Nairn, and P Greengard, 1984. “DARPP-32, a dopamine-and adenosine 3 ‘: 5 ‘-monophosphate-regulated neuronal phosphoprotein. II. Comparison of the kinetics of phosphorylation of DARPP-32 and phosphatase inhibitor 1.” *Journal of Biological Chemistry* 259(23), pp. 14491–14497.

[259] MM King et al., 1984. “Mammalian brain phosphoproteins as substrates for calcineurin.” *Journal of Biological Chemistry* 259(13), pp. 8080–8083.

[260] Gregg L Caporaso et al., 2000. “Drugs of abuse modulate the phosphorylation of ARPP-21, a cyclic AMP-regulated phosphoprotein enriched in the basal ganglia”. *Neuropharmacology* 39(9), pp. 1637–1644.

[261] SV Rakhilin et al., 2004. “A network of control mediated by regulator of calcium/calmodulin-dependent signaling”. *Science* 306(5696), pp. 698–701.

[262] Julie A Saugstad et al., 1998. “RGS4 inhibits signaling by group I metabotropic glutamate receptors”. *Journal of Neuroscience* 18(3), pp. 905–913.

[263] Paolo Calabresi et al., 1992. “Long-term potentiation in the striatum is unmasked by removing the voltage-dependent magnesium block of NMDA receptor channels”. *European Journal of Neuroscience* 4(10), pp. 929–935.

[264] Elodie Fino et al., 2010. “Distinct coincidence detectors govern the corticostriatal spike timing-dependent plasticity”. *The Journal of physiology* 588(16), pp. 3045–3062.

[265] Mai T Dang et al., 2006. “Disrupted motor learning and long-term synaptic plasticity in mice lacking NMDAR1 in the striatum”. *Proceedings of the national academy of sciences* 103(41), pp. 15254–15259.

[266] Marc Flajolet et al., 2008. “FGF acts as a co-transmitter through adenosine A2A receptor to regulate synaptic plasticity”. *Nature neuroscience* 11(12), pp. 1402–1409.

[267] Sietske M Schotanus and Karima Chergui, 2008. “Dopamine D1 receptors and group I metabotropic glutamate receptors contribute to the induction of long-term potentiation in the nucleus accumbens”. *Neuropharmacology* 54(5), pp. 837–844.

[268] Diego Centonze et al., 1999. “Unilateral dopamine denervation blocks corticostriatal LTP”. *Journal of neurophysiology* 82(6), pp. 3575–3579.

[269] Jason ND Kerr and JR Wickens, 2001. “Dopamine D-1/D-5 receptor activation is required for long-term potentiation in the rat neostriatum in vitro”. *Journal of neurophysiology* 85(1), pp. 117–124.

[270] Paolo Calabresi et al., 2000. “Dopamine and cAMP-regulated phosphoprotein 32 kDa controls both striatal long-term depression and long-term potentiation, opposing forms of synaptic plasticity”. *Journal of Neuroscience* 20(22), pp. 8443–8451.

[271] John G Partridge, Ka-Choi Tang, and David M Lovinger, 2000. “Regional and postnatal heterogeneity of activity-dependent long-term changes in synaptic efficacy in the dorsal striatum”. *Journal of Neurophysiology* 84(3), pp. 1422–1429.

[272] Diego Centonze et al., 2003. “Distinct roles of D1 and D5 dopamine receptors in motor activity and striatal synaptic plasticity”. *Journal of Neuroscience* 23(24), pp. 8506–8512.

[273] Helen S Bateup et al., 2010. “Distinct subclasses of medium spiny neurons differentially regulate striatal motor behaviors”. *Proceedings of the National Academy of Sciences* 107(33), pp. 14845–14850.

[274] Jessica A Murphy et al., 2014. “Phosphorylation of Ser1166 on GluN2B by PKA is critical to synaptic NMDA receptor function and Ca2+ signaling in spines”. *Journal of Neuroscience* 34(3), pp. 869–879.

[275] Jorge Flores-Hernandez et al., 2002. “Dopamine enhancement of NMDA currents in dissociated medium-sized striatal neurons: role of D1 receptors and DARPP-32”. *Journal of neurophysiology* 88(6), pp. 3010–3020.

[276] Yousheng Jia, Christine M Gall, and Gary Lynch, 2010. “Presynaptic BDNF promotes postsynaptic long-term potentiation in the dorsal striatum”. *Journal of Neuroscience* 30(43), pp. 14440–14445.

[277] Paolo Calabresi et al., 1999. “Activation of M1-like muscarinic receptors is required for the induction of corticostriatal LTP”. *Neuropharmacology* 38(2), pp. 323–326.

[278] Paolo Calabresi et al., 1998. “Blockade of M2-like muscarinic receptors enhances long-term potentiation at corticostriatal synapses”. *European Journal of Neuroscience* 10(9), pp. 3020–3023.

[279] Michael J Higley and Bernardo L Sabatini, 2010. “Competitive regulation of synaptic Ca2+ influx by D2 dopamine and A2A adenosine receptors”. *Nature neuroscience* 13(8), pp. 958–966.

[280] Laurent Ladepeche et al., 2013. “Single-molecule imaging of the functional crosstalk between surface NMDA and dopamine D1 receptors”. *Proceedings of the National Academy of Sciences* 110(44), pp. 18005–18010.

[281] Lena Scott et al., 2006. “Allosteric changes of the NMDA receptor trap diffusible dopamine 1 receptors in spines”. *Proceedings of the National Academy of Sciences* 103(3), pp. 762–767.

[282] Joo Min Park et al., 2013. “A prolyl-isomerase mediates dopamine-dependent plasticity and cocaine motor sensitization”. *Cell* 154(3), pp. 637–650.

[283] Verena Pawlak and Jason ND Kerr, 2008. “Dopamine receptor activation is required for corticostriatal spike-timing-dependent plasticity”. *Journal of Neuroscience* 28(10), pp. 2435–2446.

[284] Vincent Paille et al., 2013. “GABAergic circuits control spike-timing-dependent plasticity”. *Journal of Neuroscience* 33(22), pp. 9353–9363.

[285] Yihui Cui et al., 2015. “Endocannabinoids mediate bidirectional striatal spike-timing-dependent plasticity”. *The Journal of Physiology* 593(13), pp. 2833–2849.

[286] Yihui Cui et al., 2016. “Endocannabinoid dynamics gate spike-timing dependent depression and potentiation”. *Elife* 5, e13185.

[287] Yihui Cui et al., 2018. “Robustness of STDP to spike timing jitter”. *Scientific reports* 8(1), p. 8139.

[288] Hao Xu et al., 2018. “Dopamine–endocannabinoid interactions mediate spike-timing-dependent potentiation in the striatum”. *Nature communications* 9(1), p. 4118.

[289] Giuseppe Gangarossa et al., 2020. “BDNF controls bidirectional endocannabinoid plasticity at corticostriatal synapses”. *Cerebral Cortex* 30(1), pp. 197–214.

[290] Jan M Schulz, Peter Redgrave, and John NJ Reynolds, 2010. “Cortico-striatal spike-timing dependent plasticity after activation of subcortical pathways”. *Frontiers in synaptic neuroscience* 2, p. 1462.

[291] Simon D Fisher et al., 2017. “Reinforcement determines the timing dependence of corticostriatal synaptic plasticity in vivo”. *Nature communications* 8(1), p. 334.

[292] Gaëtan Vignoud, Laurent Venance, and Jonathan D Touboul, 2024. “Anti-Hebbian plasticity drives sequence learning in striatum”. *Communications biology* 7(1), p. 555.

[293] Olaf Sporns, Giulio Tononi, and Rolf Kötter, 2005. “The human connectome: a structural description of the human brain”. *PLoS computational biology* 1(4), e42.

[294] Ed Bullmore and Olaf Sporns, 2009. “Complex brain networks: graph theoretical analysis of structural and functional systems”. *Nature reviews neuroscience* 10(3), pp. 186–198.

[295] Hugh R Wilson and Jack D Cowan, 1972. “Excitatory and inhibitory interactions in localized populations of model neurons”. *Biophysical journal* 12(1), pp. 1–24.

[296] Hugh R Wilson and Jack D Cowan, 1973. “A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue”. *Kybernetik* 13(2), pp. 55–80.

[297] Michael Breakspear, 2017. “Dynamic models of large-scale brain activity”. *Nature neuroscience* 20(3), pp. 340–352.

[298] Stephen Coombes, 2010. “Large-scale neural dynamics: simple and complex”. *NeuroImage* 52(3), pp. 731–739.

[299] Gustavo Deco et al., 2008. “The dynamic brain: from spiking neurons to neural masses and cortical fields”. *PLoS computational biology* 4(8), e1000092.

[300] Federico Tesler et al., 2023. “A multiscale model of striatum microcircuit dynamics”. *bioRxiv*, pp. 2023–12.

[301] Gustavo Deco, Viktor K Jirsa, and Anthony R McIntosh, 2013. “Resting brains never rest: computational insights into potential cognitive architectures”. *Trends in neurosciences* 36(5), pp. 268–274.

[302] Gustavo Deco, Viktor K Jirsa, and Anthony R McIntosh, 2011. “Emerging concepts for the dynamical organization of resting-state activity in the brain”. *Nature reviews neuroscience* 12(1), pp. 43–56.

[303] Henry Markram et al., 2015. “Reconstruction and simulation of neocortical microcircuitry”. *Cell* 163(2), pp. 456–492.

[304] Romain Brette et al., 2007. “Simulation of networks of spiking neurons: a review of tools and strategies”. *Journal of computational neuroscience* 23, pp. 349–398.

[305] Michael L Hines and Nicholas T Carnevale, 1997. “The NEURON simulation environment”. *Neural computation* 9(6), pp. 1179–1209.

[306] Filip Ponulak and Andrzej Kasinski, 2011. “Introduction to spiking neural networks: Information processing, learning and applications”. *Acta neurobiologiae experimentalis* 71(4), pp. 409–433.

[307] Wolfgang Maass, 1997. “Networks of spiking neurons: the third generation of neural network models”. *Neural networks* 10(9), pp. 1659–1671.

[308] Guo-qiang Bi and Mu-ming Poo, 1998. “Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type”. *Journal of neuroscience* 18(24), pp. 10464–10472.

[309] Abigail Morrison, Markus Diesmann, and Wulfram Gerstner, 2008. “Phenomenological models of synaptic plasticity based on spike timing”. *Biological cybernetics* 98, pp. 459–478.

[310] Sen Song, Kenneth D Miller, and Larry F Abbott, 2000. “Competitive Hebbian learning through spike-timing-dependent synaptic plasticity”. *Nature neuroscience* 3(9), pp. 919–926.

[311] Steve B Furber et al., 2014. “The spinnaker project”. *Proceedings of the IEEE* 102(5), pp. 652–665.

[312] Mike Davies et al., 2018. “Loihi: A neuromorphic manycore processor with on-chip learning”. *Ieee Micro* 38(1), pp. 82–99.

[313] Rishidev Chaudhuri et al., 2015. “A large-scale circuit mechanism for hierarchical dynamical processing in the primate cortex”. *Neuron* 88(2), pp. 419–431.

[314] Michele Migliore et al., 2014. “Distributed organization of a brain microcircuit analyzed by three-dimensional modeling: the olfactory bulb”. *Frontiers in computational neuroscience* 8, p. 50.

[315] Alex Fornito, Andrew Zalesky, and Michael Breakspear, 2015. “The connectomics of brain disorders”. *Nature Reviews Neuroscience* 16(3), pp. 159–172.

[316] Hae-Jeong Park and Karl Friston, 2013. “Structural and functional brain networks: from connections to cognition”. *Science* 342(6158), p. 1238411.

[317] JJ Johannes Hjorth, Jeanette Hellgren Kotaleski, and Alexander Kozlov, 2021. “Predicting synaptic connectivity for large-scale microcircuit simulations using Snudda”. *Neuroinformatics* 19(4), pp. 685–701.

[318] Johanna Frost Nylén et al., 2023. “The roles of surround inhibition for the intrinsic function of the striatum, analyzed in silico”. *Proceedings of the National Academy of Sciences* 120(45), e2313058120.

[319] Johanna Frost Nylen et al., 2021. “Dopaminergic and cholinergic modulation of large scale networks in silico using snudda”. *Frontiers in Neural Circuits* 15, p. 748989.

[320] Ilaria Carannante et al., 2024. “The impact of Parkinson’s disease on striatal network connectivity and cortico-striatal drive: an in-silico study”. *Network Neuroscience*, pp. 1–35.

[321] A Paul Alivisatos et al., 2012. “The brain activity map project and the challenge of functional connectomics”. *Neuron* 74(6), pp. 970–974.

[322] Katrin Amunts et al., 2016. “The human brain project: creating a European research infrastructure to decode the human brain”. *Neuron* 92(3), pp. 574–581.

[323] Human Brain Project, 2020. “EBRAINS”, URL: `https://www.ebrains.eu`.

[324] Paula Sanz Leon et al., 2013. “The Virtual Brain: a simulator of primate brain network dynamics”. *Frontiers in neuroinformatics* 7, p. 10.

[325] Christof Koch, 1999. “Biophysics of computation: information processing in single neurons”.

[326] Andreas VM Herz et al., 2006. “Modeling single-neuron dynamics and computations: a balance of detail and abstraction”. *science* 314(5796), pp. 80–85.

[327] Alan L Hodgkin and Andrew F Huxley, 1952. “A quantitative description of membrane current and its application to conduction and excitation in nerve”. *The Journal of physiology* 117(4), p. 500.

[328] Eugene M Izhikevich, 2003. “Simple model of spiking neurons”. *IEEE Transactions on neural networks* 14(6), pp. 1569–1572.

[329] Eugene M Izhikevich, 2004. “Which model to use for cortical spiking neurons?” *IEEE transactions on neural networks* 15(5), pp. 1063–1070.

[330] Anthony N Burkitt, 2006. “A review of the integrate-and-fire neuron model: I. Homogeneous synaptic input”. *Biological cybernetics* 95, pp. 1–19.

[331] Anthony N Burkitt, 2006. “A review of the integrate-and-fire neuron model: II. Inhomogeneous synaptic input and network properties”. *Biological cybernetics* 95, pp. 97–112.

[332] Wulfram Gerstner and Werner M Kistler, 2002. “Spiking neuron models: Single neurons, populations, plasticity”.

[333] Nicolas Brunel, 2000. “Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons”. *Journal of computational neuroscience* 8, pp. 183–208.

[334] Tim P Vogels, Kanaka Rajan, and Larry F Abbott, 2005. “Neural network dynamics”. *Annu. Rev. Neurosci.* 28(1), pp. 357–376.

[335] I. Segev, 1992. “Neural network dynamics”. *Trends in neurosciences* 15(11), pp. 414–421.

[336] Wilfrid Rall, 1962. “Theory of physiological properties of dendrites”. *Annals of the New York Academy of Sciences* 96(4), pp. 1071–1092.

[337] Michael London and Michael Häusser, 2005. “Dendritic computation”. *Annu. Rev. Neurosci.* 28(1), pp. 503–532.

[338] Michael Hausser, Nelson Spruston, and Greg J Stuart, 2000. “Diversity and dynamics of dendritic signaling”. *Science* 290(5492), pp. 739–744.

[339] Michael Häusser and Bartlett Mel, 2003. “Dendrites: bug or feature?” *Current opinion in neurobiology* 13(3), pp. 372–383.

[340] Nelson Spruston, 2008. “Pyramidal neurons: dendritic structure and synaptic integration”. *Nature Reviews Neuroscience* 9(3), pp. 206–221.

[341] Mara Almog and Alon Korngreen, 2016. “Is realistic neuronal modeling realistic?” *Journal of neurophysiology* 116(5), pp. 2180–2209.

[342] Zahra Khodadadi et al., 2024. “Local, calcium-and reward-based synaptic learning rule that enhances dendritic nonlinearities can solve the nonlinear feature binding problem”. *bioRxiv*, pp. 2024–03.

[343] Robert Lindroos and Jeanette Hellgren Kotaleski, 2021. “Predicting complex spikes in striatal projection neurons of the direct pathway following neuromodulation by acetylcholine and dopamine”. *European Journal of Neuroscience* 53(7), pp. 2117–2134.

[344] Edda Klipp and Wolfram Liebermeister, 2006. “Mathematical modeling of intracellular signaling pathways”. *BMC neuroscience* 7(S1), S10.

[345] Upinder S Bhalla, 2014. “Molecular computation in neurons: a modeling perspective”. *Current opinion in neurobiology* 25, pp. 31–37.

[346] BoHung Kim et al., 2013. “Signaling pathways involved in striatal synaptic plasticity are sensitive to temporal pattern and exhibit spatial specificity”. *PLoS Comput Biol* 9(3), e1002953.

[347] Fidel Santamaria et al., 2006. “Anomalous diffusion in Purkinje cell dendrites caused by spines”. *Neuron* 52(4), pp. 635–648.

[348] Iain Hepburn et al., 2012. “STEPS: efficient simulation of stochastic reaction–diffusion models in realistic morphologies”. *BMC systems biology* 6, pp. 1–19.

[349] Eleanore Seibert and Timothy S Tracy, 2014. “Fundamentals of enzyme kinetics”. *Enzyme Kinetics in Drug Metabolism: Fundamentals and Applications*, pp. 9–22.

[350] Boris N Kholodenko, 2006. “Cell-signalling dynamics in time and space”. *Nature reviews Molecular cell biology* 7(3), pp. 165–176.

[351] Bree B Aldridge et al., 2006. “Physicochemical modelling of cell signalling pathways”. *Nature cell biology* 8(11), pp. 1195–1203.

[352] Cédric Yapo et al., 2018. “Switch-like PKA responses in the nucleus of striatal neurons”. *Journal of Cell Science* 131(14), jcs216556.

[353] Anu G Nair et al., 2015. “Sensing positive versus negative reward signals through adenylyl cyclase-coupled GPCRs in direct and indirect pathway striatal medium spiny neurons”. *Journal of Neuroscience* 35(41), pp. 14017–14030.

[354] Olivia Eriksson et al., 2019. “Uncertainty quantification, propagation and characterization by Bayesian analysis combined with global sensitivity analysis applied to dynamical intracellular pathway models”. *Bioinformatics* 35(2), pp. 284–292.

[355] Darren J Wilkinson, 2018. “Stochastic modelling for systems biology”.

[356] Harley H McAdams and Adam Arkin, 1997. “Stochastic mechanisms in gene expression”. *Proceedings of the National Academy of Sciences* 94(3), pp. 814–819.

[357] Daniel T Gillespie, 2007. “Stochastic simulation of chemical kinetics”. *Annu. Rev. Phys. Chem.* 58(1), pp. 35–55.

[358] Daniel T Gillespie, 1977. “Exact stochastic simulation of coupled chemical reactions”. *The journal of physical chemistry* 81(25), pp. 2340–2361.

[359] Paul C Bressloff, 2014. “Stochastic processes in cell biology”.

[360] Claire Ribrault, Ken Sekimoto, and Antoine Triller, 2011. “From the stochasticity of molecular processes to the variability of synaptic transmission”. *Nature Reviews Neuroscience* 12(7), pp. 375–387.

[361] A Aldo Faisal, Luc PJ Selen, and Daniel M Wolpert, 2008. “Noise in the nervous system”. *Nature reviews neuroscience* 9(4), pp. 292–303.

[362] Avigdor Eldar and Michael B Elowitz, 2010. “Functional roles for noise in genetic circuits”. *Nature* 467(7312), pp. 167–173.

[363] Kim T Blackwell et al., 2019. “Molecular mechanisms underlying striatal synaptic plasticity: relevance to chronic alcohol consumption and seeking”. *European Journal of Neuroscience* 49(6), pp. 768–783.

[364] Upinder S Bhalla, 2014. “Multiscale modeling and synaptic plasticity”. *Progress in molecular biology and translational science* 123, pp. 351–386.

[365] Zhilin Qu et al., 2011. “Multi-scale modeling in biology: how to bridge the gaps between scales?” *Progress in biophysics and molecular biology* 107(1), pp. 21–31.

[366] Rishidev Chaudhuri and Ila Fiete, 2016. “Computational principles of memory”. *Nature neuroscience* 19(3), pp. 394–403.

[367] Siri C van Keulen et al., 2023. “Multiscale molecular simulations to investigate adenylyl cyclase-based signaling in the brain”. *Wiley Interdisciplinary Reviews: Computational Molecular Science* 13(1), e1623.

[368] Robert Lindroos et al., 2018. “Basal ganglia neuromodulation over multiple temporal and structural scales—simulations of direct pathway MSNs investigate the fast onset of dopaminergic effects and predict the role of Kv4. 2”. *Frontiers in neural circuits* 12, p. 3.

[369] E Weinan, 2011. “Principles of multiscale modeling”.

[370] Ekaterina Brocke et al., 2017. “Multirate method for co-simulation of electrical-chemical systems in multiscale modeling”. *Journal of computational neuroscience* 42, pp. 245–256.

[371] Gustavo Deco et al., 2015. “Rethinking segregation and integration: contributions of whole-brain modelling”. *Nature Reviews Neuroscience* 16(7), pp. 430–439.

[372] Adam H Marblestone, Greg Wayne, and Konrad P Kording, 2016. “Toward an integration of deep learning and neuroscience”. *Frontiers in computational neuroscience* 10, p. 215943.

[373] Blake A Richards et al., 2019. “A deep learning framework for neuroscience”. *Nature neuroscience* 22(11), pp. 1761–1770.

[374] Mark D Wilkinson et al., 2016. “The FAIR Guiding Principles for scientific data management and stewardship”. *Scientific data* 3(1), pp. 1–9.

[375] Krzysztof J Gorgolewski et al., 2016. “The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments”. *Scientific data* 3(1), pp. 1–9.

[376] Michael Hucka et al., 2003. “The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models”. *Bioinformatics* 19(4), pp. 524–531.

[377] Timo Lubitz et al., 2016. “SBtab: a flexible table format for data exchange in systems biology”. *Bioinformatics* 32(16), pp. 2559–2561.

[378] Michael L Hines et al., 2004. “ModelDB: a database to support computational neuroscience”. *Journal of computational neuroscience* 17, pp. 7–11.

[379] Chen Li et al., 2010. “BioModels Database: An enhanced, curated and annotated resource for published quantitative kinetic models”. *BMC systems biology* 4, pp. 1–14.

[380] Russell A Poldrack and Krzysztof J Gorgolewski, 2017. “OpenfMRI: Open sharing of task fMRI data”. *Neuroimage* 144, pp. 259–261.

[381] Giorgio A Ascoli, Duncan E Donohue, and Maryam Halavi, 2007. “NeuroMorpho. Org: a central resource for neuronal morphologies”. *Journal of Neuroscience* 27(35), pp. 9247–9251.

[382] Ed S Lein et al., 2007. “Genome-wide atlas of gene expression in the adult mouse brain”. *Nature* 445(7124), pp. 168–176.

[383] The MathWorks, Inc., 2024. “Simbiology and SBML - SimBiology Documentation”, URL: `https://www.mathworks.com/help/simbio/gs/simbiology-and-sbml.html`.

[384] David C Van Essen et al., 2013. “The WU-Minn human connectome project: an overview”. *Neuroimage* 80, pp. 62–79.

[385] Padraig Gleeson et al., 2010. “NeuroML: a language for describing data driven models of neurons and networks with a high degree of biological detail”. *PLoS computational biology* 6(6), e1000815.

[386] Padraig Gleeson et al., 2019. “Open source brain: a collaborative resource for visualizing, analyzing, simulating, and developing standardized models of neurons and circuits”. *Neuron* 103(3), pp. 395–411.

[387] Richard C Gerkin et al., 2022. “Databases and Data Repositories in Computational Neuroscience: Overview”. *Encyclopedia of Computational Neuroscience*, pp. 32–35.

[388] Creative Commons, 2001. “Creative Commons: Free and flexible copyright licenses”, Licensing framework for sharing creative works.

[389] Creative Commons, 2002. “Creative Commons Licenses”, Initial release of CC licenses.

[390] The Git Community, 2005. “Git: Fast Version Control System”, Version control software.

[391] GitHub, Inc., 2008. “GitHub: Where the world builds software”, Online repository hosting service.

[392] Mathew Birdsall Abrams et al., 2022. “A standards organization for open and FAIR neuroscience: the international neuroinformatics coordinating facility”. *Neuroinformatics* 20(1), pp. 25–36.

[393] Katherine S Button et al., 2013. “Power failure: why small sample size undermines the reliability of neuroscience”. *Nature reviews neuroscience* 14(5), pp. 365–376.

[394] Olivia Eriksson et al., 2022. “Combining hypothesis-and data-driven neuroscience modeling in FAIR workflows”. *Elife* 11, e69013.

[395] Jeffery L Teeters et al., 2015. “Neurodata without borders: creating a common data format for neurophysiology”. *Neuron* 88(4), pp. 629–634.

[396] The MathWorks Inc., 2024. “MATLAB version: 24.2.0 (R2024b)”, URL: `https://www.mathworks.com`.

[397] João PG Santos, Kadri Pajo, and Jeanette Hellgren Kotaleski, 2024. “Subcellular Workflow Documentation”, URL: `https://subcellular-workflow.readthedocs.io/`, Online; accessed 25-09-2024.

[398] Stefan Hoops et al., 2006. “COPASI—a complex pathway simulator”. *Bioinformatics* 22(24), pp. 3067–3074.

[399] Vito Volterra, 1927. “Variazioni e fluttuazioni del numero d’individui in specie animali conviventi”.

[400] Hiroaki Kitano, 2002. “Systems biology: a brief overview”. *science* 295(5560), pp. 1662–1664.

[401] Warren J Hedley et al., 2001. “A short introduction to CellML”. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 359(1783), pp. 1073–1089.

[402] Warren Hedley et al., 2001. “CellML 1.0 Specification”. *URL: http://www. cellml. org*.

[403] Michael Clerx et al., 2020. “CellML 2.0”. *Journal of Integrative Bioinformatics* 17(2-3), p. 20200021.

[404] The MathWorks, Inc., 2024. “What Is a Model? - SimBiology Documentation”, URL: `https://www.mathworks.com/help/simbio/ug/what-is-a-model.html`.

[405] Timo Lubitz et al., 2016. “SBtab converter”, URL: `https://rumo.biologie.hu-berlin.de/sbtab/default/converter`.

[406] Franklin C McLean, 1938. “Application of the law of chemical equilibrium (law of mass action) to biological problems”. *Physiological Reviews* 18(4), pp. 495–523.

[407] Cornelis Vuik et al., 2023. “Numerical methods for ordinary differential equations”.

[408] C Radhakrishna Rao, 2008. “Linear models and generalizations”.

[409] The MathWorks, Inc., 2024. “fmincon Interior Point Algorithm”, URL: https://www.mathworks.com/help/optim/ug/constrained-nonlinear-optimization-algorithms.html#brnpd5f.

[410] The MathWorks, Inc., 2024. “How Simulated Annealing Works”, URL: https://www.mathworks.com/help/gads/how-simulated-annealing-works.html.

[411] The MathWorks, Inc., 2024. “How Pattern Search Polling Works”, URL: https://www.mathworks.com/help/gads/how-pattern-search-polling-works.html.

[412] The MathWorks, Inc., 2024. “How the Genetic Algorithm Works”, URL: https://www.mathworks.com/help/gads/how-the-genetic-algorithm-works.html.

[413] The MathWorks, Inc., 2024. “Particle Swarm Optimization Algorithm”, URL: https://www.mathworks.com/help/gads/particle-swarm-optimization-algorithm.html.

[414] The MathWorks, Inc., 2024. “Surrogate Optimization Algorithm”, URL: https://www.mathworks.com/help/gads/surrogate-optimization-algorithm.html.

[415] Max D Morris, 1991. “Factorial sampling plans for preliminary computational experiments”. *Technometrics* 33(2), pp. 161–174.

[416] Jérôme Morio, 2011. “Global and local sensitivity analysis methods for a physical system”. *European journal of physics* 32(6), p. 1577.

[417] Ilya M Sobol, 2001. “Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates”. *Mathematics and computers in simulation* 55(1-3), pp. 271–280.

[418] Andrea Saltelli, 2002. “Making best use of model evaluations to compute sensitivity indices”. *Computer physics communications* 145(2), pp. 280–297.

[419] Geir Halnes et al., 2009. “Modelling and sensitivity analysis of the reactions involving receptor, G-protein and effector in vertebrate olfactory receptor neurons”. *Journal of Computational Neuroscience* 27(3), p. 471.

[420] DJ Venzon and SH Moolgavkar, 1988. “A method for computing profile-likelihood-based confidence intervals”. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 37(1), pp. 87–94.

[421] Patrick Royston, 2007. “Profile likelihood for estimation and confidence intervals”. *The Stata Journal* 7(3), pp. 376–387.

[422] Abraham Charnes, Edward L Frome, and Po-Lung Yu, 1976. “The equivalence of generalized least squares and maximum likelihood estimates in the exponential family”. *Journal of the American Statistical Association* 71(353), pp. 169–171.

[423] The MathWorks, Inc., 2024. “sbiosimulate - SimBiology Documentation”, URL: https://www.mathworks.com/help/simbio/ref/sbiosimulate.html.

[424] The MathWorks, Inc., 2024. “ModelObject - SimBiology Documentation”, URL: https://www.mathworks.com/help/simbio/ref/modelobject.html.

[425] The MathWorks, Inc., 2024. “ConfigSetObject - SimBiology Documentation”, URL: https://www.mathworks.com/help/simbio/ref/configsetobject.html.

[426] The MathWorks, Inc., 2024. “SimData - SimBiology Documentation”, URL: https://www.mathworks.com/help/simbio/ref/simdata.html.

[427] The MathWorks, Inc., 2024. “MaximumWallClock - SimBiology Documentation”, URL: https://www.mathworks.com/help/simbio/ref/maximumwallclock.html.

[428] The MathWorks, Inc., 2024. “TimeUnits - SimBiology Documentation”, URL: https://www.mathworks.com/help/simbio/ref/timeunits.html.

[429] The MathWorks, Inc., 2024. “StopTime - SimBiology Documentation”, URL: https://www.mathworks.com/help/simbio/ref/stoptime.html.

[430] The MathWorks, Inc., 2024. “DimensionalAnalysis - SimBiology Documentation”, URL: https://www.mathworks.com/help/simbio/ref/dimensionalanalysis.html.

[431] The MathWorks, Inc., 2024. “UnitConversion - SimBiology Documentation”, URL: https://www.mathworks.com/help/simbio/ref/unitconversion.html.

[432] The MathWorks, Inc., 2024. “OutputTimes - SimBiology Documentation”, URL: https://www.mathworks.com/help/simbio/ref/outputtimes.html.

[433] The MathWorks, Inc., 2024. “MaxStep - SimBiology Documentation”, URL: https://www.mathworks.com/help/simbio/ref/maxstep.html.

[434] The MathWorks, Inc., 2024. “RelativeTolerance - SimBiology Documentation”, URL: https://www.mathworks.com/help/simbio/ref/relativetolerance.html.

[435] The MathWorks, Inc., 2024. “AbsoluteTolerance - SimBiology Documentation”, URL: https://www.mathworks.com/help/simbio/ref/absolutetolerance.html.

[436] The MathWorks, Inc., 2024. “AbsoluteToleranceScaling - SimBiology Documentation”, URL: https://www.mathworks.com/help/simbio/ref/absolutetolerancescaling.html.

[437] The MathWorks, Inc., 2024. “AbsoluteToleranceStepSize - SimBiology Documentation”, URL: https://www.mathworks.com/help/simbio/ref/absolutetolerancestepsize.html.

[438] The MathWorks, Inc., 2024. “Selecting Absolute Tolerance and Relative Tolerance for Simulation - SimBiology Documentation”, URL: https://www.mathworks.com/help/simbio/ug/selecting-absolute-tolerance-and-relative-tolerance-for-simulation.html.

[439] The MathWorks, Inc., 2024. “ode15s - MATLAB Documentation”, URL: https://www.mathworks.com/help/matlab/ref/ode15s.html.

[440] Jill B Jensen et al., 2009. “Fluorescence changes reveal kinetic steps of muscarinic receptor–mediated modulation of phosphoinositides and Kv7. 2/7.3 K+ channels”. *Journal of General Physiology* 133(4), pp. 347–359.

[441] The MathWorks, Inc., 2024. “First-order optimality measure - Optimization Toolbox Documentation”, URL: https://www.mathworks.com/help/optim/ug/first-order-optimality-measure.html.

[442] The MathWorks, Inc., 2024. “fmincon - Optimization Toolbox Documentation”, URL: https://www.mathworks.com/help/optim/ug/fmincon.html.

[443] The MathWorks, Inc., 2024. “simulannealbnd - Global Optimization Toolbox Documentation”, URL: https://www.mathworks.com/help/gads/simulannealbnd.html.

[444] The MathWorks, Inc., 2024. “patternsearch - Global Optimization Toolbox Documentation”, URL: https://www.mathworks.com/help/gads/patternsearch.html.

[445] The MathWorks, Inc., 2024. “ga - Global Optimization Toolbox Documentation”, URL: https://www.mathworks.com/help/gads/ga.html.

[446] The MathWorks, Inc., 2024. “Genetic Algorithm Options - Global Optimization Toolbox Documentation”, URL: https://www.mathworks.com/help/gads/genetic-algorithm-options.html.

[447] The MathWorks, Inc., 2024. “particleswarm - Global Optimization Toolbox Documentation”, URL: https://www.mathworks.com/help/gads/particleswarm.html.

[448] The MathWorks, Inc., 2024. “surrogateopt - Global Optimization Toolbox Documentation”, URL: https://www.mathworks.com/help/gads/surrogateopt.html.

[449] The MathWorks, Inc., 2024. “boxchart - MATLAB Documentation”, URL: https://www.mathworks.com/help/matlab/ref/boxchart.html.

[450] Paul M Stemmer and Claude B Klee, 1994. “Dual calcium ion regulation of calcineurin by calmodulin and calcineurin B”. *Biochemistry* 33(22), pp. 6859–6866.

[451] Lu Li, Melanie I Stefan, and Nicolas Le Novère, 2012. “Calcium input frequency, duration and amplitude differentially modulate the relative activation of calcineurin and CaMKII”.

[452] Björn H Falkenburger, Jill B Jensen, and Bertil Hille, 2010. “Kinetics of M1 muscarinic receptor and G protein signaling to phospholipase C in living cells”. *Journal of General Physiology* 135(2), pp. 81–97.

[453] Björn H Falkenburger, Jill B Jensen, and Bertil Hille, 2010. “Kinetics of PIP2 metabolism and KCNQ2/3 channel regulation studied with a voltage-sensitive phosphatase in living cells”. *Journal of general physiology* 135(2), pp. 99–114.

[454] Purves, 2017. “Neuroscience 6th Edition”. Chap. 7, pp. 149–154.

[455] Purves, 2017. “Neuroscience 6th Edition”. Chap. 7, p. 153.

[456] Yulia V Shulga et al., 2010. “Molecular species of phosphatidylinositol-cycle intermediates in the endoplasmic reticulum and plasma membrane”. *Biochemistry* 49(2), pp. 312–317.

[457] Björn H Falkenburger, Jill B Jensen, and Bertil Hille, 2010. “Kinetics of M1 muscarinic receptor and G protein signaling to phospholipase C in living cells”. *Journal of General Physiology* 135(2), pp. 81–97.

A SBtab

Reaction Fields	
Field Name	**Description**
!ID	Unique identifier for the reaction within the SBtab
!Name	Name of the reaction
!KineticLaw	Mathematical expression representing the reaction kinetic law
!IsReversible	Boolean value (True/False) indicating reaction reversibility
!Location	Cellular compartment where the reaction occurs
!ReactionFormula	Chemical equation representing the reaction

Table A.1: Fields utilized in the 'Reaction' sheet of the SBtabs. Each field provides specific information about the biochemical reactions modelled in the system, enabling a comprehensive representation of reaction properties and dynamics.

Parameter Fields	
Field Name	**Description**
!ID	Unique identifier for the parameter within the SBtab
!Name	Name of the parameter
!Unit	Measurement unit for the parameter
!Value:log2	Parameter value in base 2 logarithm
!Value:linspace	Parameter value in linear space
!Value:log10	Parameter value in base 10 logarithm
!Location	Cellular compartment to which the parameter is assigned
!Comment	Full descriptive name or additional information about the parameter

Table A.2: Fields utilized in the 'Parameter' sheet of the SBtabs. Each field provides specific information about model parameters, allowing for comprehensive representation of parameter properties, values, and associated metadata.

Output Fields	
Field Name	**Description**
!ID	Unique identifier for the output within the SBtab
!Name	Name of the output for model incorporation
!Comment	Full descriptive name of the output for clarity
!ErrorName	Identifier for the associated error or uncertainty
!Unit	Measurement unit for the output
!Location	Cellular compartment to which the output is assigned
!Formula	Mathematical expression defining the output in terms of model species and parameters

Table A.3: Fields utilized in the 'Output' sheet of the SBtabs. Each field provides specific information about model outputs, enabling a comprehensive representation of measurable quantities, their definitions, and associated metadata.

Experiments Fields	
Field Name	**Description**
!ID	Unique identifier for the experiment within the SBtab
!Name	Name of the experiment
!Type	Classification or category of the experiment
>Sx	Initial concentrations of compounds for this experiment (references Compound sheet)
!Normalize	Lists IDs of Outputs to be normalized in the experiment
!ReferenceDOI	Digital Object Identifier (DOI) for the experimental data source

Table A.4: Fields utilized in the 'Experiments' sheet of the SBtabs. Each field provides specific information about individual experiments, including their setup, measured outputs, and associated metadata, enabling comprehensive representation of experimental conditions and data sources.

Individual Experiment Fields (E#)	
Field Name	**Description**
!TimePoint	Unique identifier for each data point within the experiment
!Time	Time at which the measurement was taken (units specified in simulation settings)
>Yx	Measured value of output Y at time x (references Output sheet)
SD_Yx	Standard deviation or error of output Y at time x (references Output sheet)

Table A.5: Fields utilized in the individual experiments sheets of the SBtabs. Each E# sheet corresponds to a specific experiment, containing time series data for measured outputs and their associated uncertainties. This structure enables detailed representation of experimental results for model validation and parameter estimation.

Individual Experiment Input Fields (E#I)	
Field Name	**Description**
!Input_Time_Sx	Time point for input application (units specified in simulation settings)
>Sx	Concentration or value of input compound Sx at the specified time (references Compound sheet)
!Sim_Time	Total duration for which the model should be simulated to match experimental conditions

Table A.6: Fields utilized in the individual experiments input sheets of the SBtabs. Each E#I sheet corresponds to a specific experiment, detailing the timing and magnitude of input perturbations applied during the simulation. This structure enables precise representation of experimental interventions for accurate model simulation and analysis.

B Example models

!ID	!KineticLaw	!ReactionFormula
R0	kf_0*aR*L - kr_0*aRL	aR + L <=> aRL
R1	kf_1*iR - kr_1*aR	iR <=> aR
R2	kf_2*iRL - kr_2*aRL	iRL <=> aRL
R3	kf_3*iR*L - kr_3*iRL	iR + L <=> iRL
R4	kf_4*iE*aR - kr_4*aE*aR	iE + aR <=> aE + aR
R5	kf_5*iE*aRL - kr_5*aE*aRL	iE + aRL <=> aE + aRL
R6	kf_6*aE	aE <=> iE

Table B.1: LiRMo and LiREM kinetic laws and reactions as present in the SBtab (some columns not shown). Reactions $R_{0–3}$ are present in both LiRMo and LiREM, while $R_{4–6}$ are exclusive to LiREM. The table shows reaction IDs, kinetic laws, and reaction formulas, illustrating the increasing complexity from LiRMo to LiREM.

```
1 disp(modelobj.reactions)
```

```
Index:    Reaction:
1         aR + L <-> aRL
2         iR <-> aR
3         iRL <-> aRL
4         iR + L <-> iRL
```

```
1 for n = 1:size(modelobj.reactions,1)
2 disp(modelobj.reactions(n,1).reactionrate)
3 end
```

```
kf_0*aR*L - kr_0*aR_L
kf_1*iR - kr_1*aR
kf_2*iRL - kr_2*aRL
kf_3*iR*L - kr_3*iRL
```

Figure B.1: LiRMo reactions and their kinetic rates as displayed in MATLAB after model import. The upper output shows the symbolic representation of reactions, while the lower output details the kinetic rate equations for each reaction. This demonstrates the successful translation of SBtab information into MATLAB's model object format.

!ID	!Name	!Unit	!Value:log10
K0	kf_0	1/(nanomolarity*second)	3.835
K1	kr_0	1/second	0.700
K2	kf_1	1/second	-0.714
K3	kr_1	1/second	1.436
K4	kf_2	1/second	3.584
K5	kr_2	1/second	-1.068
K6	kf_3	1/(nanomolarity*second)	3.664
K7	kr_3	1/second	7.331
K8	kf_4	1/(nanomolarity*second)	1.957
K9	kr_4	1/(nanomolarity*second)	-1.045
K10	kf_5	1/(nanomolarity*second)	4.000
K11	kr_5	1/(nanomolarity*second)	-3.967
K12	kf_6	1/second	-2.454

Table B.2: LiRMo and LiREM parameters as present in the SBtab (some columns not shown). The table shows parameter IDs, names, units, and log10 values. Parameters K_{0-7} are present in both LiRMo and LiREM, while K_{8-12} are exclusive to LiREM, illustrating the increased complexity of the latter model. Units are provided to ensure dimensional consistency in model simulations.

```
1  disp(modelobj.Parameters)
```

```
Index:    Name:    Value:        Units:
1         kf_0     6832.93       1/(nanomolarity*second)
2         kr_0     5.00951       1/second
3         kf_1     0.192976      1/second
4         kr_1     27.2655       1/second
5         kf_2     3837.29       1/second
6         kr_2     0.085493      1/second
7         kf_3     4610.52       1/(nanomolarity*second)
8         kr_3     2.1436e+07    1/second
```

Figure B.2: Parameters of the LiRMo model as displayed in MATLAB after import from SBtab. The output shows the index, name, value, and units for each parameter. Note the conversion from log10 values in the SBtab to actual values in MATLAB, and the preservation of units. This demonstrates the successful translation and interpretation of parameter information during the import process.

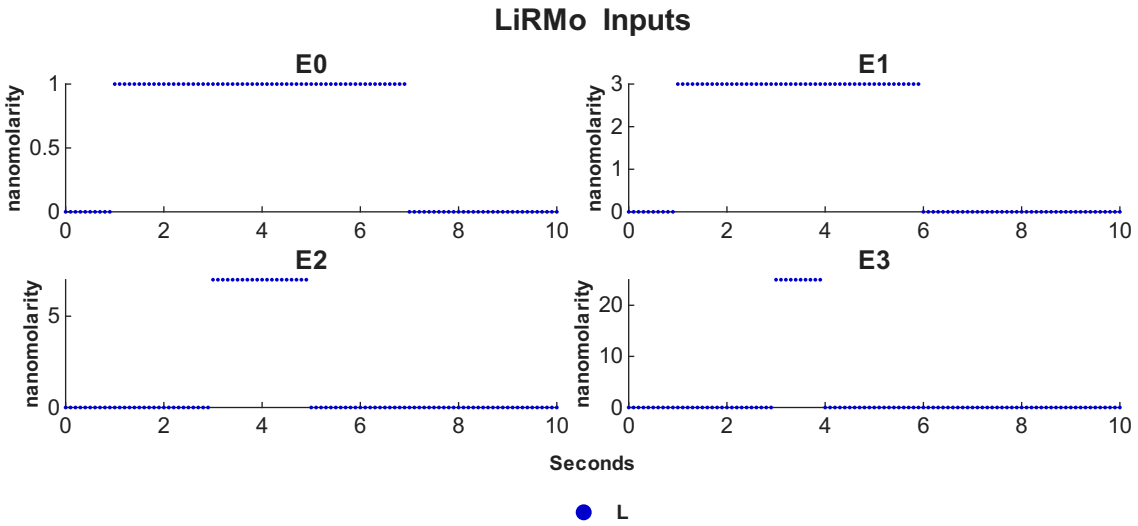


Figure B.3: LiRMo inputs E_{0-3}. Experimental input profiles for LiRMo from E_{0-3}, displaying the administration of Ligand (L) at different concentrations and times.

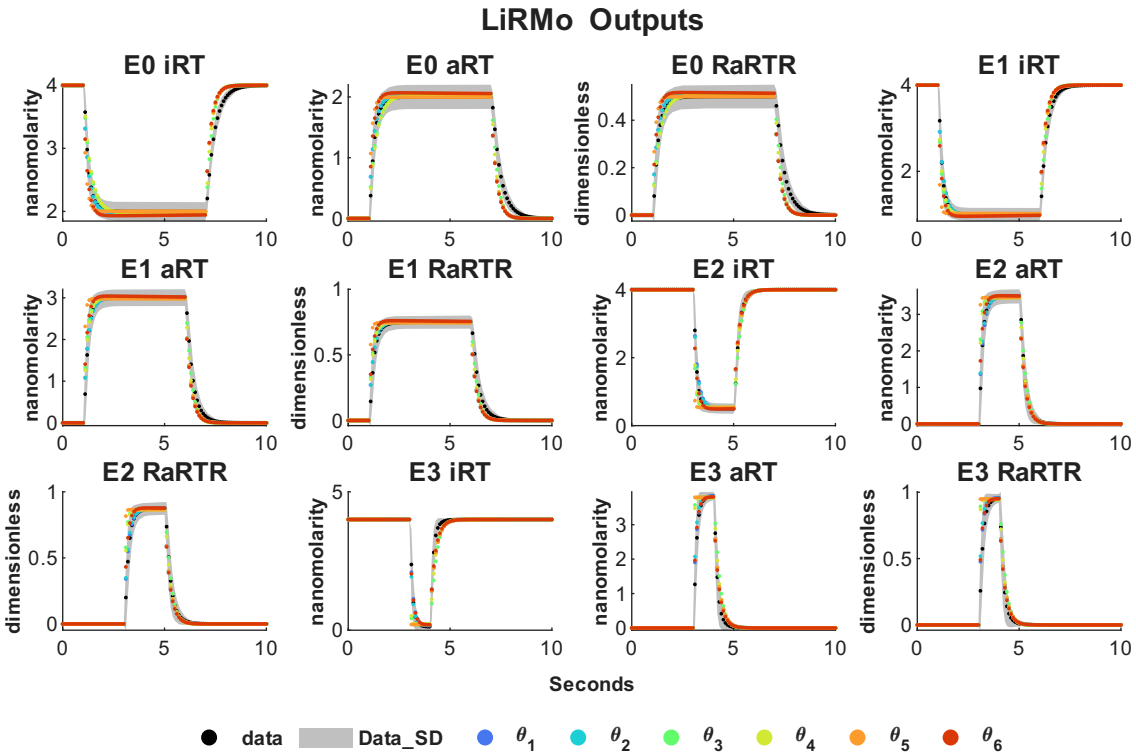


Figure B.4: LiRMo outputs E_{0-3}. The figures display the temporal dynamics of key molecular components: inactive receptor total amount (iRT), active receptor total amount (aRT), ratio of active receptor to total receptor amount (RaRTR), inactive receptor total amount (iRT). Each plot contrasts multiple parameter sets (θ_{0-6}) against the generated data and its standard deviation (Data_SD), reflecting the model's ability to capture the biochemical behaviour in E_{0-3}.

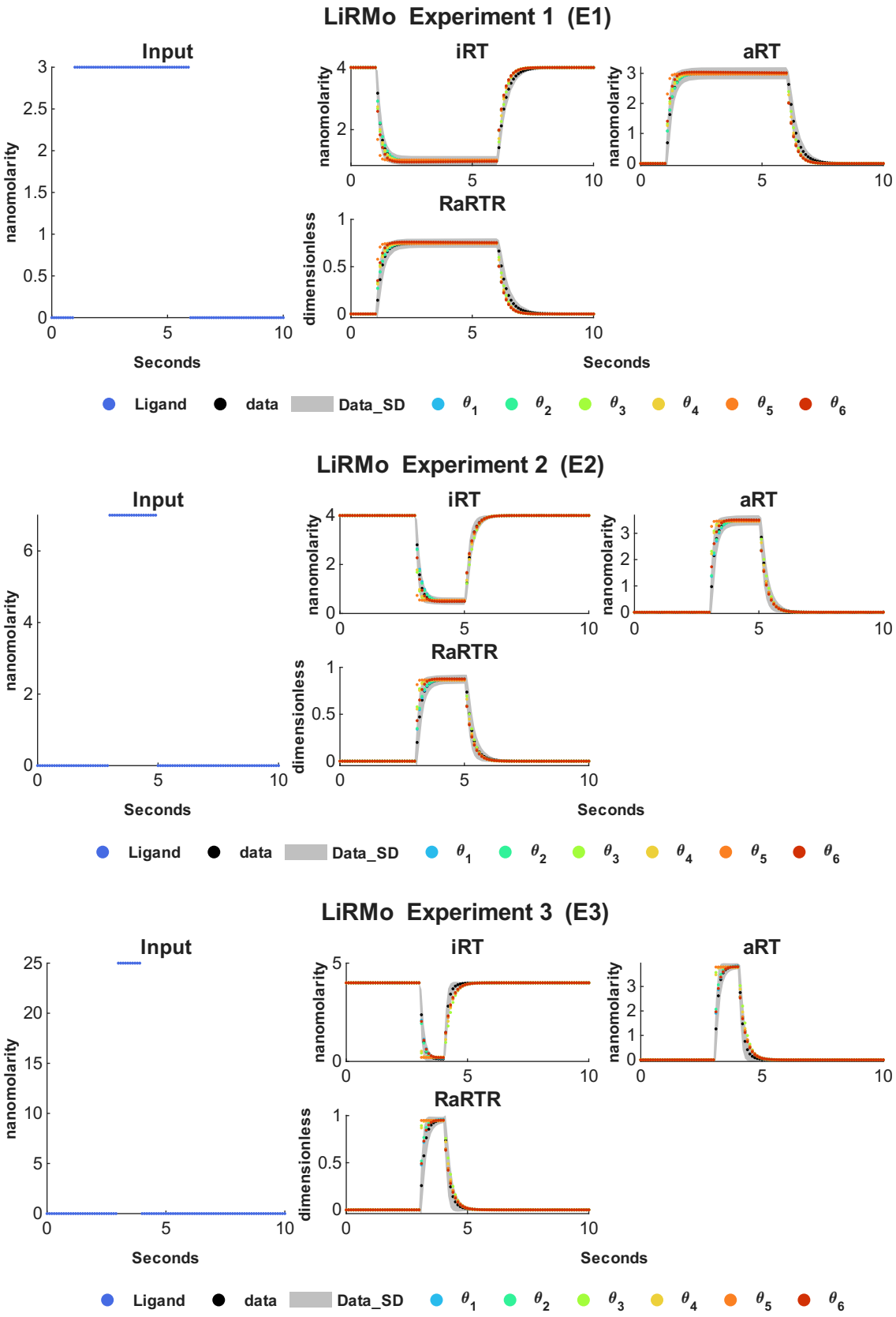


Figure B.5: LiRMo E_{1-3} diagnostics plot. Here the input, output data, and output of LiRMo E_0 given various θ are shown. θ_{1-6} are the results of optimizing the model to this data as shown in figure 4.10, each corresponding to an optimization algorithm as follows: 1 – Genetic algorithm, 2 – fmincon, 3 – Simulated annealing, 4 – Pattern search, 5 – Particle swarm, and 6 – Surrogate optimization. Output data are the same as in figure 4.7.

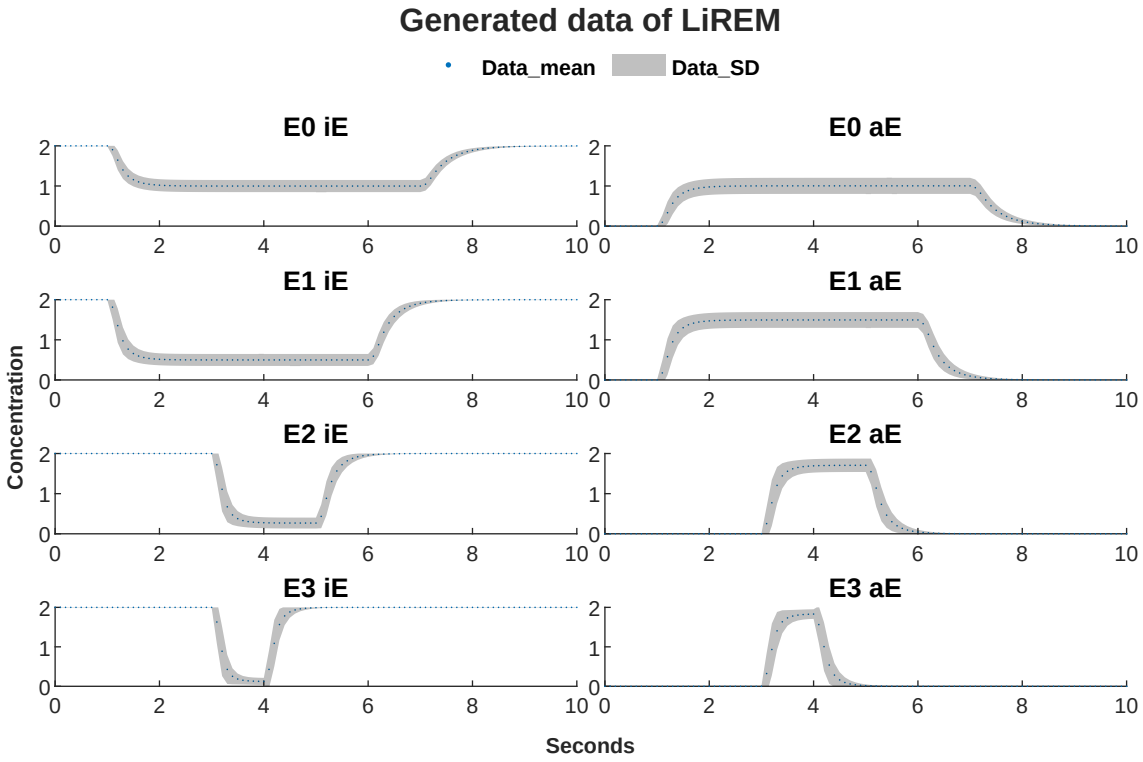


Figure B.6: LiREM dummy data. Experiments 0 through 3 are denoted by $E_{0–3}$. Each experiment has two outputs, iE and aE. The blue points and the grey area represent the mean and the area encapsulated within one standard deviation of the mean of the generated data, respectively. Together, this data enables the evaluation of the model's performance under varying experimental conditions.

$Name$	$maxA$	$\overline{A}$	$A\sigma$	$start$	$\overline{start_d}$	$start_d\sigma$	$\overline{start_s}$	$start_s\sigma$	end	$\overline{end_d}$	$end_d\sigma$	$\overline{end_s}$	$end_s\sigma$	E_t	N
E0 iE	-2	-1.000	0.15	1.00	0.10	0.08	0.20	0.08	7.00	0.10	0.08	0.40	0.08	10	100
E0 aE	2	1.000	0.20	1.00	0.10	0.10	0.20	0.10	7.00	0.10	0.10	0.40	0.10	10	100
E1 iE	-2	-1.500	0.15	1.00	0.10	0.08	0.15	0.08	6.00	0.10	0.08	0.30	0.08	10	100
E1 aE	2	1.500	0.20	1.00	0.10	0.10	0.15	0.10	6.00	0.10	0.10	0.30	0.10	10	100
E2 iE	-2	-1.750	0.15	3.00	0.10	0.08	0.10	0.08	5.00	0.10	0.08	0.20	0.08	10	100
E2 aE	2	1.750	0.20	3.00	0.10	0.10	0.10	0.10	5.00	0.10	0.10	0.20	0.10	10	100
E3 iE	-2	-2.000	0.15	3.00	0.10	0.08	0.05	0.08	4.00	0.10	0.08	0.10	0.08	10	100
E3 aE	2	2	0.20	3.00	0.10	0.10	0.05	0.10	4	0.10	0.10	0.10	0.10	10	100

Table B.3: LiREM dummy data generating parameters. Parameters used for 4 experiments (E_{0-3}) for the two outputs, iE and aE. These parameters determine the characteristics of the generated data, including the maximum amplitude change ($maxA$), mean amplitude change ($\overline{A}$), standard deviation of the amplitude change($A\sigma$), default start and end times($start$,end), mean delays at the start and end times($\overline{start_d}$,$\overline{end_d}$) and their corresponding standard deviations($start_d\sigma$,$end_d\sigma$). Additionally, the mean of the shape parameters of the start and end times($\overline{start_s}$,$\overline{end_s}$) and their standard deviations are also listed($start_s\sigma$,$end_s\sigma$). The table also specifies the total duration of each experiment(E_t) and the number of data points collected (N).

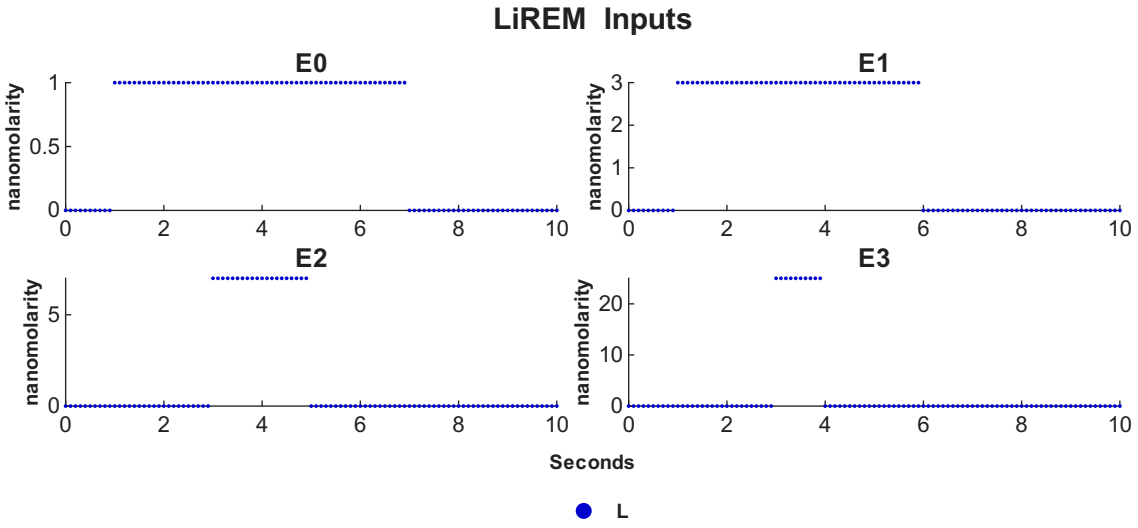


Figure B.7: LiREM inputs $E_{0–3}$. Experimental input profiles for LiREM from $E_{0–3}$, displaying the administration of Ligand (L) at different concentrations and times

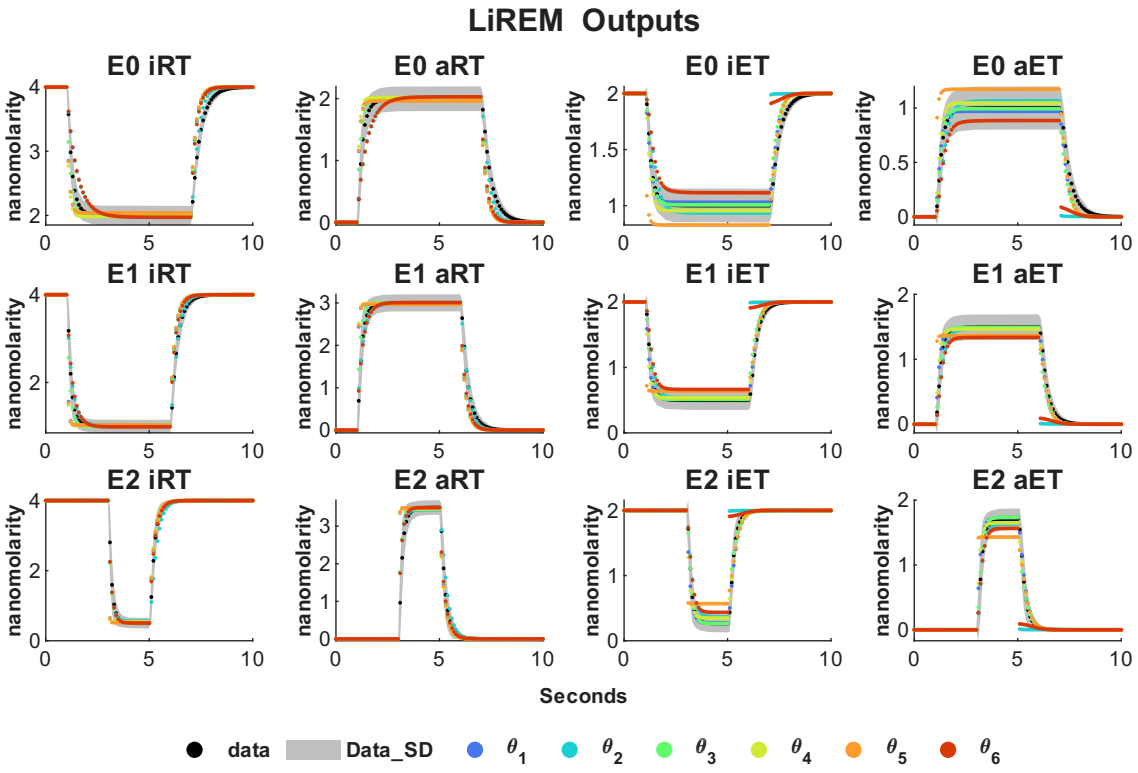


Figure B.8: LiREM outputs $E_{0–2}$. The figures display the temporal dynamics of key molecular components: inactive receptor total amount (iRT), active receptor total amount (aRT), inactive effector total amount (iET), active effector total amount (aET). Each plot contrasts multiple parameter sets ($\theta_{1–6}$) against the generated data and its standard deviation (Data_SD), reflecting the model's ability to capture the biochemical behaviour in $E_{0–2}$.

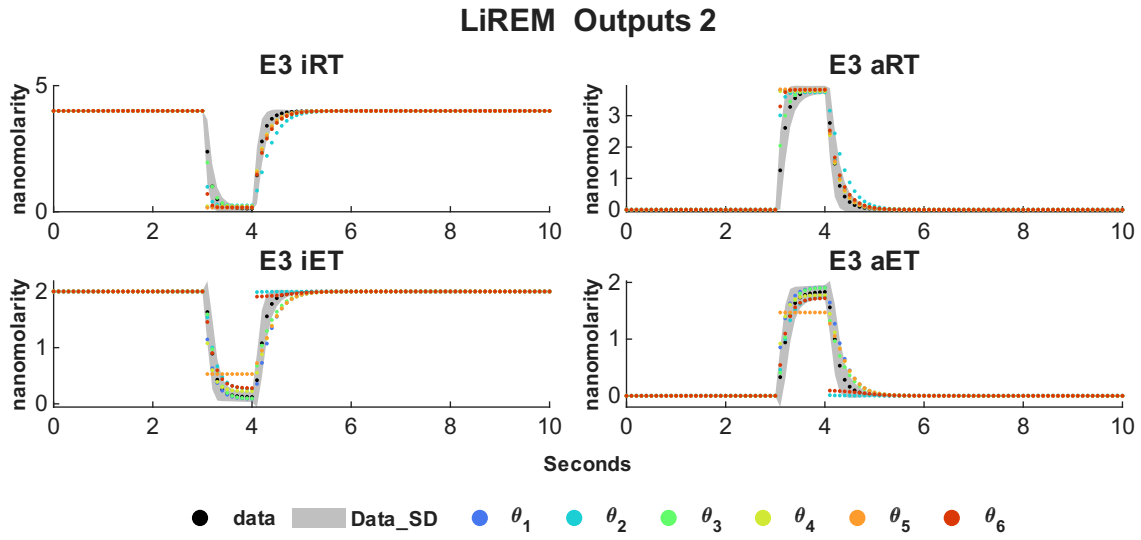


Figure B.9: LiREM outputs E_3. The figures display the temporal dynamics of key molecular components: inactive receptor total amount (iRT), active receptor total amount (aRT), inactive effector total amount (iET), active effector total amount (aET). Each plot contrasts multiple parameter sets (θ_{1-6}) against the generated data and its standard deviation (Data_SD), reflecting the model's ability to capture the biochemical behaviour in E_3.

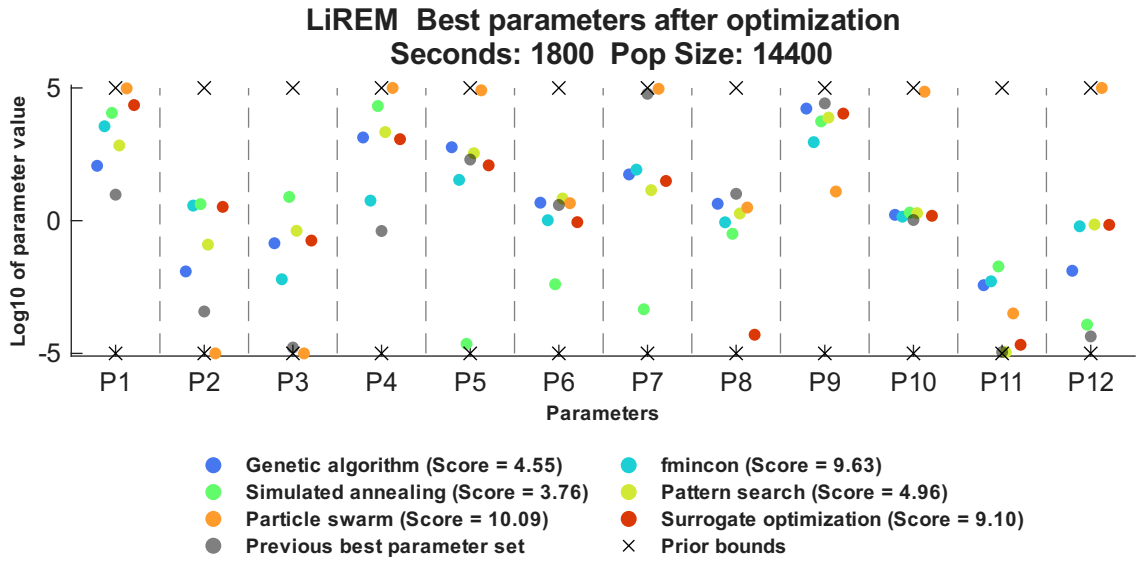


Figure B.10: LiREM optimized θ. This scatter plot presents the best θs obtained from each optimization algorithm, expressed as the log-transformed values for parameters P_{1-12}. Each color represents a different algorithm, with their respective optimization scores indicated in the legend. Black crosses denote the prior parameter bounds. The plot reveals the convergence of parameter values toward optimal solutions within the defined search space. The computational time was fixed at 7200 seconds with a population size of 3600, illustrating the effectiveness of each algorithm within these constraints.

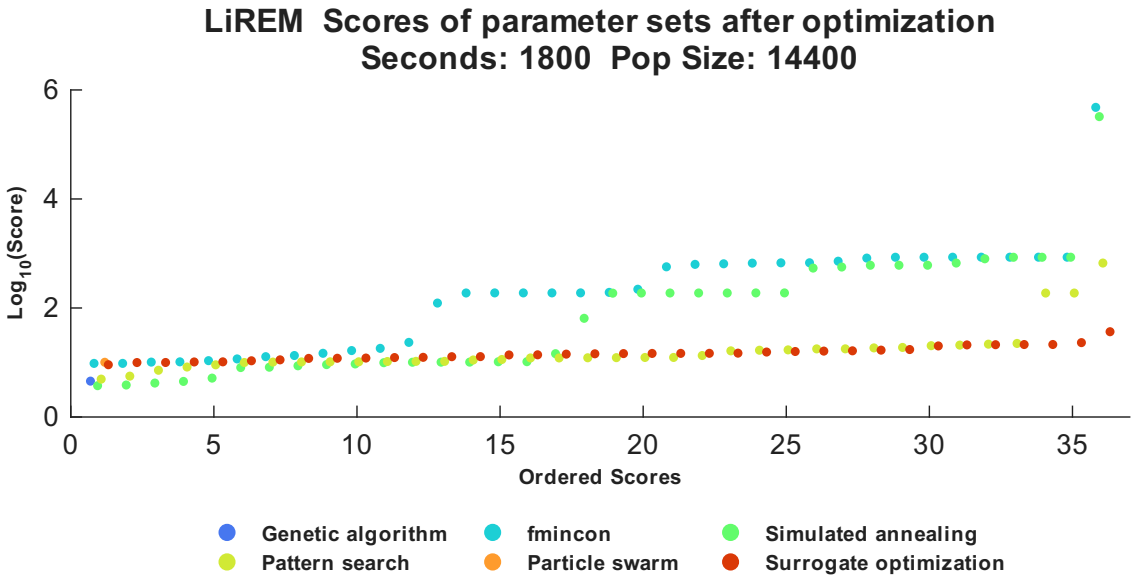


Figure B.11: LiREM optimized θ comparative scores. This plot showcases the log-scaled scores for optimized θs derived from various algorithms for the LiREM. Each dot corresponds to the score of a θ, arranged on the horizontal axis by increasing score. The vertical axis denotes the logarithm of the score, reflecting the fit quality. All algorithms were assessed under identical computational constraints, with a fixed runtime of 1800 seconds (using an Intel i9-10980XE) and a population size of 14400 (genetic algorithm and particle swarm), to facilitate an equitable comparison of their performance in optimizing the LiREM parameters.

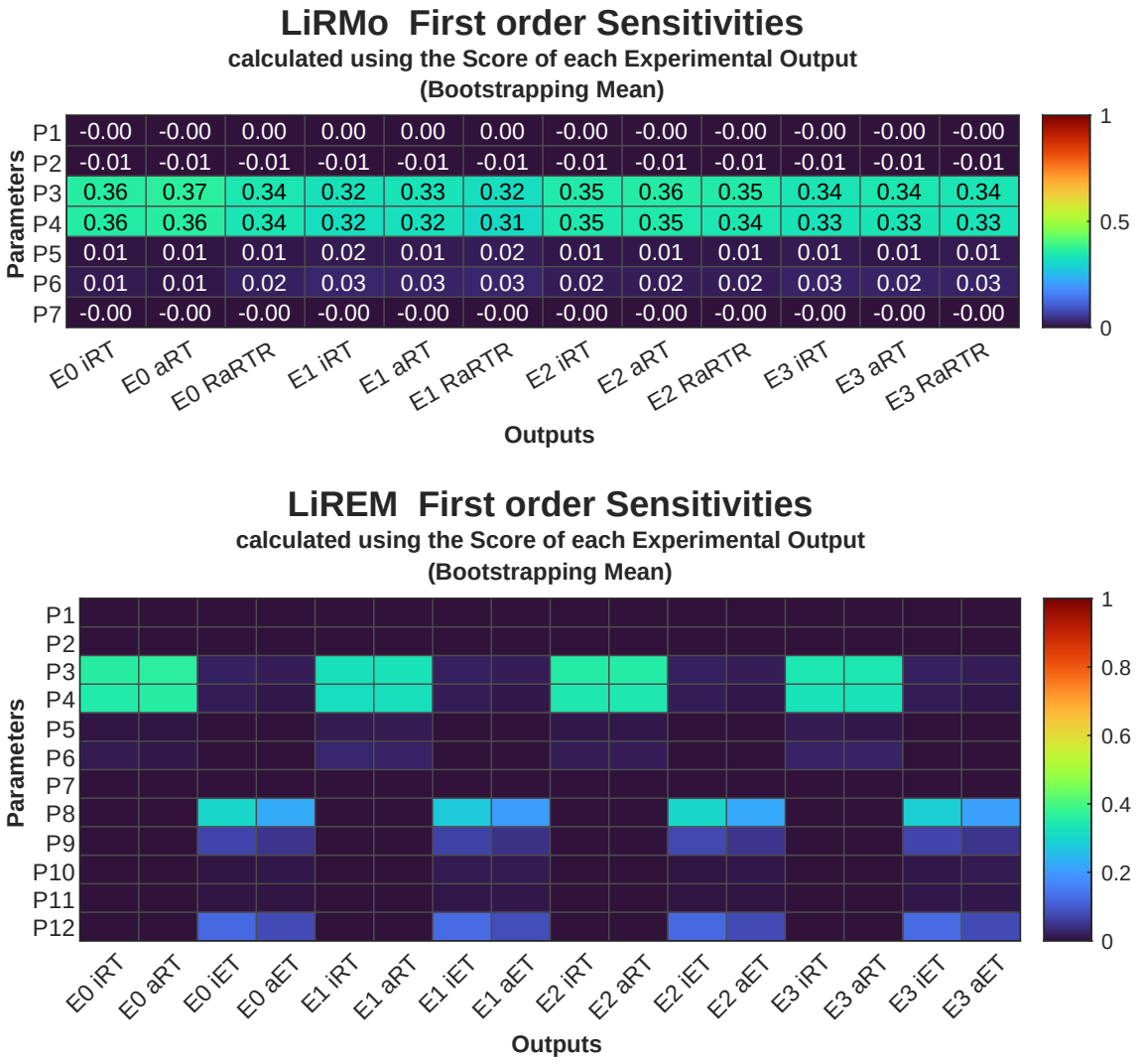


Figure B.12: LiRMo and LiREM GSA S_{Fi} for each Eo. Sensitivity indices are calculated for all model parameters ($P_{1–7}$ for LiRMo and $P_{1–12}$ for LiREM) relative to the score per experimental output (s_{Eo}) of all model outputs (iRT, aRT, and RaTR for LiRMo and iRT, aRT, iE, and aE for LiREM) from $E_{0–3}$ as shown in section 3.6.2.1. A sample size of 10000 θ sampled as in section 3.6.2.2 was used. Higher values indicate greater sensitivity of a specific parameter given a certain experimental output.

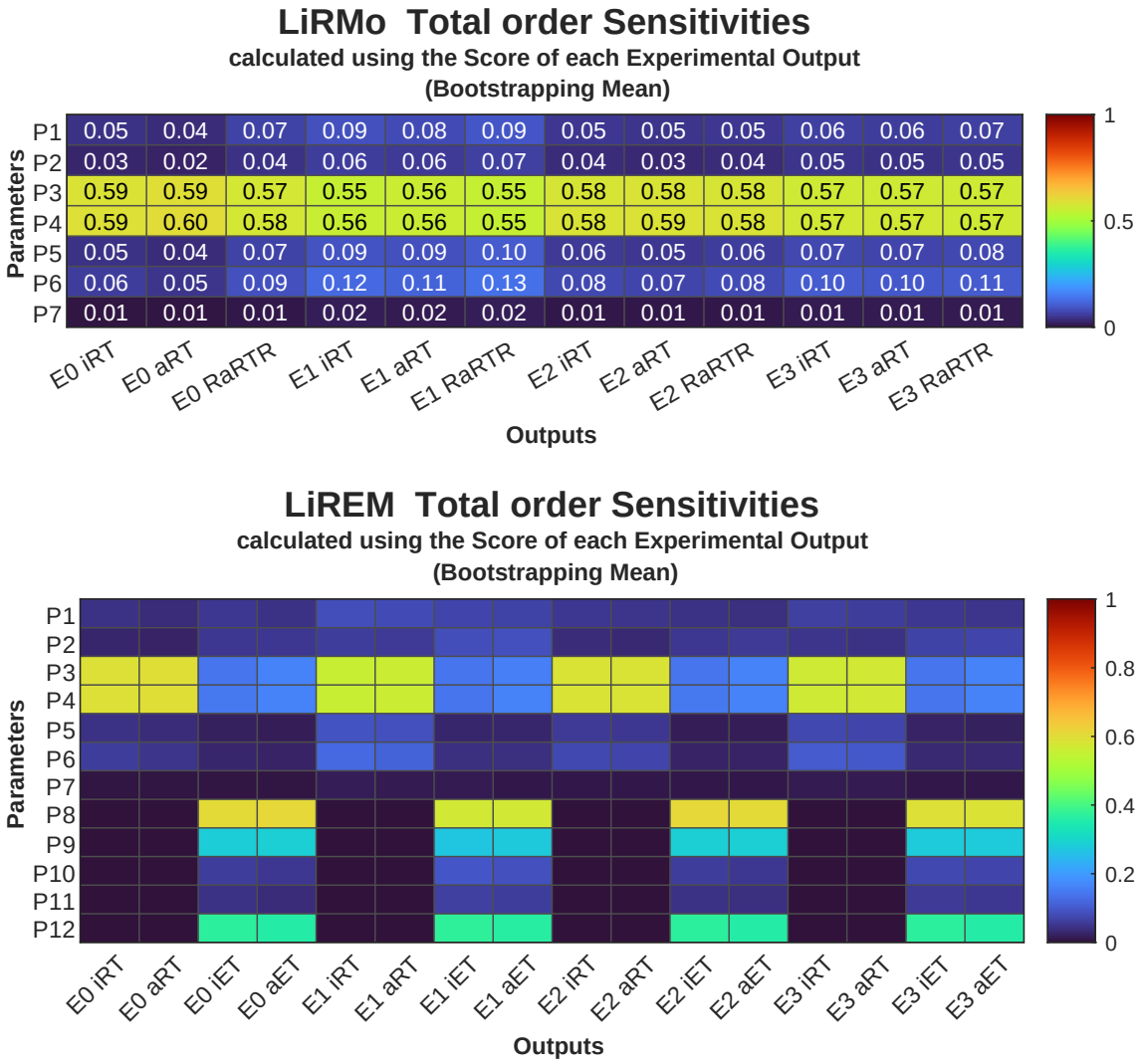


Figure B.13: LiRMo and LiREM GSA S_{Ti} for each Eo. Sensitivity indices are calculated for all model parameters ($P_{1–7}$ for LiRMo and $P_{1–12}$ for LiREM) relative to the s_{Eo} of all model outputs (iRT, aRT, and RaRTR) from $E_{0–3}$ as shown in section 3.6.2.1. A sample size of 10000 θ sampled as in section 3.6.2.2 was used. Higher values indicate greater sensitivity of a specific parameter given a certain experimental output.

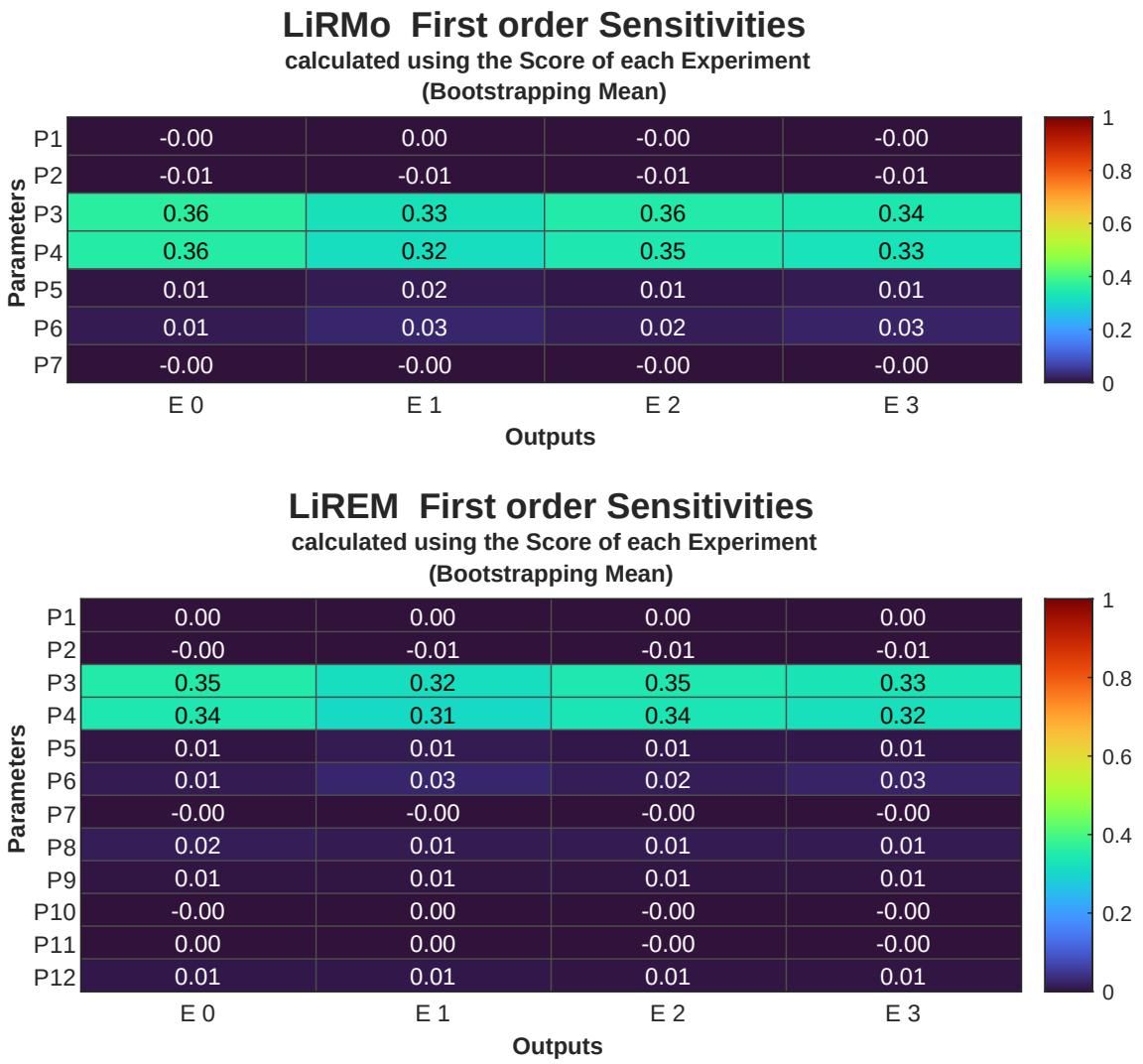


Figure B.14: LiRMo and LiREM GSA S_{Fi} for each E. Sensitivity indices are calculated for all model parameters (P_{1-7} for LiRMo and P_{1-12} for LiREM) relative to the score per experiment (s_E) from E_{0-3} as shown in section 3.6.2.1. A sample size of 10000 θ sampled as in section 3.6.2.2 was used. Higher values indicate greater sensitivity of a specific parameter given a certain experimental score.

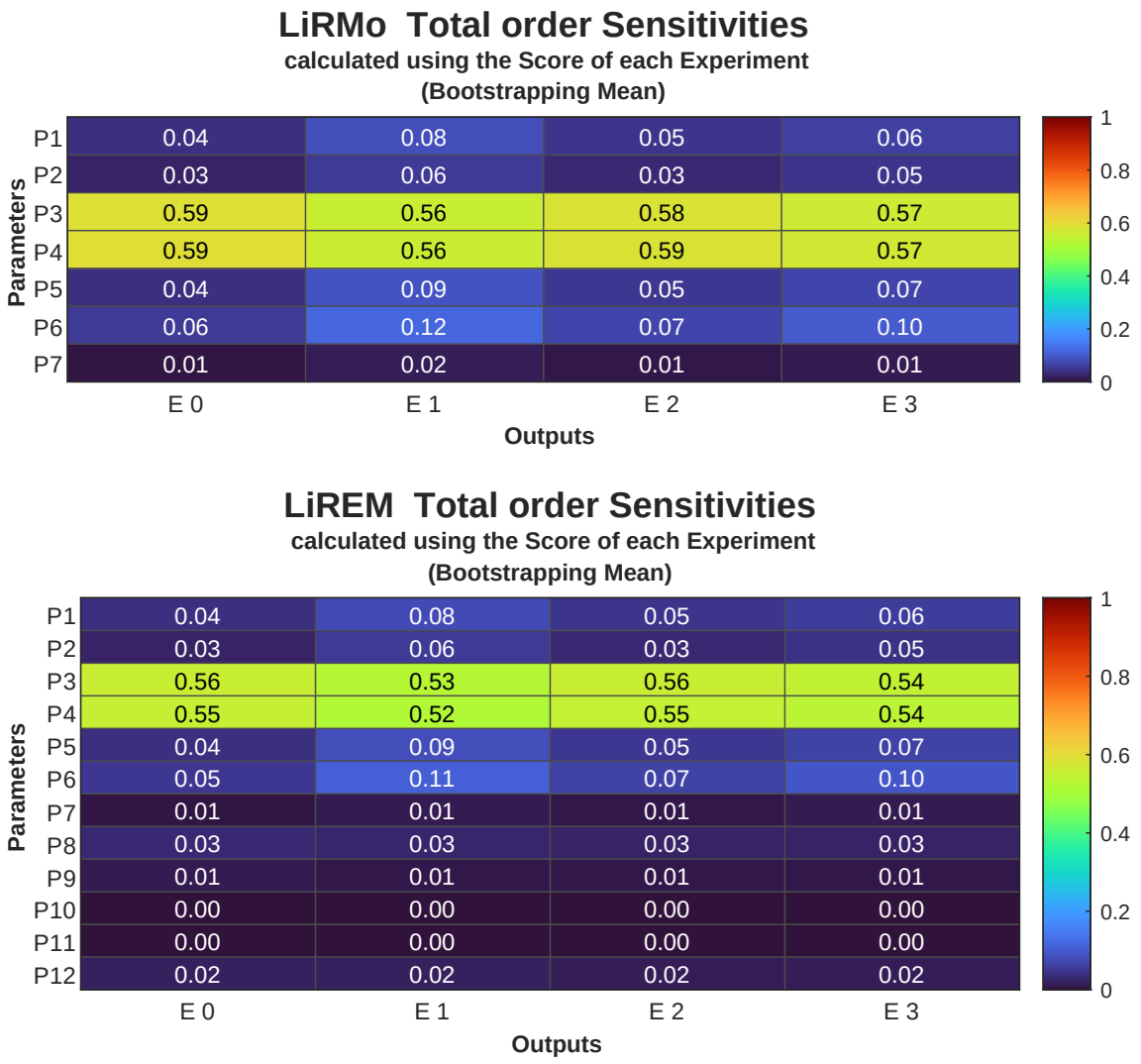


Figure B.15: LiRMo and LiREM GSA S_{Ti} for each E. Sensitivity indices are calculated for all model parameters (P_{1-7} for LiRMo and P_{1-12} for LiREM) relative to the s_E from E_{0-3} as shown in section 3.6.2.1. A sample size of 10000 θ sampled as in section 3.6.2.2 was used. Higher values indicate greater sensitivity of a specific parameter given a certain experimental score.

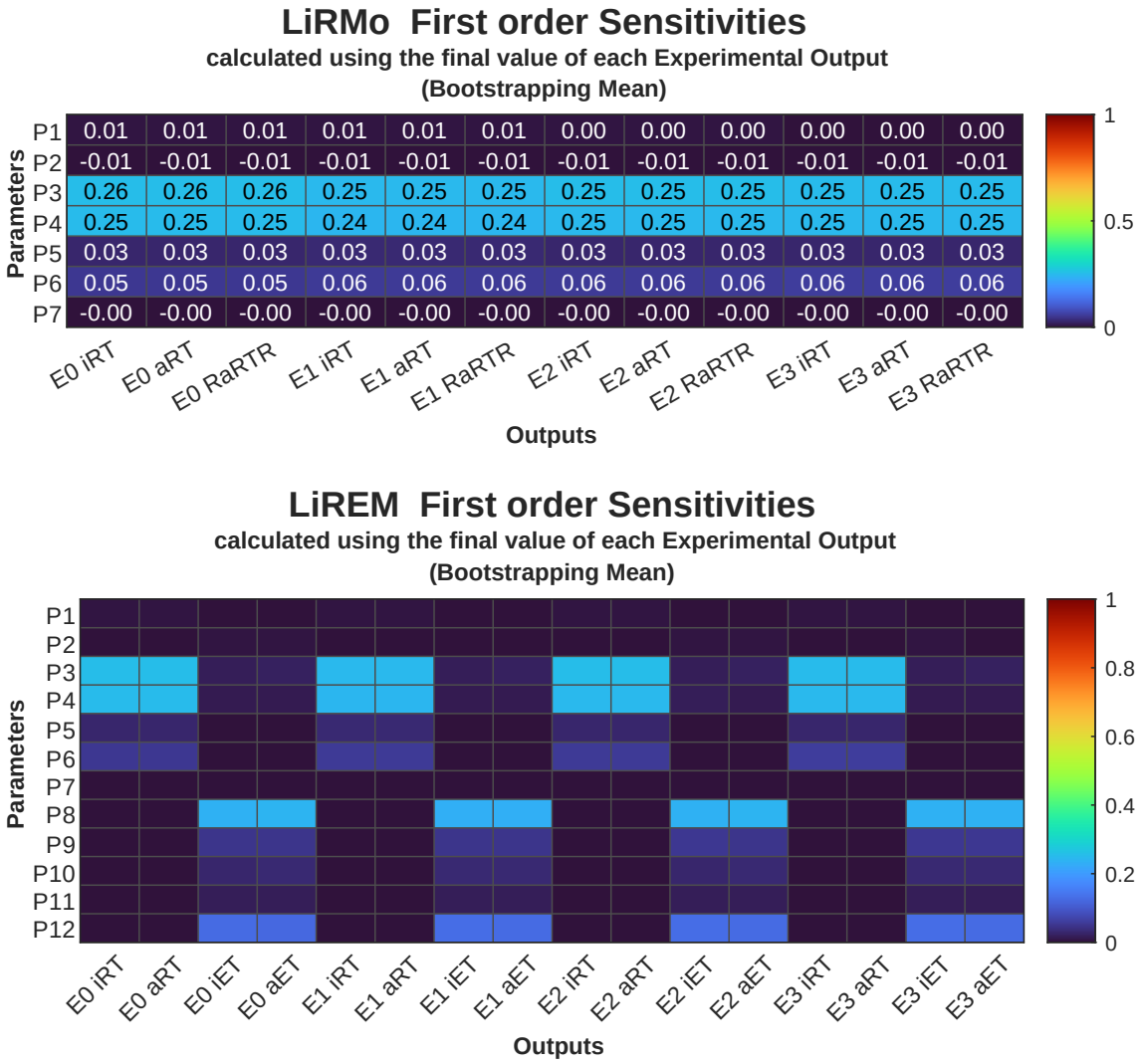


Figure B.16: LiRMo and LiREM GSA S_{Fi} for final values of each Eo. Sensitivity indices are calculated for all model parameters (P_{1-7} for LiRMo and P_{1-12} for LiREM) relative to all the model outputs (iRT, aRT, and RaRTR) from E_{0-3} as shown in section 3.6.2.1. A sample size of 10000 θ sampled as in section 3.6.2.2 was used. Higher values indicate greater sensitivity of a specific parameter given a certain experimental output.

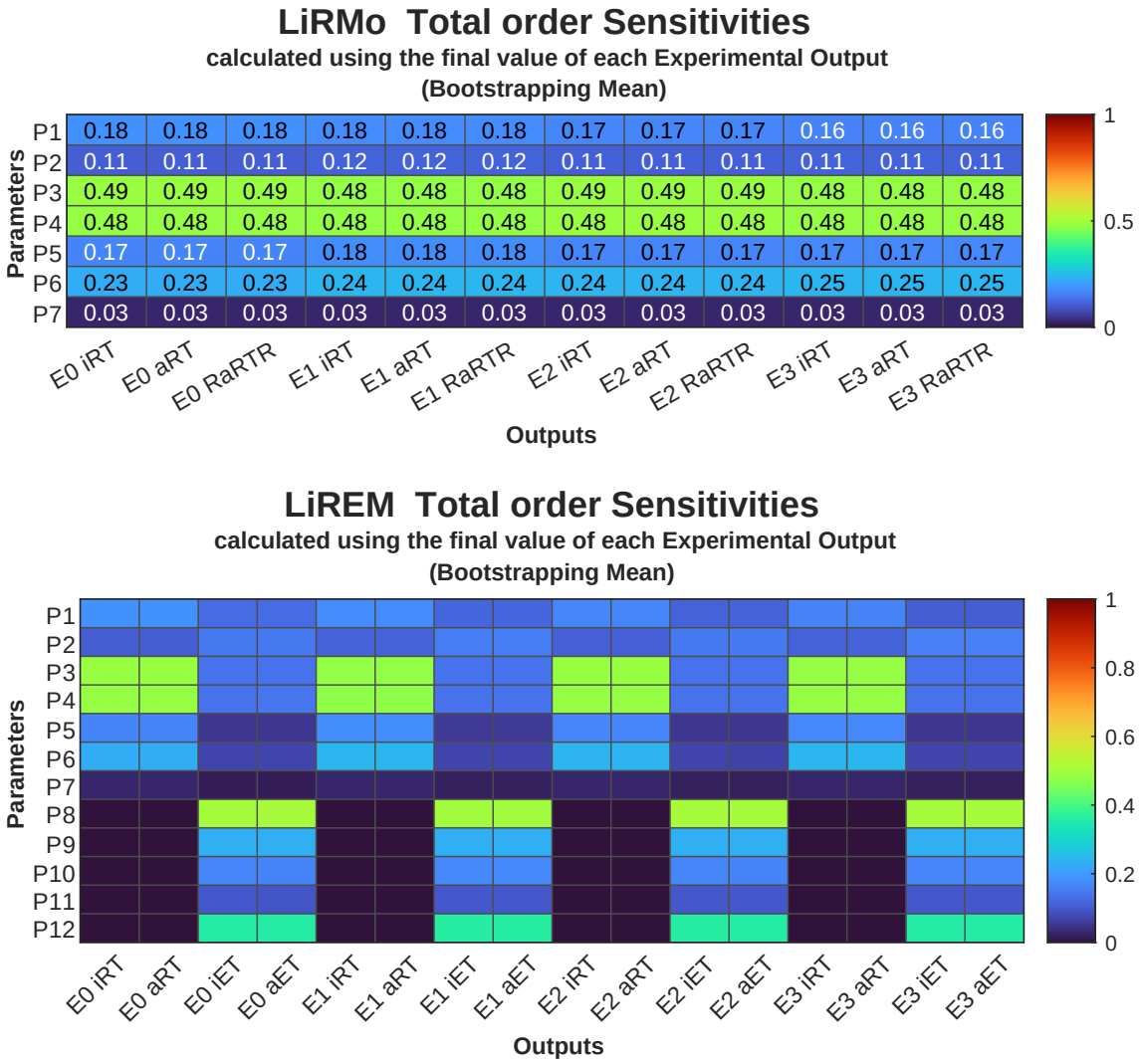


Figure B.17: LiRMo and LiREM GSA S_{Ti} for final values of each Eo. Sensitivity indices are calculated for all model parameters (P_{1-7} for LiRMo and P_{1-12} for LiREM) relative to all the model outputs (iRT, aRT, and RaRTR) from E_{0-3} as shown in section 3.6.2.1. A sample size of 10000 θ sampled as in section 3.6.2.2 was used. Higher values indicate greater sensitivity of a specific parameter given a certain experimental output.

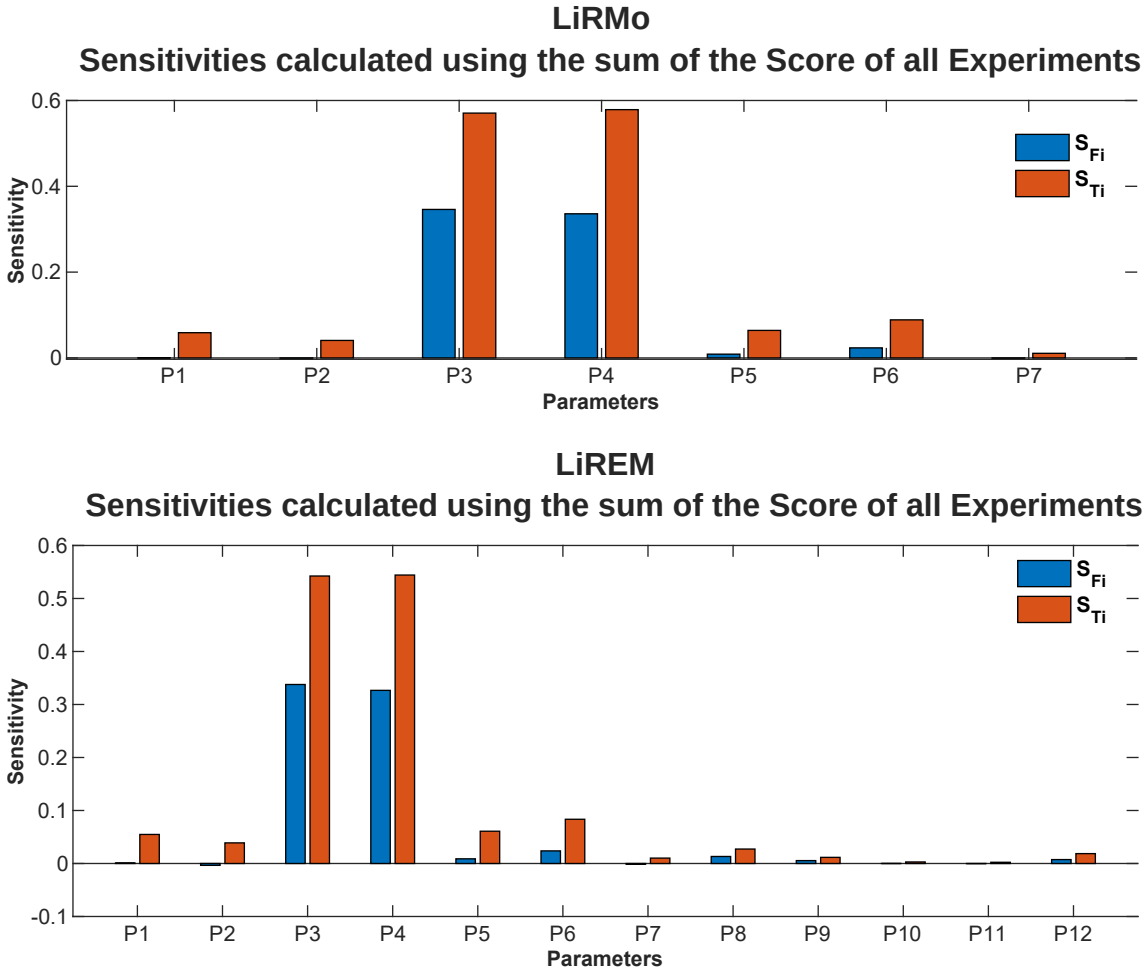


Figure B.18: LiRMo and LiREM GSA S_{Fi} and S_{Ti} for all E. S_{Fi} and S_{Ti} are calculated for all model parameters (P_{1-7} for LiRMo and P_{1-12} for LiREM) given LiRMo s_t as shown in section 3.6.2.1. A sample size of 10000 θ sampled as in section 3.6.2.2 was used. Higher values indicate greater sensitivity of a specific parameter given a certain experimental output.

C TM-M

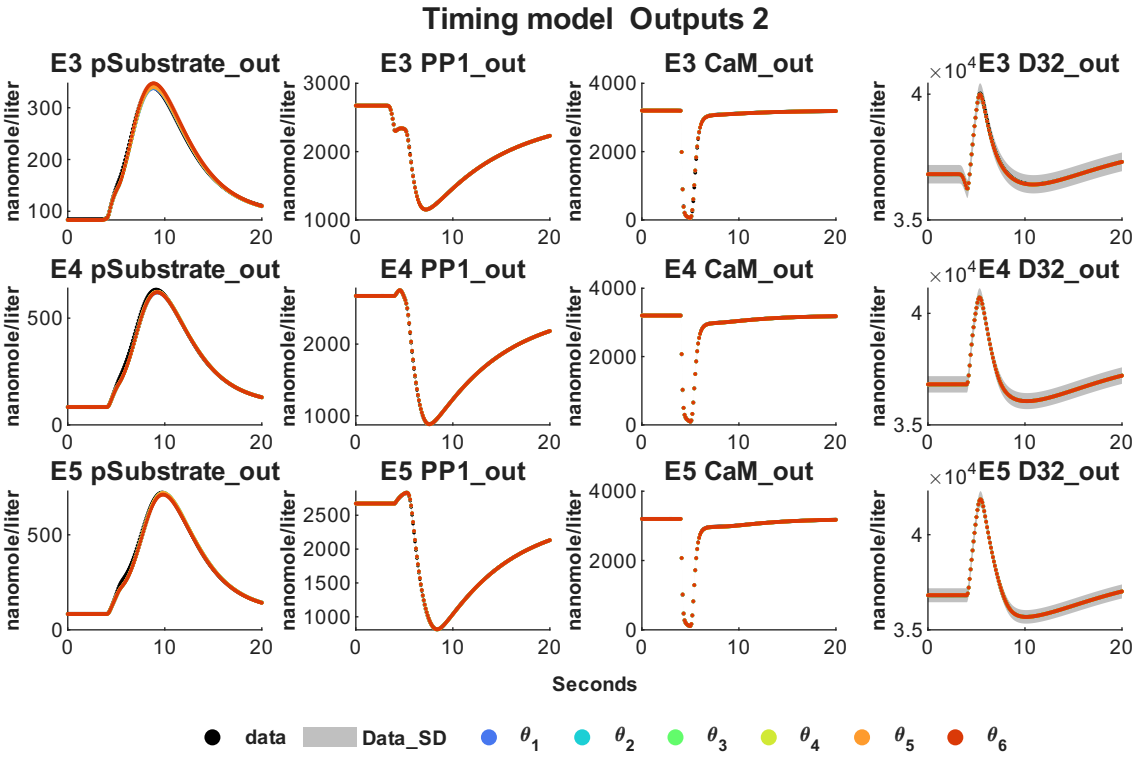


Figure C.1: TM-M outputs $E_{3–5}$. The figures display the temporal dynamics of key molecular components: pSubstrate, PP1, CaM, and D32 (respectively pSubstrate_out, PP1_out, CaM_out, D32_out) within the Timing Model. Each plot contrasts multiple parameter sets ($\theta_{1–6}$) against the observed data and its standard deviation (Data_SD), reflecting the model's ability to capture the biochemical behaviour in $E_{3–5}$.

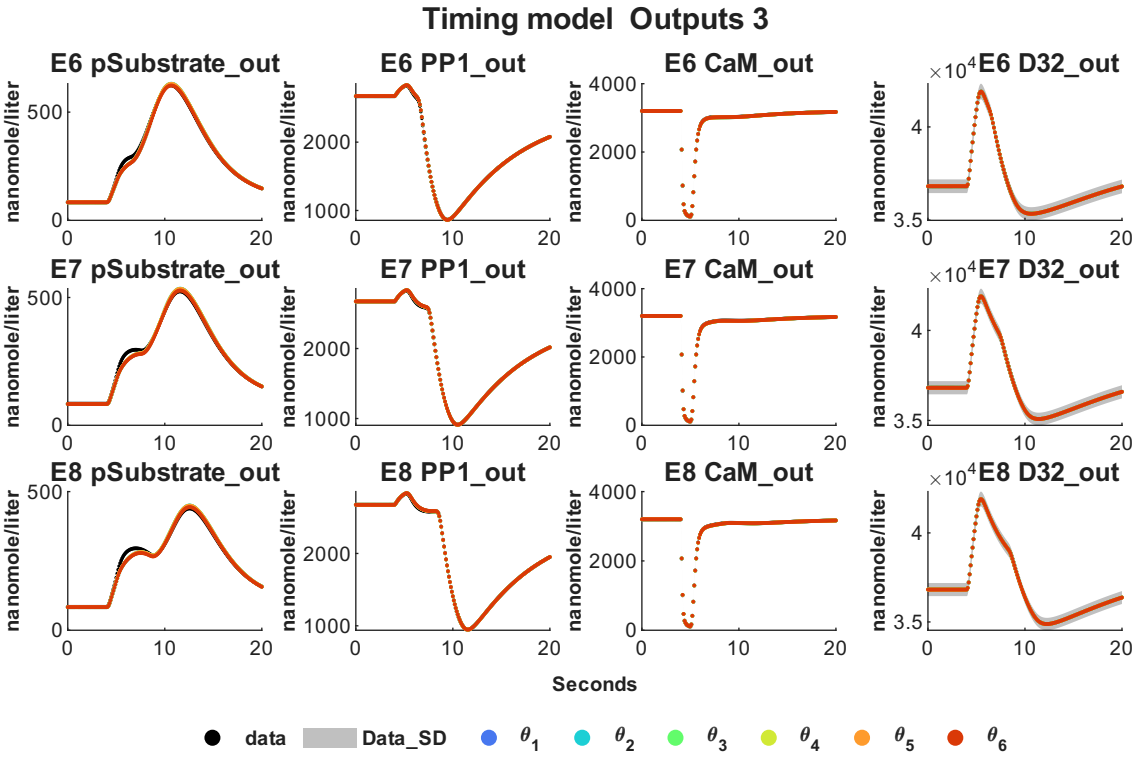


Figure C.2: TM-M outputs E_{6-8}. The figures display the temporal dynamics of key molecular components: pSubstrate, PP1, CaM, and D32 (respectively pSubstrate_out, PP1_out, CaM_out, D32_out) within the Timing Model. Each plot contrasts multiple parameter sets (θ_{1-6}) against the observed data and its standard deviation (Data_SD), reflecting the model's ability to capture the biochemical behaviour in scenarios E_{6-8}.

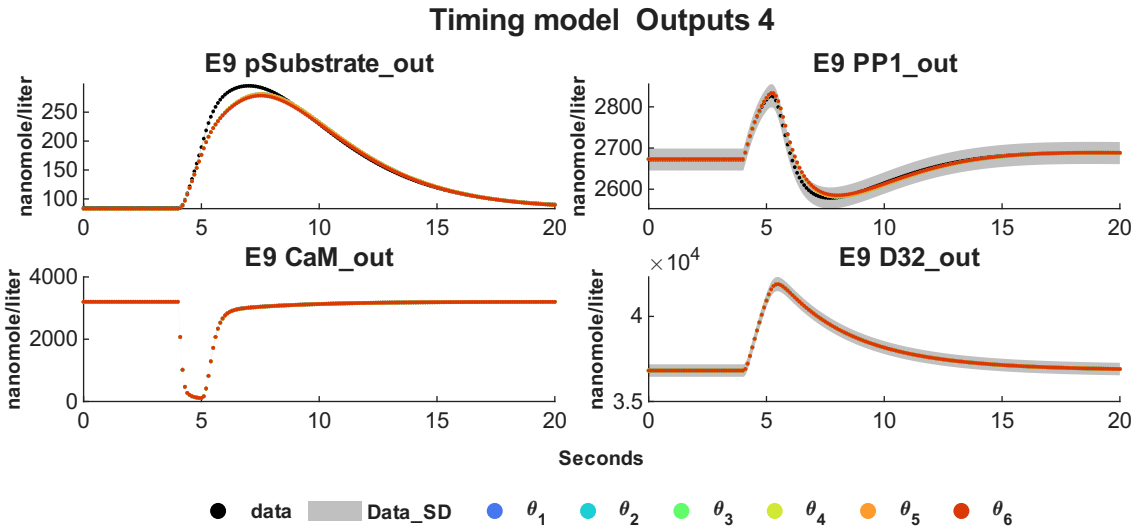


Figure C.3: TM-M output E_9. The figures display the temporal dynamics of key molecular components: pSubstrate, PP1, CaM, and D32 (respectively pSubstrate_out, PP1_out, CaM_out, D32_out) within the Timing Model. Each plot contrasts multiple parameter sets (θ_{1-6}) against the observed data and its standard deviation (Data_SD), reflecting the model's ability to capture the biochemical behaviour in E_9.

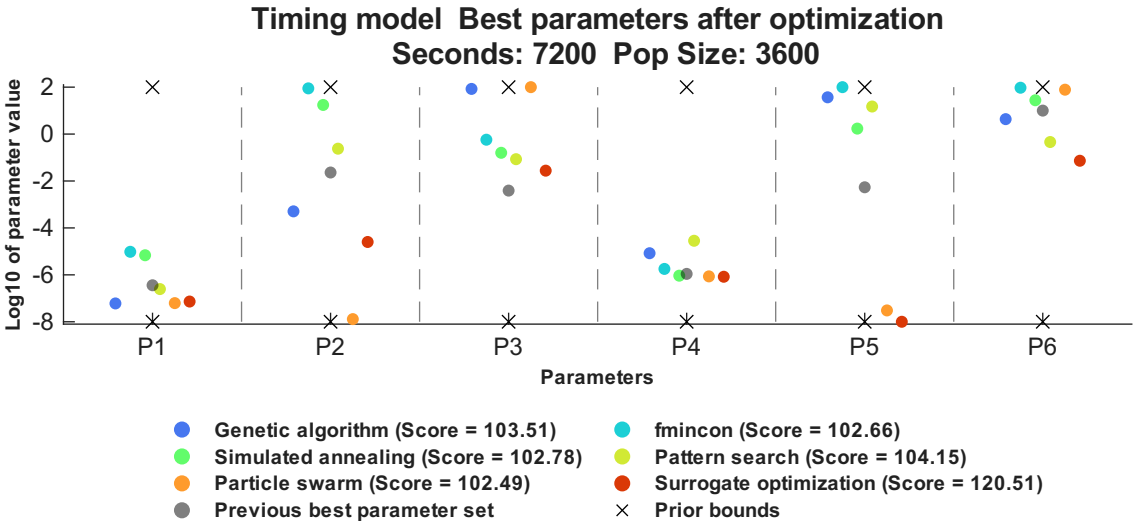


Figure C.4: TM-M optimized θ. This scatter plot presents the best parameter sets obtained from each optimization algorithm, expressed as the log-transformed values for parameters P_{1-6}. Each color represents a different algorithm, with their respective optimization scores indicated in the legend. Black crosses denote the prior parameter bounds. The plot reveals the convergence of parameter values toward optimal solutions within the defined search space. The computational time was fixed at 7200 seconds with a population size of 3600, illustrating the effectiveness of each algorithm within these constraints.

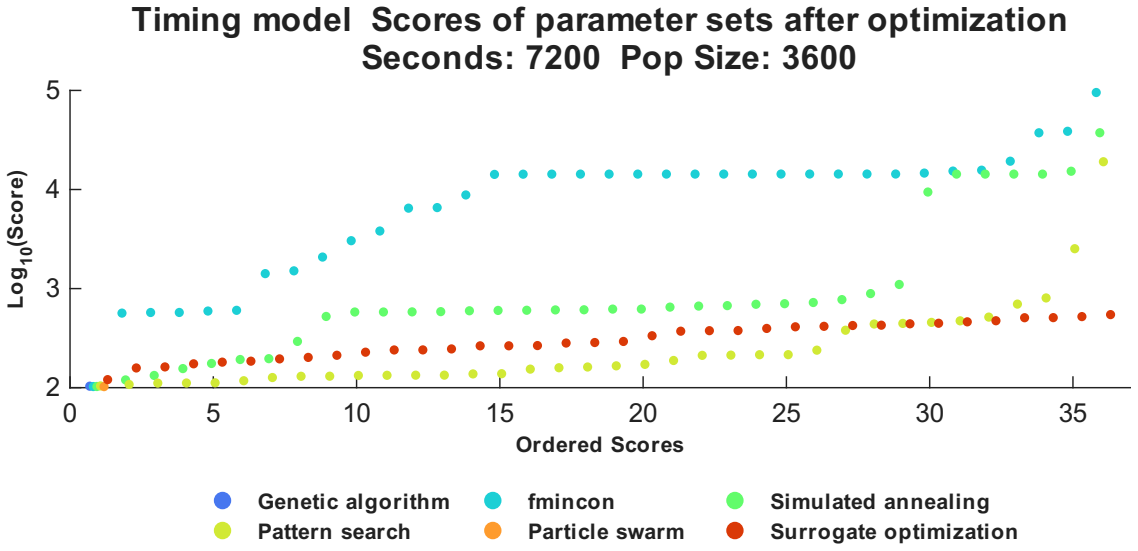


Figure C.5: TM-M optimized θ comparative scores. This plot showcases the log-scaled scores for optimized parameter sets derived from various algorithms for the Timing Model. Each dot corresponds to the score of a parameter set, arranged on the horizontal axis by increasing score. The vertical axis denotes the logarithm of the score, reflecting the fit quality. All algorithms were assessed under identical computational constraints, with a fixed runtime of 7200 seconds (using an Intel i9-10980XE) and a population size of 3600 (genetic algorithm and particle swarm), to facilitate an equitable comparison of their performance in optimizing the Timing Model parameters.

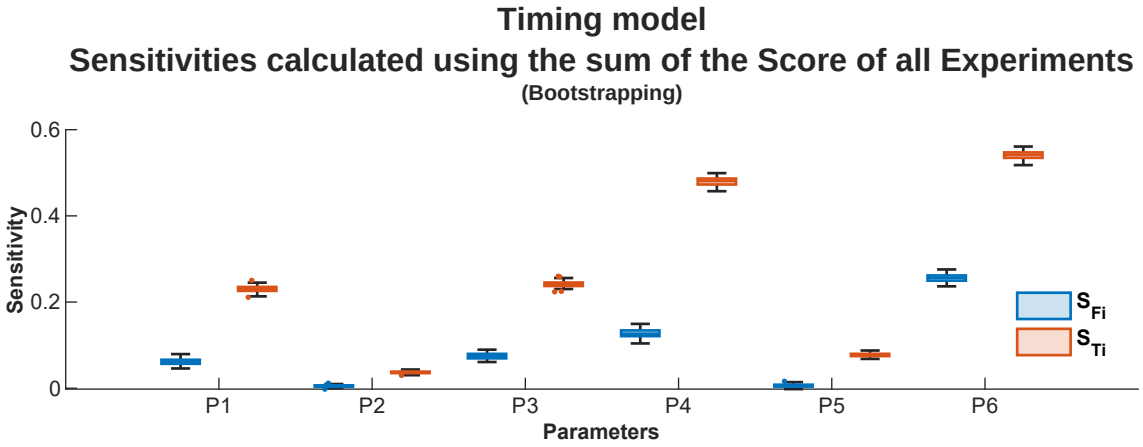


Figure C.6: TM-M GSA S_{Fi} and S_{Ti} for all E. S_{Fi} and S_{Ti} are calculated for all model parameters ($P_{1\text{-}6}$) given each model s_t, as shown in section 3.6.2.1. A sample size of 10000 θ sampled as in section 3.6.2.2 was used. Bootstrapping was performed as shown in section 3.7. The box plots have a darker line for the median of the sampled sensitivities, the top and bottom of the box correspond to the 75% and 25% percentile, respectively, and the top and bottom of the whisker correspond to the non outlier maximum and minimum of the sample, respectively [449]. Higher values indicate greater sensitivity of a specific parameter given a certain experimental output.

D G-PIPK

Analysis	Model	Diagram	Description
P_1	kf_1, kf_2, kf_3, kf_4, kf_30, kf_31, kf_32, kf_33	1f, 2f, 3f, 4f	Endogenous ligand binds receptor
P_2	kr_1, kr_30	1r	Endogenous ligand unbinds lone receptor
P_3	kf_5, kf_34, kf_7, kf_36	5f, 7f	$G_{\beta\gamma}$ or G-protein bind receptor
P_4	kr_5, kr_34, kr_7, kr_36	5r, 7r	$G_{\beta\gamma}$ or G-protein unbind receptor
P_5	kf_6, kf_23, kf_35, kf_52, kf_8, kf_24, kf_37, kf_53	6f, 8f	$G_{\beta\gamma}$ or G-protein bind receptor-ligand complex
P_6	kr_6, kr_23, kr_35, kr_52, kr_8, kr_24, kr_37, kr_53	6r, 8r	$G_{\beta\gamma}$ or G-protein unbind receptor-ligand complex
P_7	kf_9, kf_15, kf_38, kf_44, kf_59	9, 15, 19	G_αGDP binds $G_{\beta\gamma}$ free or in receptor complex (no ligand)
P_8	kf_10, kf_16, kf_25, kf_28, kf_39, kf_45, kf_54, kf_57	10, 16	$G_{\beta\gamma}$ binds G_αGDP in receptor-ligand complex
P_9	kf_11, kf_17, kf_40, kf_46, kf_60	11, 17, 20	G-protein splits to $G_{\beta\gamma}$ and G_αGTP free or in receptor complex (no ligand)
P_{10}	kf_12, kf_18, kf_26, kf_29, kf_41, kf_47, kf_55, kf_58	12, 18	G-protein splits to $G_{\beta\gamma}$ and G_αGTP in receptor-ligand complex
P_{11}	kf_13, kf_42, kf_63, kf_64	13f, 23f, 24f	PLC binds free G_αGTP/G_αGDP or G-protein in receptor (no ligand)
P_{12}	kr_13, kr_42, kr_63, kr_64	13r, 23r, 24r	PLC unbinds free G_αGTP/G_αGDP or G-protein in receptor (no ligand)
P_{13}	kf_14, kf_27, kf_43, kf_56	14f	PLC binds G-protein in receptor-ligand complex
P_{14}	kr_14, kr_27, kr_43, kr_56	14r	PLC unbinds to G-protein in receptor-ligand complex
P_{15}	kf_19, kf_20, kf_21, kf_22, kf_48, kf_49, kf_50, kf_51	1f, 2f, 3f, 4f	Exogenous ligand binds receptor

Table D.1 continued from previous page

Analysis	Model	Diagram	Description
P_{16}	kr_19, kr_48	1r	Exogenous ligand unbinds lone receptor
P_{17}	kf_61	21	G_{α}GTP hydrolyzes to G_{α}GDP
P_{18}	kf_62	22f	G_{α}GTP-PLC hydrolyzes to G_{α}GDP-PLC
P_{19}	kr_62	22r	G_{α}GDP-PLC turns into G_{α}GTP-PLC
P_{20}	kf_65	25f	PI converts to PIP
P_{21}	kr_65	25r	PIP converts back to PI
P_{22}	kf_66	26f	PIP converts to PIP_2
P_{23}	kr_66	26r	PIP_2 converts back to PIP
P_{24}	kf_67	27	PIP_2 cleaves to IP_3 and DAG
P_{25}	kf_68	28	PIP_2 cleaves to IP_3 and DAG catalized by G_{α}GTP-PLC
P_{26}	kf_69	29f	PIP_2 binds KCNQ2/3
P_{27}	kr_69	29r	PIP_2 unbinds KCNQ2/3
P_{28}	kf_70	30f	PIP_2 binds PH
P_{29}	kr_70	30r	PIP_2 unbinds PH
P_{30}	kf_71	31f	IP_3 binds PH
P_{31}	kr_71	31r	IP_3 unbinds PH
P_{32}	kf_72	32	IP_3 converts to Ino
P_{33}	kf_73	33	DAG converts to PA
P_{34}	kf_74	34	PA converts to CDPDAG
P_{35}	kf_75	35	CDPDAG and Ino form PI

Table D.1: G-PIPK free parameters. The table shows the mapping between different parameter notations: Analysis: P_{1-35} represent the 35 free parameters used in model analysis. Model: kf_X and kr_X correspond to the rate constants in the SBtab model file, where X is the reaction number (1–75 in the SBtab, with only some shown in this table). Diagram: numbers refer to the 35 reactions in figure 4.25. Description: briefly explains the biological meaning of each parameter group. f indicates forward reactions, while r denotes reverse reactions.

Analysis	Model	Diagram	Description
$\frac{P_2 \times P_3 \times P_6}{P_4 \times P_5}$	kr_2, kr_31, kr_3, kr_32	2r, 3r	Endogenous ligand unbinds receptor bound to $G_{\beta\gamma}$ or G-protein
$\frac{P_{11} \times P_3 \times P_2 \times P_{14} \times P_6}{P_{13} \times P_5 \times P_{12} \times P_4}$	kr_4, kr_33	4r	Endogenous ligand unbinds receptor bound to G-protein and PLC
$\frac{P_3 \times P_{16} \times P_6}{P_5 \times P_4}$	kr_20, kr_49, kr_21, kr_50	2r, 3r	Exogenous ligand unbinds receptor bound to $G_{\beta\gamma}$ or G-protein
$\frac{P_{11} \times P_3 \times P_{14} \times P_{16} \times P_6}{P_{13} \times P_5 \times P_{12} \times P_4}$	kr_22, kr_51	4r	Exogenous ligand unbinds receptor bound to G-protein and PLC

Table D.2: G-PIPK constrained parameters. The table shows the mapping between different parameter notations: Analysis: Represents the mathematical relationships between free parameters (P_{1-35}) due to thermodynamic constraints that restricts some parameters. Model: kr_X correspond to the constrained rate constants in the SBtab model file, where X is the reaction number (1–75 in the SBtab, with only some shown in this table). Diagram: numbers refer to the 35 reactions in figure 4.25. Description: briefly explains the biological meaning of each constrained parameter. r denotes reverse reactions.

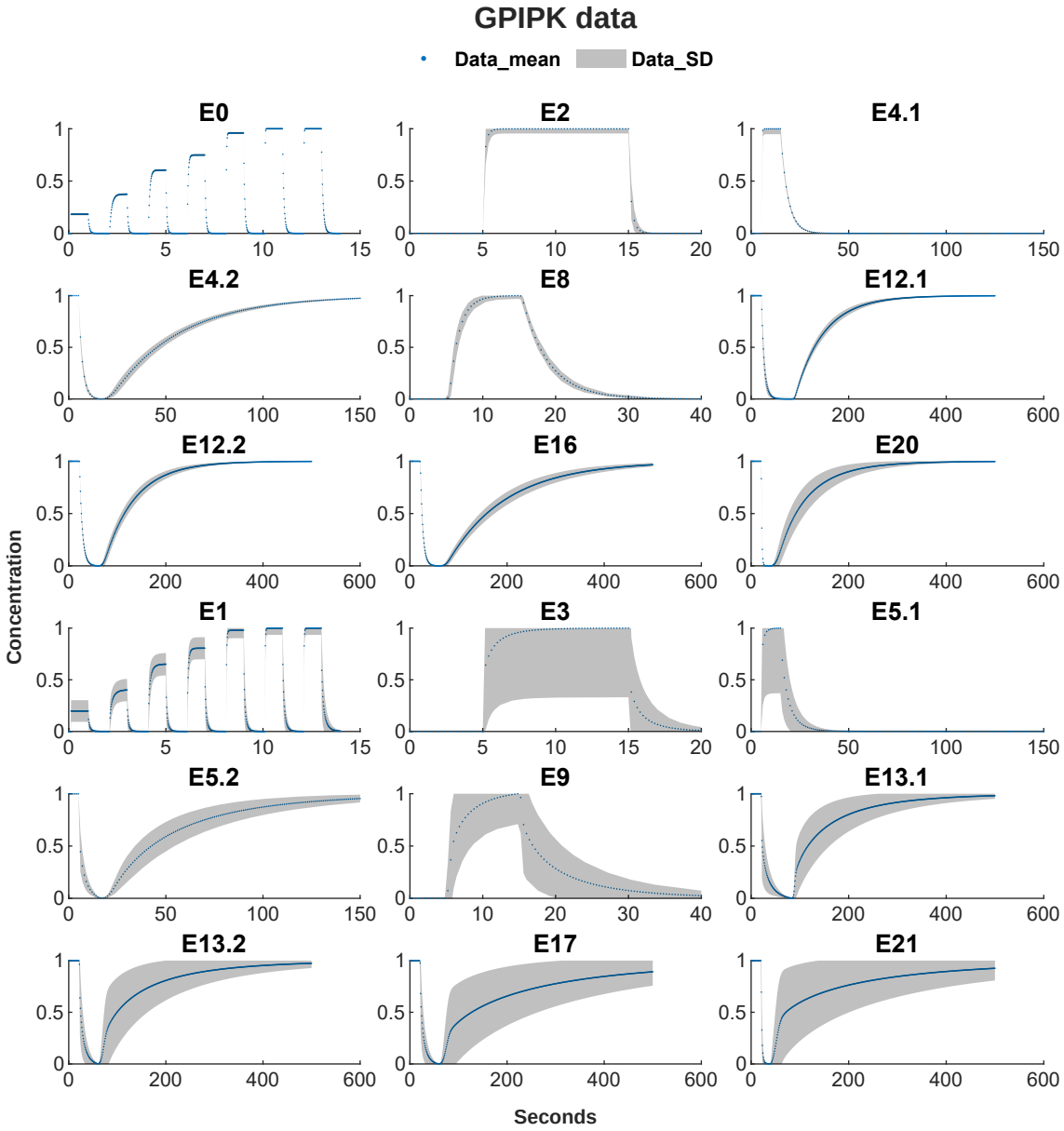


Figure D.1: G-PIPK data. This figure presents experimental and derived data for the G-PIPK signalling pathway model. Blue dots represent mean values of measured or derived data points, while gray shaded areas indicate standard deviation. Even-numbered experiments (top nine plots) use data sourced directly from published articles, while odd-numbered experiments (bottom nine plots) use artificial data derived from corresponding even-numbered experiments to further constrain the model. Each plot corresponds to a specific output, with E_0 showing total amount of Glu bound mGluR (mGluR_Glu_T); E_1 total amount of DHPG bound mGluR (mGluR_DHPG_T); E_2 total amount of OxoM bound M1R _OxoM_T; E_3 total amount of ACh bound M1R _ACh_T; $E_{4.1}$ total amount of M1R bound Gq (M1R_Gbc_T); $E_{5.1}$ total amount of mGluR bound Gq (mGluR_Gbc_T); $E_{4.2,5.2}$ total amount of the Gqabc complex (Gabc_T); $E_{8,9}$ total amount of PLC bound Gqa (Ga_PLC_T); $E_{12.1,13.1}$ total amount of PH bound PIP2 (PIP2_PH_T); $E_{12.2,13.2,16,17,20,21}$ measure of Kv7.2/3 membrane current (M_current). The varied durations and response patterns across experiments reflect different input conditions and measured outputs, capturing the complexity of the G-PIPK signalling pathway.

$Name$	$maxA$	$\overline{A}$	$A\sigma$	$start$	$\overline{start_d}$	$start_d\sigma$	$\overline{start_s}$	$start_s\sigma$	end	$\overline{end_d}$	$end_d\sigma$	$\overline{end_s}$	$end_s\sigma$	E_t	N
E0.1	1	0.181	0.010	0.1	0	0	0	0	1	0	0	0.059	0.005	2	100
E0.2	1	0.371	0.010	0.1	0	0	0.109	0.01	1	0	0	0.047	0.005	2	100
E0.3	1	0.600	0.010	0.1	0	0	0.069	0.0075	1	0	0	0.045	0.005	2	100
E0.4	1	0.743	0.010	0.1	0	0	0.043	0.003	1	0	0	0.05	0.005	2	100
E0.5	1	0.952	0.010	0.1	0	0	0.02	0.001	1	0	0	0.056	0.005	2	100
E0.6	1	1	0.010	0.1	0	0	0.01	0.001	1	0	0	0.072	0.005	2	100
E0.7	1	1	0.010	0.1	0	0	0.01	0.001	1	0	0	0.08	0.01	2	100
E2	1	1	0.071	5	0	0	0.1	0.1	15	0	0	0.18	0.1	20	100
E4.1	1	0.95	0.064	5	0	0	0.2	0.1	15	0	0	3.7	0.2	150	150
E4.2	-1	-1	0.042	5	0	0.05	2	0.3	15	5.8	2.9	35	2	150	150
E8	1	0.94	0.032	5	0.38	0.25	1.3	0.3	15	0.34	0.14	3.6	0.5	40	100
E12.1	-1	-1	0.015	20	1.3	0.2	5.4	1.6	60	29	2	59	7	500	500
E12.2	-1	-1	0.010	20	1.4	0.3	5	0.6	60	34	6	123	20	500	500
E16	-1	-1	0.010	20	0.78	0.07	1.2	0.1	40	11	7	62	22	500	500
E20	-1	-1	0.010	20	2.1	0.1	5.7	0.7	40	11	5	63	9	500	500

Table D.3: G-PIPK data generating parameters 1. Parameters used for 7 experiments ($E_{0,2,4,8,12,16,20}$) for all the outputs of each experiment. These parameters determine the characteristics of the generated data, including the maximum amplitude change ($maxA$), mean amplitude change ($\overline{A}$), standard deviation of the amplitude change($A\sigma$), default start and end times($start$,end), mean delays at the start and end times($\overline{start_d}$,$\overline{end_d}$) and their corresponding standard deviations($start_d\sigma$,$end_d\sigma$). Additionally, the mean of the shape parameters of the start and end times($\overline{start_s}$,$\overline{end_s}$) and their standard deviations are also listed($start_s\sigma$,$end_s\sigma$). The table also specifies the total duration of each experiment(E_t) and the number of data points collected (N).

$Name$	$maxA$	$\overline{A}$	$A\sigma$	$start$	$\overline{start_d}$	$start_d\sigma$	$\overline{start_s}$	$start_s\sigma$	end	$\overline{end_d}$	$end_d\sigma$	$\overline{end_s}$	$end_s\sigma$	E_t	N
E1.1	1	0.181	0.100	0.1	0	0	0	0	1	0	0	0.059	0.05	2	100
E1.2	1	0.371	0.100	0.1	0	0	0.109	0.1	1	0	0	0.047	0.05	2	100
E1.3	1	0.6	0.100	0.1	0	0	0.069	0.075	1	0	0	0.045	0.05	2	100
E1.4	1	0.743	0.100	0.1	0	0	0.043	0.03	1	0	0	0.05	0.05	2	100
E1.5	1	0.952	0.100	0.1	0	0	0.02	0.01	1	0	0	0.056	0.05	2	100
E1.6	1	1	0.100	0.1	0	0	0.01	0.01	1	0	0	0.072	0.05	2	100
E1.7	1	1	0.100	0.1	0	0	0.01	0.01	1	0	0	0.08	0.1	2	100
E3	1	1	0.707	5	0	0	0.1	1	15	0	0	0.18	1	20	100
E5.1	1	0.95	0.643	5	0	0	0.2	1	15	0	0	3.7	2	150	150
E5.2	-1	-1	0.416	5	0	0.05	2	3	15	5.8	2.9	35	20	150	150
E9	1	0.94	0.316	5	0.38	0.25	1.3	3	15	0.34	0.14	3.6	5	40	100
E13.1	-1	-1	0.145	20	1.3	0.2	5.4	16	60	29	2	59	70	500	500
E13.2	-1	-1	0.100	20	1.4	0.3	5	6	60	34	6	123	200	500	500
E17	-1	-1	0.100	20	0.78	0.07	1.2	1	40	11	7	62	220	500	500
E21	-1	-1	0.100	20	2.1	0.1	5.7	7	40	11	5	63	90	500	500

Table D.4: G-PIPK data generating parameters 2. Parameters used for 7 experiments ($E_{1,3,5,9,13,17,21}$) for all the outputs of each experiment. These parameters determine the characteristics of the generated data, including the maximum amplitude change ($maxA$), mean amplitude change ($\overline{A}$), standard deviation of the amplitude change($A\sigma$), default start and end times($start$,end), mean delays at the start and end times($\overline{start_d}$,$\overline{end_d}$) and their corresponding standard deviations($start_d\sigma$,$end_d\sigma$). Additionally, the mean of the shape parameters of the start and end times($\overline{start_s}$,$\overline{end_s}$) and their standard deviations are also listed($start_s\sigma$,$end_s\sigma$). The table also specifies the total duration of each experiment(E_t) and the number of data points collected (N).

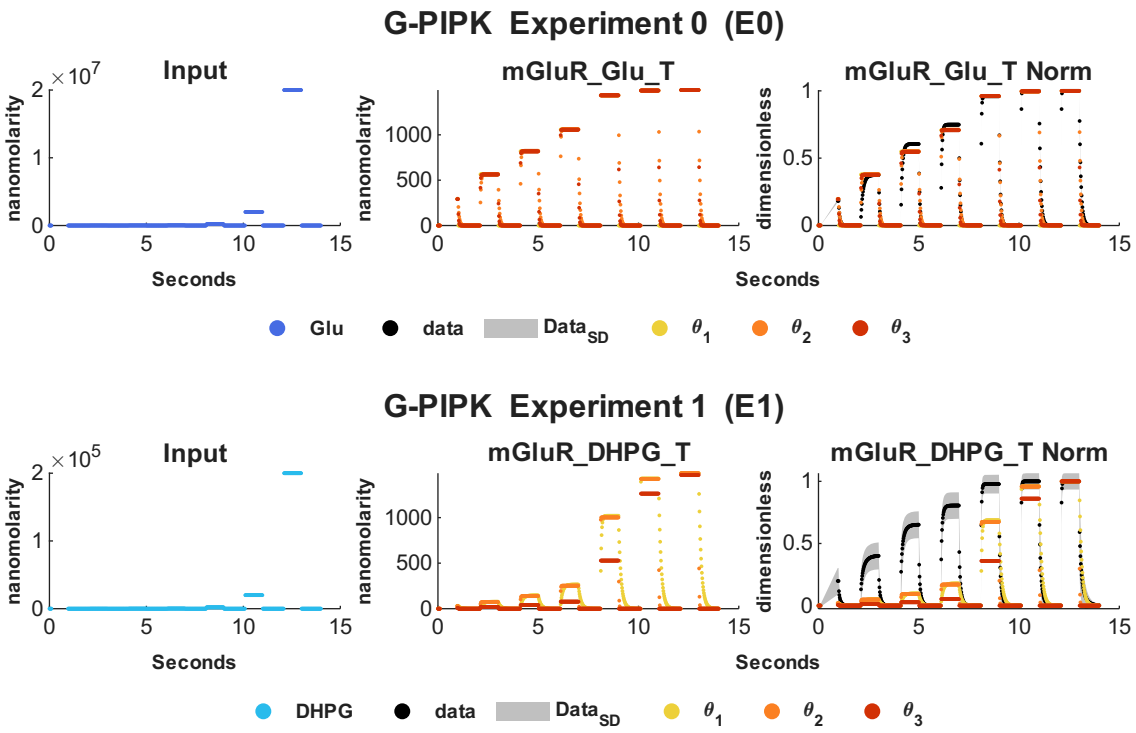


Figure D.2: G-PIPK $E_{0,1}$ diagnostics plot. Shows the input, output data with standard deviation, and θ_{1-3} outputs for G-PIPK $E_{0,1}$. Inputs: multiple impulses of Glu for E_0 and DHPG for E_1 at varying concentrations (2×10^1, 5×10^1, 1×10^2, 2×10^2, 2×10^3, 2×10^4, 2×10^5 nMol/L for Glu, 100 times less for DHPG) to represent the influence of different input amounts. Outputs: mGluR_Glu_T (sum of all model species where glutamate is bound to mGluR), mGluR_Glu_T_Norm (normalized version), mGluR_DHPG_T (sum of all model species where DHPG is bound to mGluR), mGluR_DHPG_T_Norm (normalized version). Normalized outputs scaled to 0–1 range based on min/max model responses. θ_{1-3} represents the model parametrization after the first, second or third optimization. Experimental data and standard deviation (data_SD) from Jensen *et al.* [440].

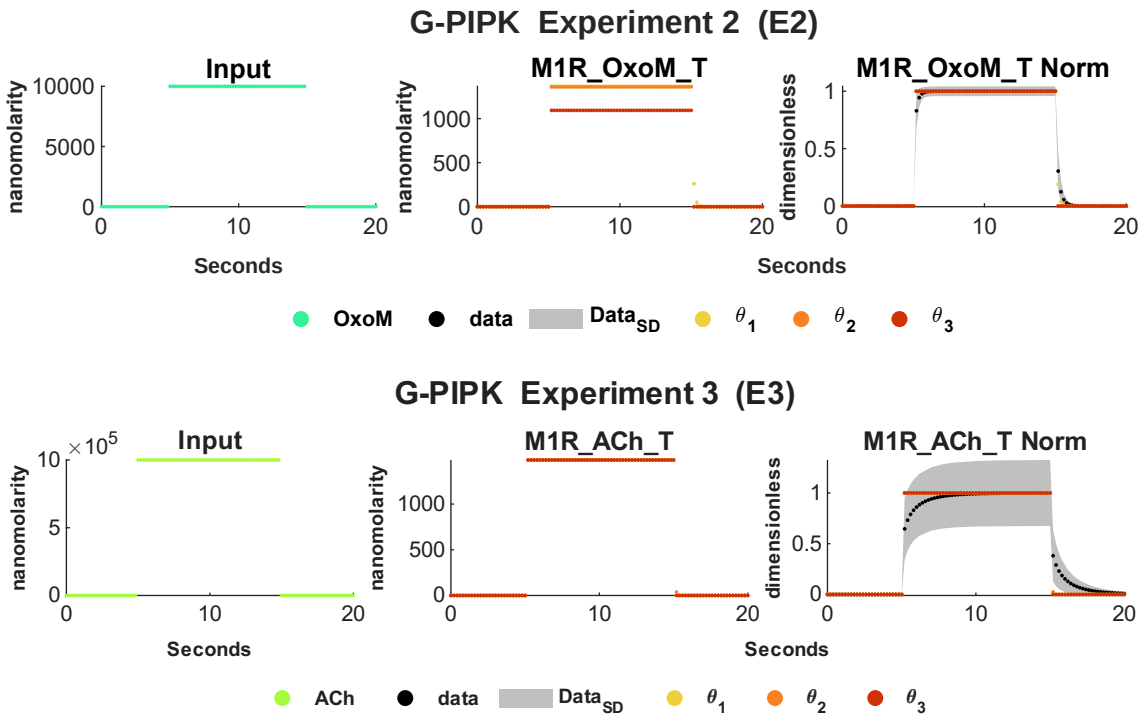


Figure D.3: G-PIPK $E_{2,3}$ diagnostics plot. Shows the input, output data with standard deviation, and θ_{1-3} outputs for G-PIPK $E_{2,3}$. Inputs: 10-second impulse of OxoM (1×10^4 nMol/L) for E_2 and ACh (1×10^6 nMol/L) for E_3 from 5 to 15 seconds. Outputs: M1R_OxoM_T (sum of all model species where OxoM is bound to M1R), M1R_OxoM_T Norm (normalized version), M1R_Ach_T (sum of all model species where ACh is bound to M1R), M1R_Ach_T Norm (normalized version). Normalized outputs scaled to 0–1 range based on min/max model responses. θ_{1-3} represents the model parametrization after the first, second or third optimization. Experimental data and standard deviation (data_SD) from Jensen *et al.* [440].

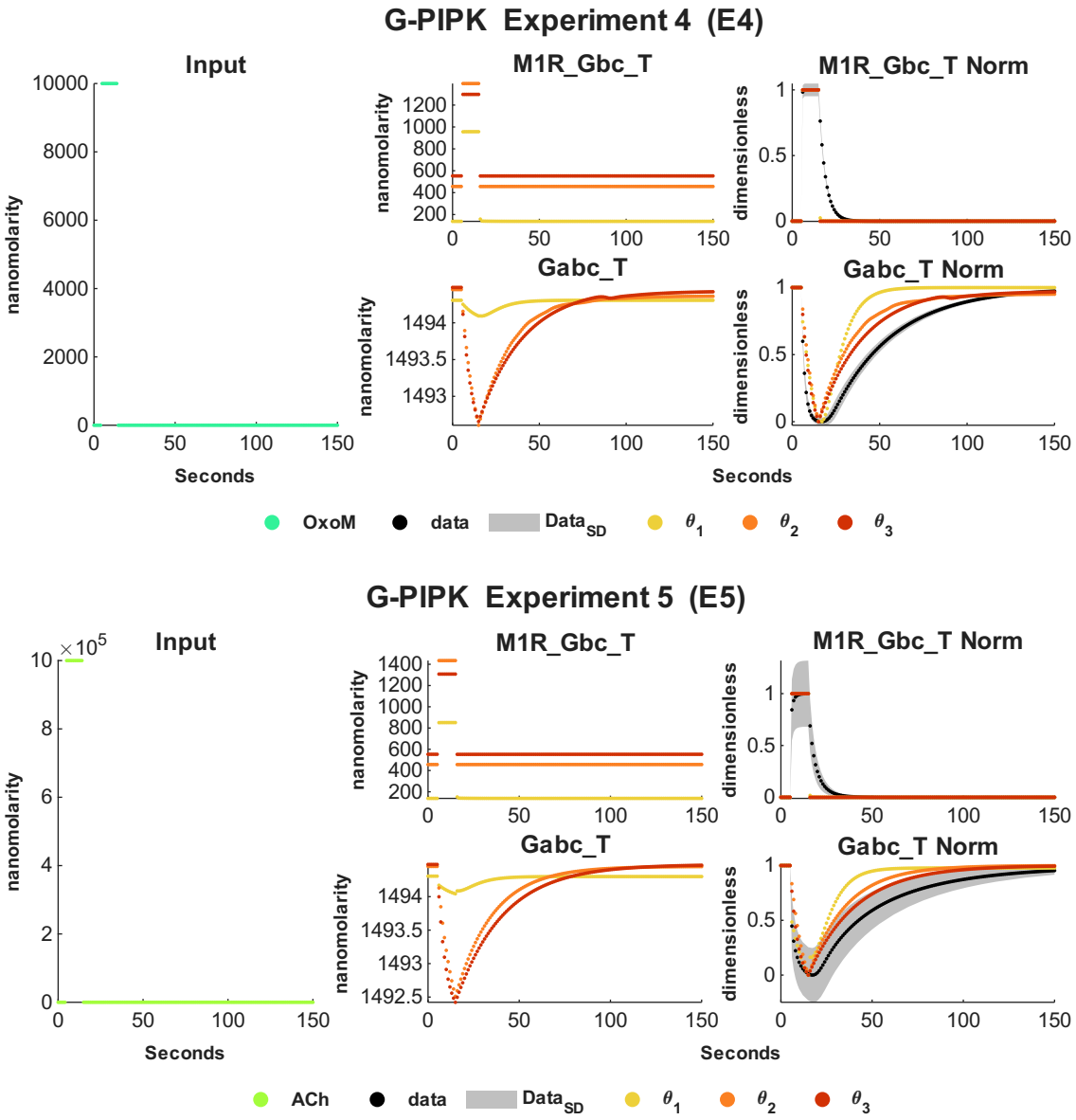


Figure D.4: G-PIPK $E_{4,5}$ diagnostics plot. Shows the input, output data with standard deviation, and θ_{1-3} outputs for G-PIPK $E_{4,5}$. Inputs: 10-second impulse of OxoM (1×10^4 nMol/L) for E_4 and ACh (1×10^6 nMol/L) for E_5 from 1 to 11 seconds. Outputs: M1R_Gbc_T (sum of all the model species where mGluR is bound to Gbc), M1R_Gbc_T Norm (normalized version), Gabc_T (sum of all the model species where G-protein trimers are present), Gabc_T_Norm (normalized version). Normalized outputs scaled to 0–1 range based on min/max model responses. θ_{1-3} represents the model parametrization after the first, second or third optimization. Experimental data and standard deviation (data_SD) from Jensen *et al.* [440].

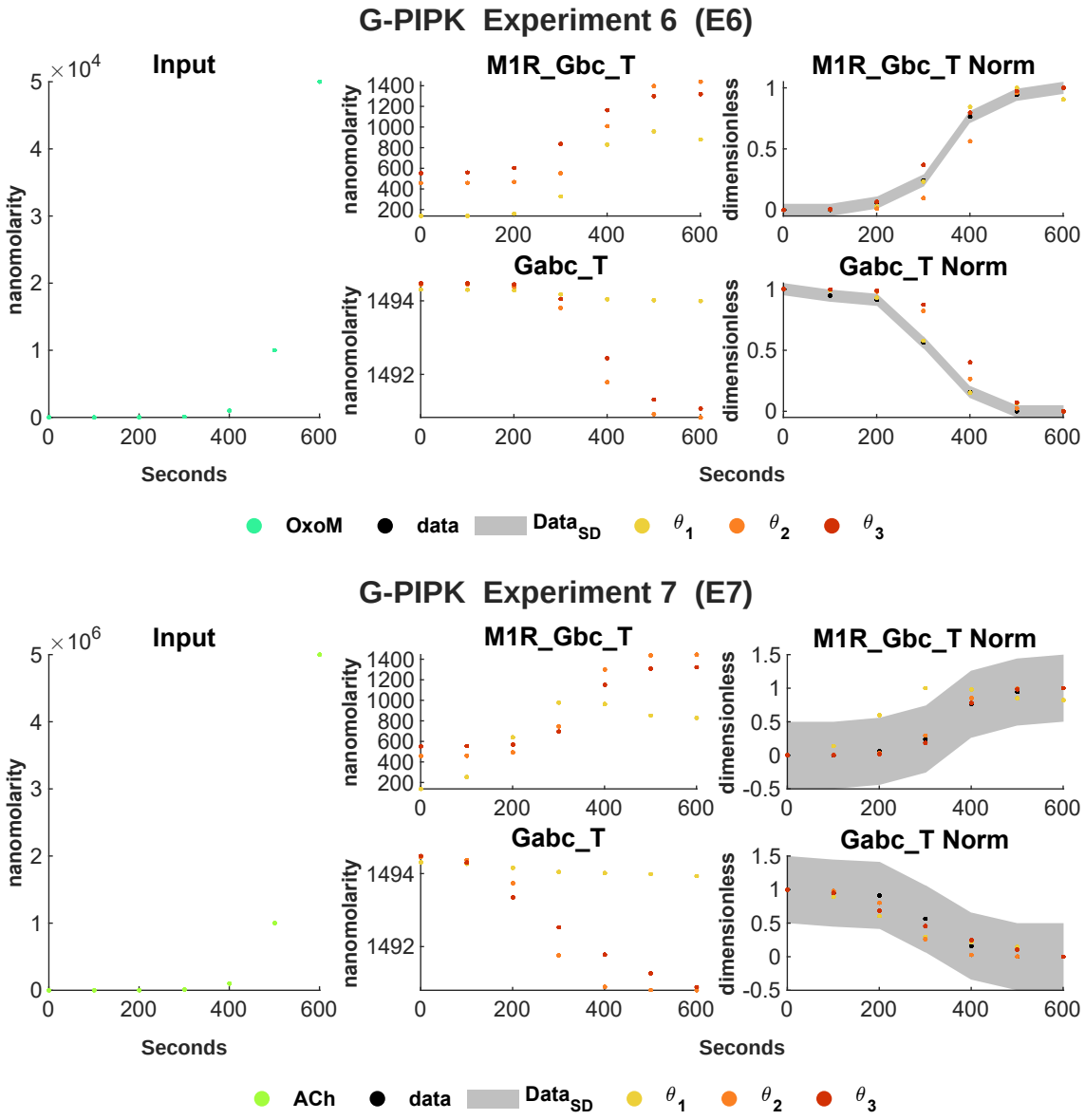


Figure D.5: G-PIPK $E_{6,7}$ diagnostics plot. Shows the input, output data with standard deviation, and θ_{1-3} outputs for G-PIPK $E_{6,7}$. Inputs: Multiple impulses of OxoM for E_6 and ACh for E_7 at varying concentrations (1×10^0, 1×10^1, 1×10^2, 1×10^3, 1×10^4, 5×10^4 nMol/L for OxoM, 100 times more for ACh) to represent the influence of different input amounts. Outputs: M1R_Gbc_T (sum of all the model species where M1R is bound to Gbc), M1R_Gbc_T Norm (normalized version), Gabc_T (sum of all the model species where G-protein trimers are present), Gabc_T_Norm (normalized version). Normalized outputs scaled to 0–1 range based on min/max model responses. θ_{1-3} represents the model parametrization after the first, second or third optimization. Experimental data and standard deviation (data_SD) from Jensen *et al.* [440].

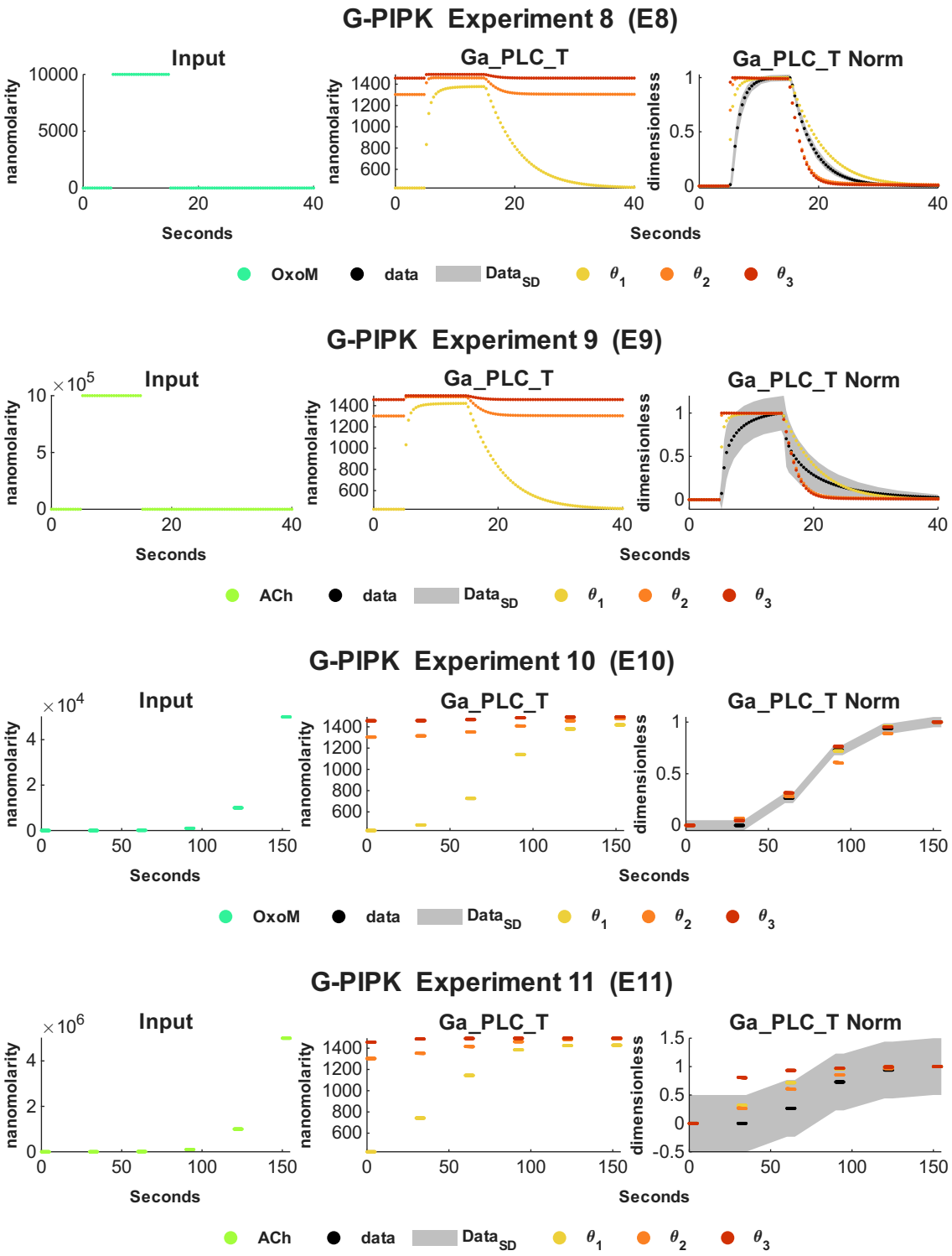


Figure D.6: G-PIPK $E_{8–11}$ diagnostics plot. Shows the input, output data with standard deviation, and $\theta_{1–3}$ outputs for G-PIPK $E_{8–11}$. Inputs: 10-second impulse of OxoM (1×10^4 nMol/L) for E_8 and ACh (1×10^6 nMol/L) for E_9 from 1 to 11 seconds. Multiple impulses of OxoM for E_{10} and ACh for E_{11} at varying concentrations (1×10^1, 1×10^2, 1×10^3, 1×10^4, 5×10^4 nMol/L for OxoM, 100 times more for ACh) to represent the influence of different input amounts. Outputs: Ga_PLC_T (sum of all the model species where G_α is bound to PLC), Ga_PLC_T_Norm (normalized version). Normalized outputs scaled to 0–1 range based on min/max model responses. $\theta_{1–3}$ represents the model parametrization after the first, second or third optimization. Experimental data and standard deviation (data_SD) from Jensen *et al.* [440].

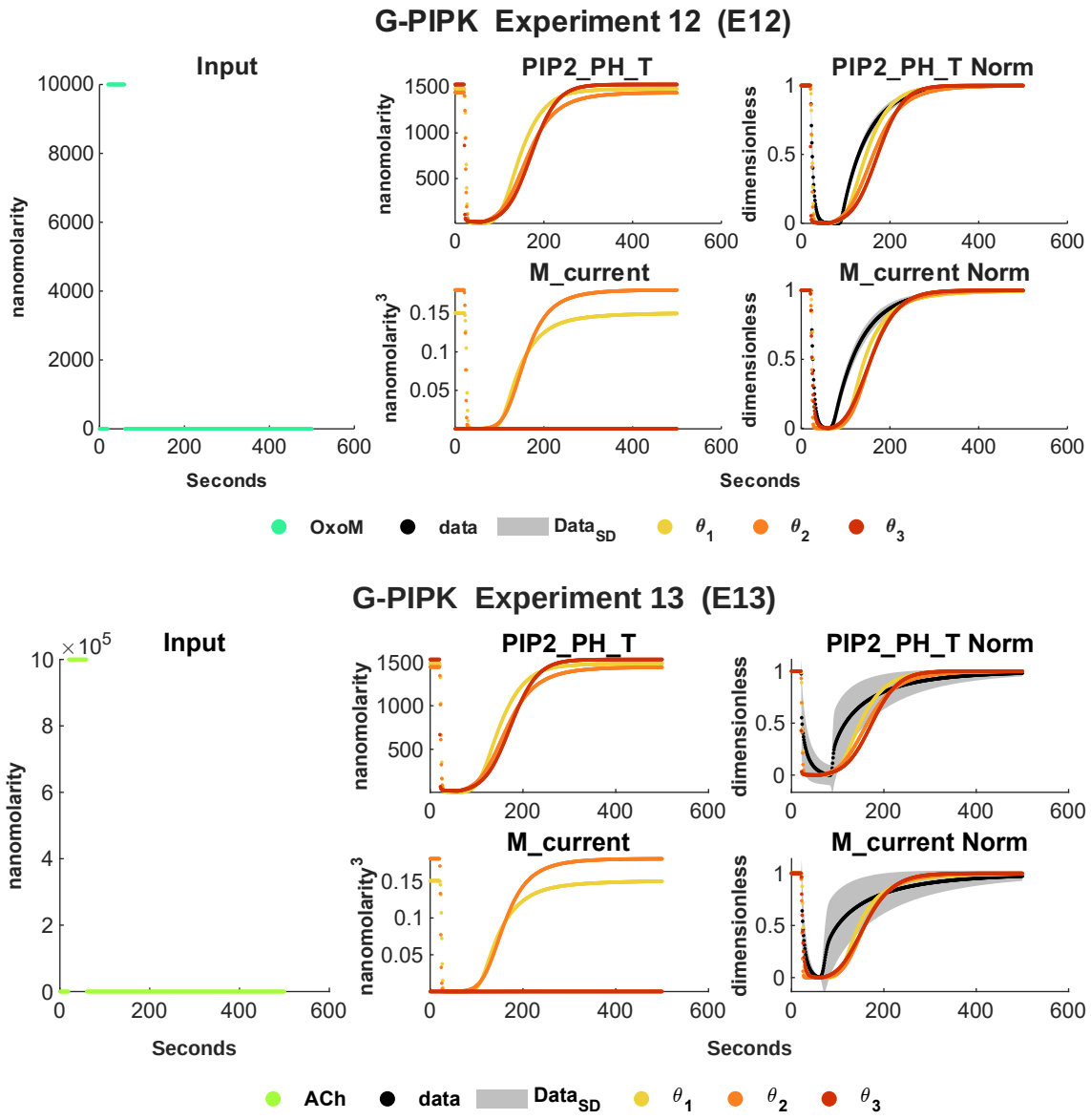


Figure D.7: G-PIPK $E_{12,13}$ diagnostics plot. Shows the input, output data with standard deviation, and θ_{1-3} outputs for G-PIPK $E_{12,13}$. Inputs: 40-second impulse of OxoM (1×10^4 nMol/L) for E_{12} and ACh (1×10^6 nMol/L) for E_{13} from 20 to 60 seconds. Outputs: M_current (PIP2-dependent Kv7.2/3 membrane current), M_current norm (normalized version), PIP2_PH_T (PH probes bound to PIP2), PIP2_PH_T Norm (normalized version). M_current calculated as $(KCNQ + PIP2_KCNQ) \times 0.5 \times 0.055 \times PIP2_KCNQ^2$. Normalized outputs scaled to 0–1 range based on min/max model responses. θ_{1-3} represents the model parametrization after the first, second or third optimization. Experimental data and standard deviation (data_SD) from Jensen *et al.* [440].

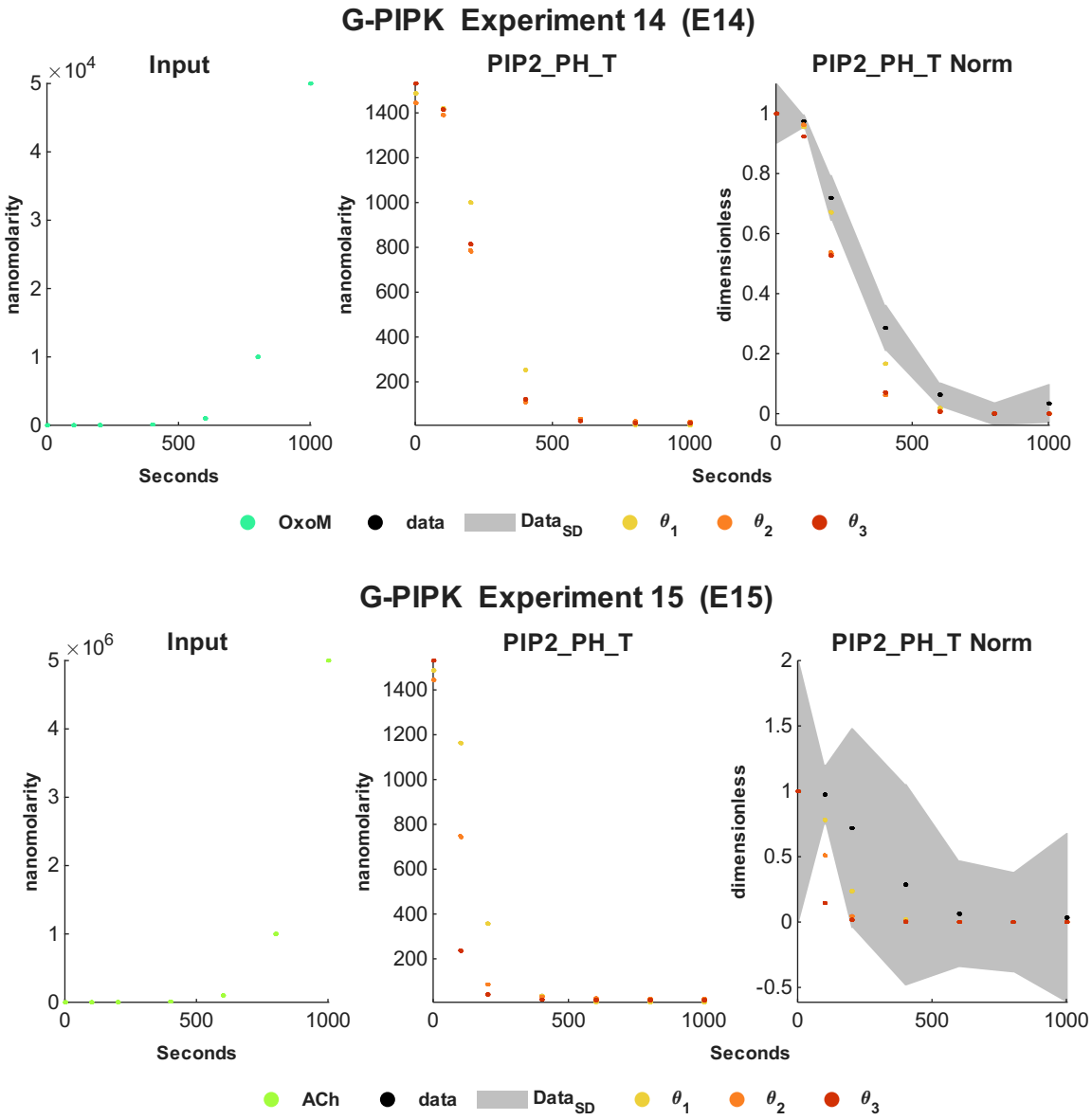


Figure D.8: G-PIPK $E_{14,15}$ diagnostics plot. Shows the input, output data with standard deviation, and θ_{1-3} outputs for G-PIPK $E_{14,15}$. Inputs: multiple impulses of OxoM for E_{14} and ACh for E_{15} at varying concentrations (1×10^0, 1×10^1, 1×10^2, 1×10^3, 1×10^4, 5×10^4 nMol/L for OxoM, 100 times more for ACh) to represent the influence of different input amounts. Outputs: PIP2_PH_T (PH probes bound to PIP2), PIP2_PH_T Norm (normalized version). Normalized outputs scaled to 0–1 range based on min/max model responses. θ_{1-3} represents the model parametrization after the first, second or third optimization. Experimental data and standard deviation (data_SD) from Jensen *et al.* [440].

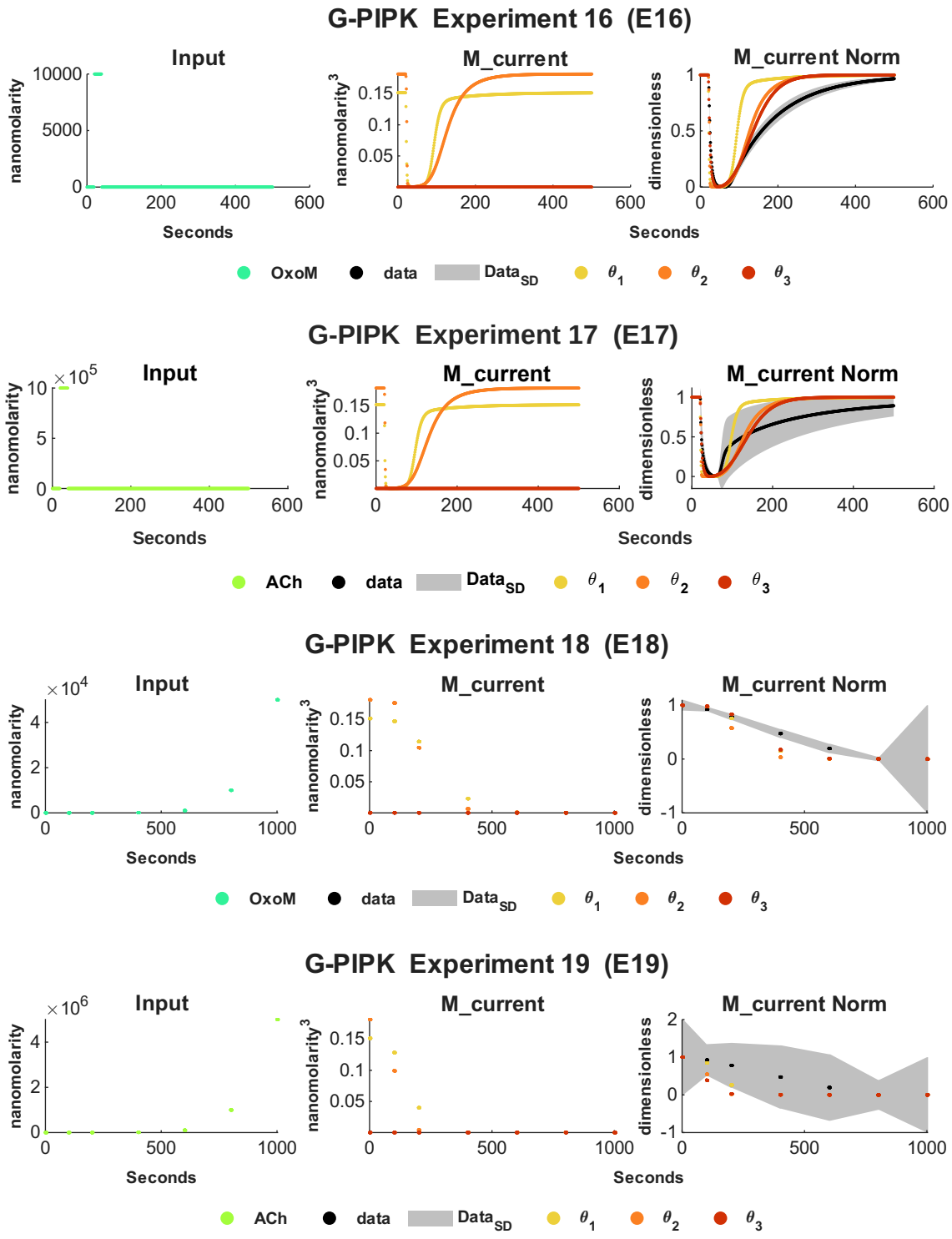


Figure D.9: G-PIPK $E_{16–19}$ diagnostics plot. Shows the input, output data with standard deviation, and $\theta_{1–3}$ outputs for G-PIPK $E_{16–19}$. Inputs: 20-second impulse of OxoM (1×10^4 nMol/L) for E_{16} and ACh (1×10^6 nMol/L) for E_{17} from 20 to 40 seconds. Multiple impulses of OxoM for E_{18} and ACh for E_{19} at varying concentrations (1×10^0, 1×10^1, 1×10^2, 1×10^3, 1×10^4, 5×10^4 nMol/L for OxoM, 100 times more for ACh) to represent the influence of different input amounts. Outputs: M_current (PIP2-dependent Kv7.2/3 membrane current), M_current norm (normalized version). M_current calculated as $(KCNQ + PIP2_KCNQ) \times 0.5 \times 0.055 \times PIP2_KCNQ^2$. Normalized outputs scaled to 0–1 range based on min/max model responses. $\theta_{1–3}$ represents the model parametrization after the first, second or third optimization. Experimental data and standard deviation (data_SD) from Jensen *et al.* [440].

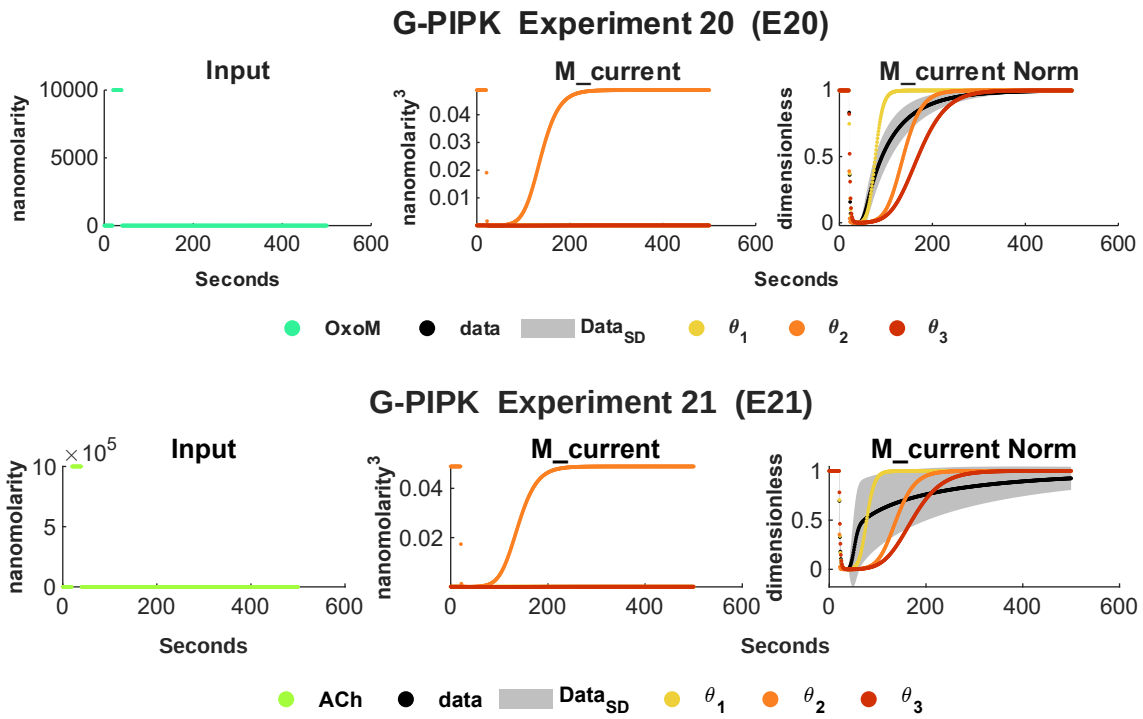


Figure D.10: G-PIPK $E_{20,21}$ diagnostics plot. Shows the input, output data with standard deviation, and θ_{1-3} outputs for G-PIPK $E_{20,21}$. Inputs: 20-second impulse of OxoM (1×10^4 nMol/L) for E_{20} and ACh (1×10^6 nMol/L) for E_{21} from 20 to 40 seconds. Outputs: M_current (PIP2-dependent Kv7.2/3 membrane current), M_current norm (normalized version). M_current calculated as $(KCNQ + PIP2_KCNQ) \times 0.5 \times 0.055 \times PIP2_KCNQ^2$. Normalized outputs scaled to 0–1 range based on min/max model responses. θ_{1-3} represents the model parametrization after the first, second or third optimization. Experimental data and standard deviation (data_SD) from Jensen *et al.* [440].

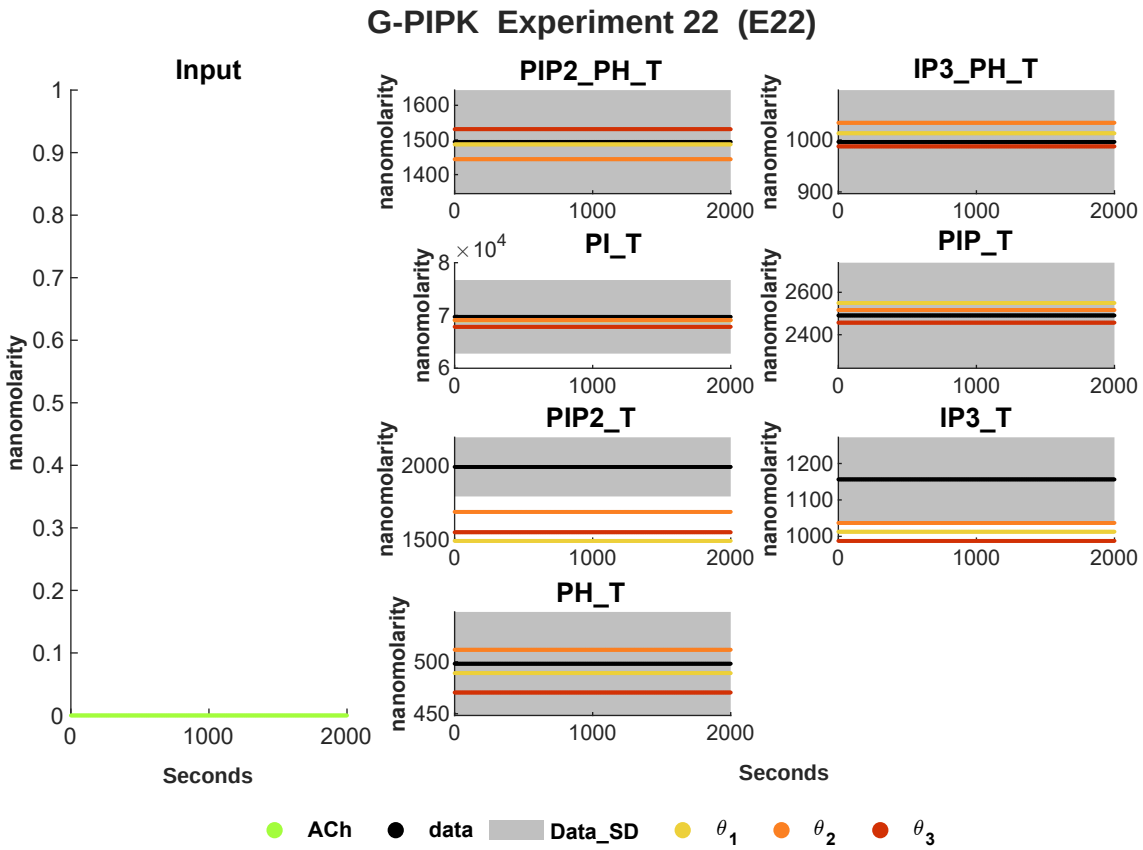


Figure D.11: G-PIPK E_{22} diagnostics plot. Shows the input, output data with standard deviation, and θ_{1-3} outputs for G-PIPK E_{22}. Input: No input is given to the model, for compatibility reasons this is shown as a ACh at 0 at the various measured times. Outputs: PIP2_PH_T (sum of all PIP2 with PH probes), IP3_PH_T (sum of all IP3 with PH probes), PI_T (sum of all PI), PIP_T (sum of all PIP), PIP2_T (sum of all PIP2), IP3_T (sum of all IP3), PH_T (sum of all PH probes). Normalized outputs scaled to 0–1 range based on min/max model responses. θ_{1-3} represents the model parametrization after the first, second or third optimization. Experimental data and standard deviation (data_SD) from Jensen *et al.* [440].

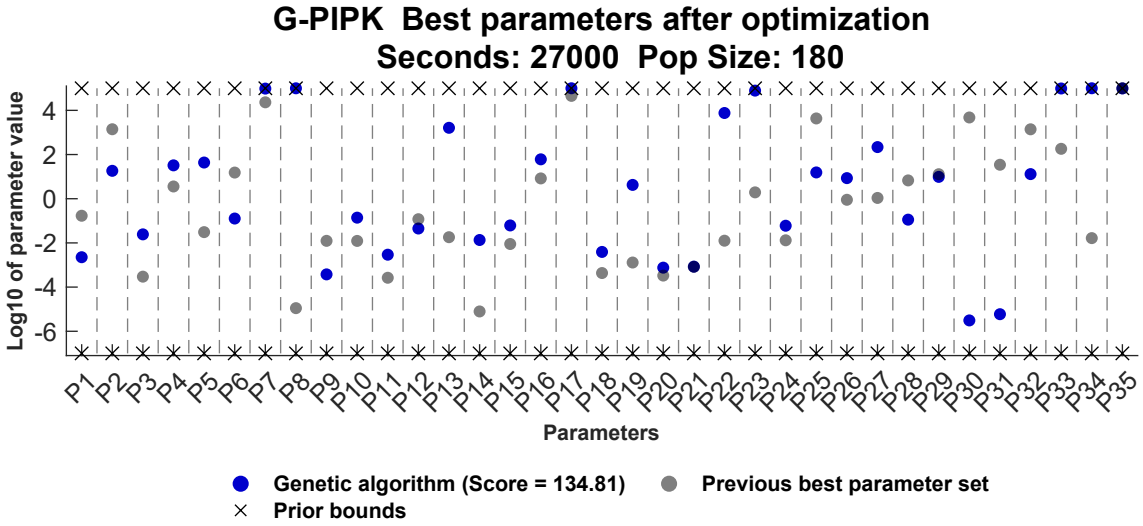


Figure D.12: G-PIPK second optimized θ. This figure illustrates the results of parameter optimization for the G-PIPK model, comparing outcomes from two rounds of estimation. Each vertical line represents one of the model's free parameters (P_{1-35}). Blue dots show the best parameter values (θ) obtained from the second parameter estimation using the genetic algorithm, which achieved a score of 134.81. Grey dots indicate the best parameter values from the first optimization round. Black crosses denote the upper and lower bounds for each parameter. The y-axis represents the $\log_{10}$ of parameter values, allowing comparison across different scales. This is the best of 216 optimizations that were computationally intensive, running for (27000×6) seconds (45 hours, run on an Intel i9-10980XE processor)) with a population size of 180 in the genetic algorithm for each of the optimizations. The figure enables a direct comparison of how parameter estimates changed between optimization round 1 and 2.

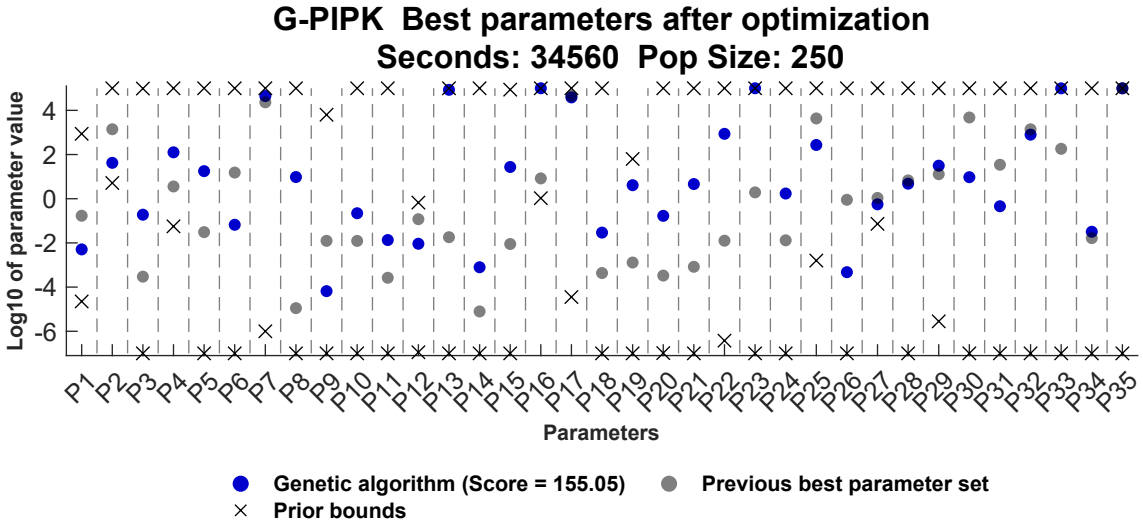


Figure D.13: G-PIPK third optimized θ. This figure illustrates the results of parameter optimization for the G-PIPK model, comparing outcomes from two rounds of estimation. Each vertical line represents one of the model's free parameters ($P_{1–35}$). Blue dots show the best parameter values (θ) obtained from the third parameter estimation using the genetic algorithm, which achieved a score of 155.05. Grey dots indicate the best parameter values from the first optimization round. Black crosses denote the upper and lower bounds for each parameter, as constrained by the results of the first Profile Likelihood Analysis (PLA). The y-axis represents the $\log_{10}$ of parameter values, allowing comparison across different scales. This is the best of 360 optimizations that were computationally intensive, running for around 4 days (run on an Intel i9-10980XE processor) with a population size of 250 in the genetic algorithm for each of the optimizations. The figure enables a direct comparison of how parameter estimates changed between optimization round 1 and 3.

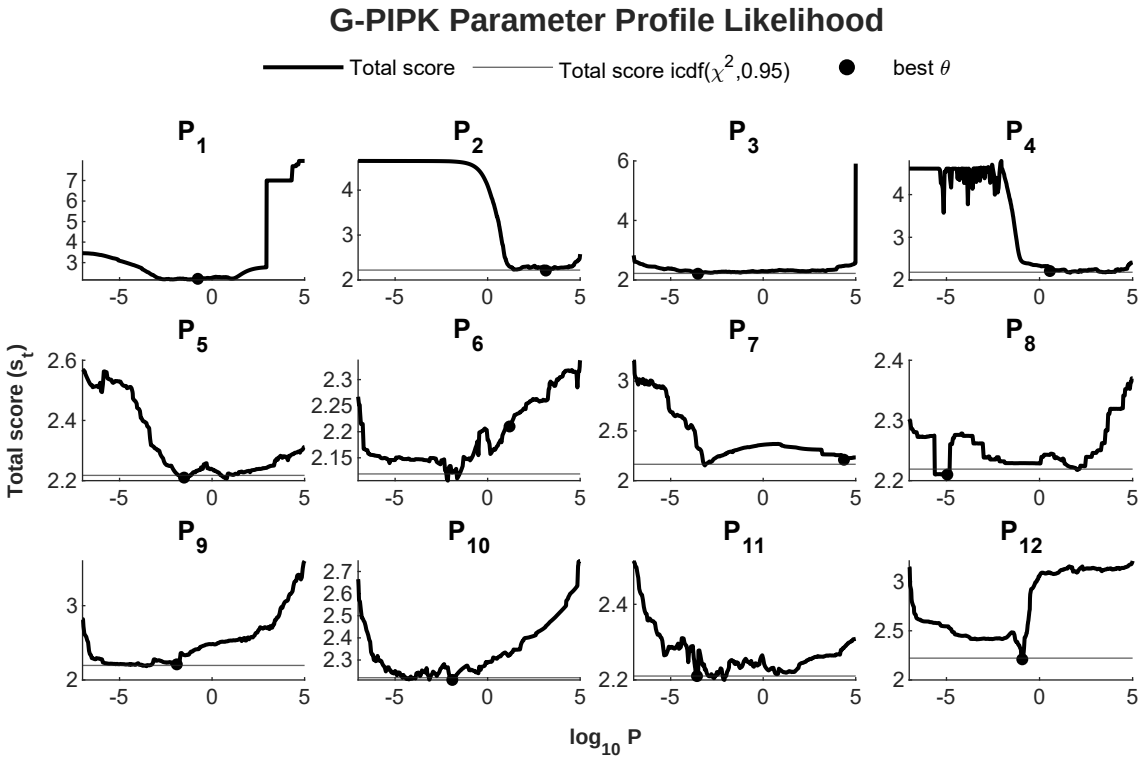


Figure D.14: G-PIPK Profile Likelihood Analysis for all Es P_{1-12}. The x-axis represents the log-transformed parameter values (log10 P) and the y-axis depicts the total score (s_t). The confidence regions are delimited by the intersection of the thick and thin black lines, with the region falling below the horizontal dashed threshold. Note that this region can be discontinuous. The thick black line represents $-2PL\left(\theta_j\right)$, which correspond to the scores being measured as shown in equation (3.25). The horizontal thin line represents the critical threshold, calculated as $\chi^2_{1;0.95} - 2LL(\hat{\theta})$, derived from the maximum log-likelihood value ($\hat{\theta}$) obtained with the model score, adjusted by the 95th percentile of the χ^2 distribution, reflecting the 95% confidence level and 1 degree of freedom as shown in equations (3.26) and (3.27). The black point is $-2LL\left(\hat{\theta}\right)$, which represents the best log-likelihood score obtained during parameter estimation, as shown in equation (3.24).

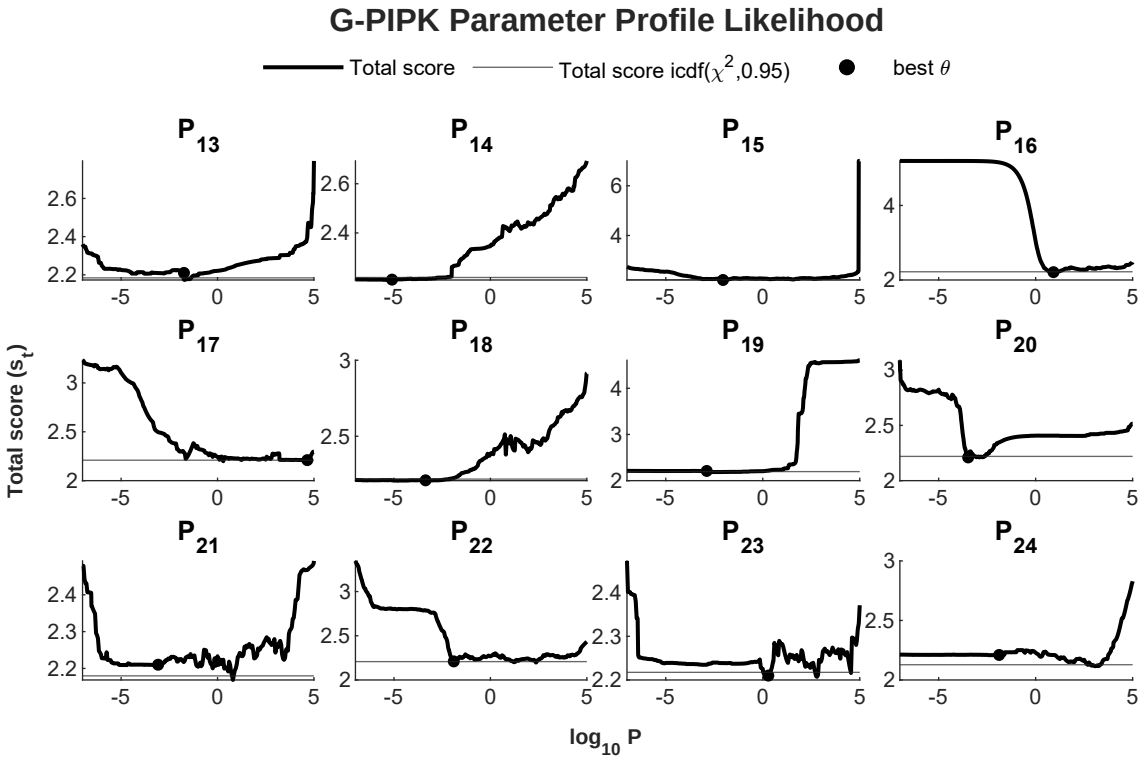


Figure D.15: G-PIPK Profile Likelihood Analysis for all Es $P_{13–24}$. The x-axis represents the log-transformed parameter values (log10 P) and the y-axis depicts the total score (s_t). The confidence regions are delimited by the intersection of the thick and thin black lines, with the region falling below the horizontal dashed threshold. Note that this region can be discontinuous. The thick black line represents $-2PL\left(\theta_j\right)$, which correspond to the scores being measured as shown in equation (3.25). The horizontal thin line represents the critical threshold, calculated as $\chi^2_{1;0.95} - 2LL(\hat{\theta})$, derived from the maximum log-likelihood value ($\hat{\theta}$) obtained with the model score, adjusted by the 95th percentile of the χ^2 distribution, reflecting the 95% confidence level and 1 degree of freedom as shown in equations (3.26) and (3.27). The black point is $-2LL\left(\hat{\theta}\right)$, which represents the best log-likelihood score obtained during parameter estimation, as shown in equation (3.24).

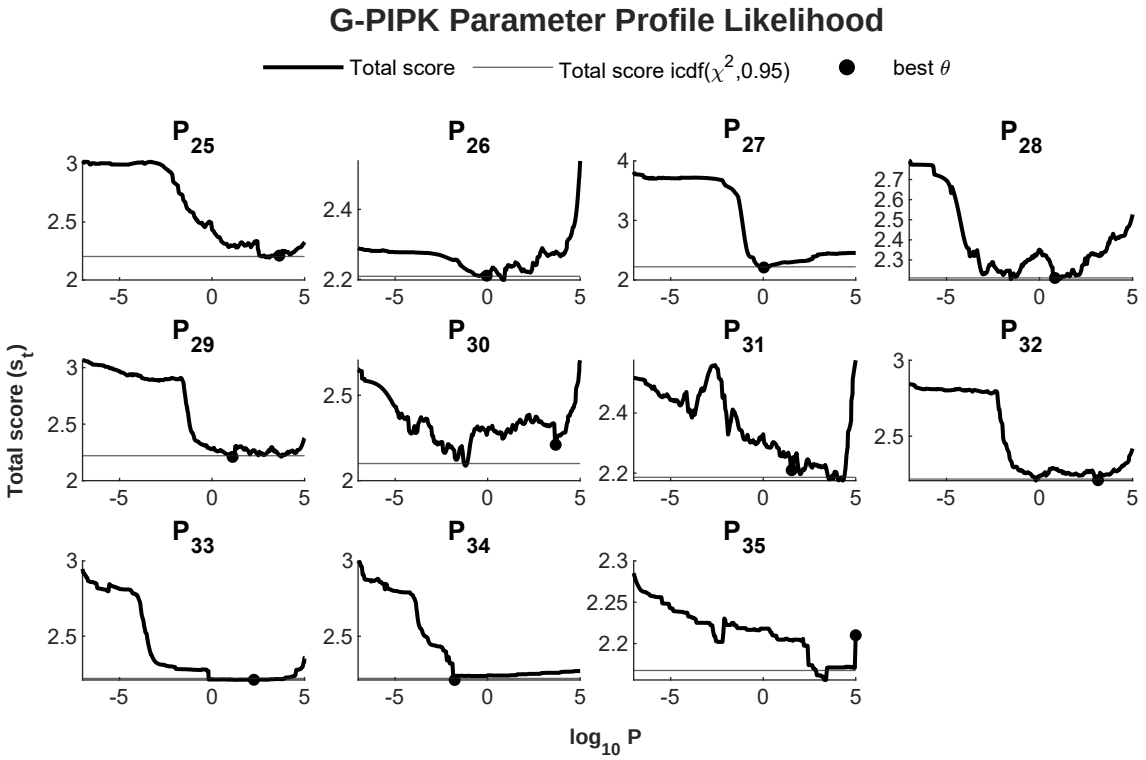


Figure D.16: G-PIPK Profile Likelihood Analysis for all Es $P_{25–35}$. The x-axis represents the log-transformed parameter values (log10 P) and the y-axis depicts the total score (s_t). The confidence regions are delimited by the intersection of the thick and thin black lines, with the region falling below the horizontal dashed threshold. Note that this region can be discontinuous. The thick black line represents $-2PL\left(\theta_j\right)$, which correspond to the scores being measured as shown in equation (3.25). The horizontal thin line represents the critical threshold, calculated as $\chi^2_{1;0.95} - 2LL(\hat{\theta})$, derived from the maximum log-likelihood value ($\hat{\theta}$) obtained with the model score, adjusted by the 95th percentile of the χ^2 distribution, reflecting the 95% confidence level and 1 degree of freedom as shown in equations (3.26) and (3.27). The black point is $-2LL\left(\hat{\theta}\right)$, which represents the best log-likelihood score obtained during parameter estimation, as shown in equation (3.24).

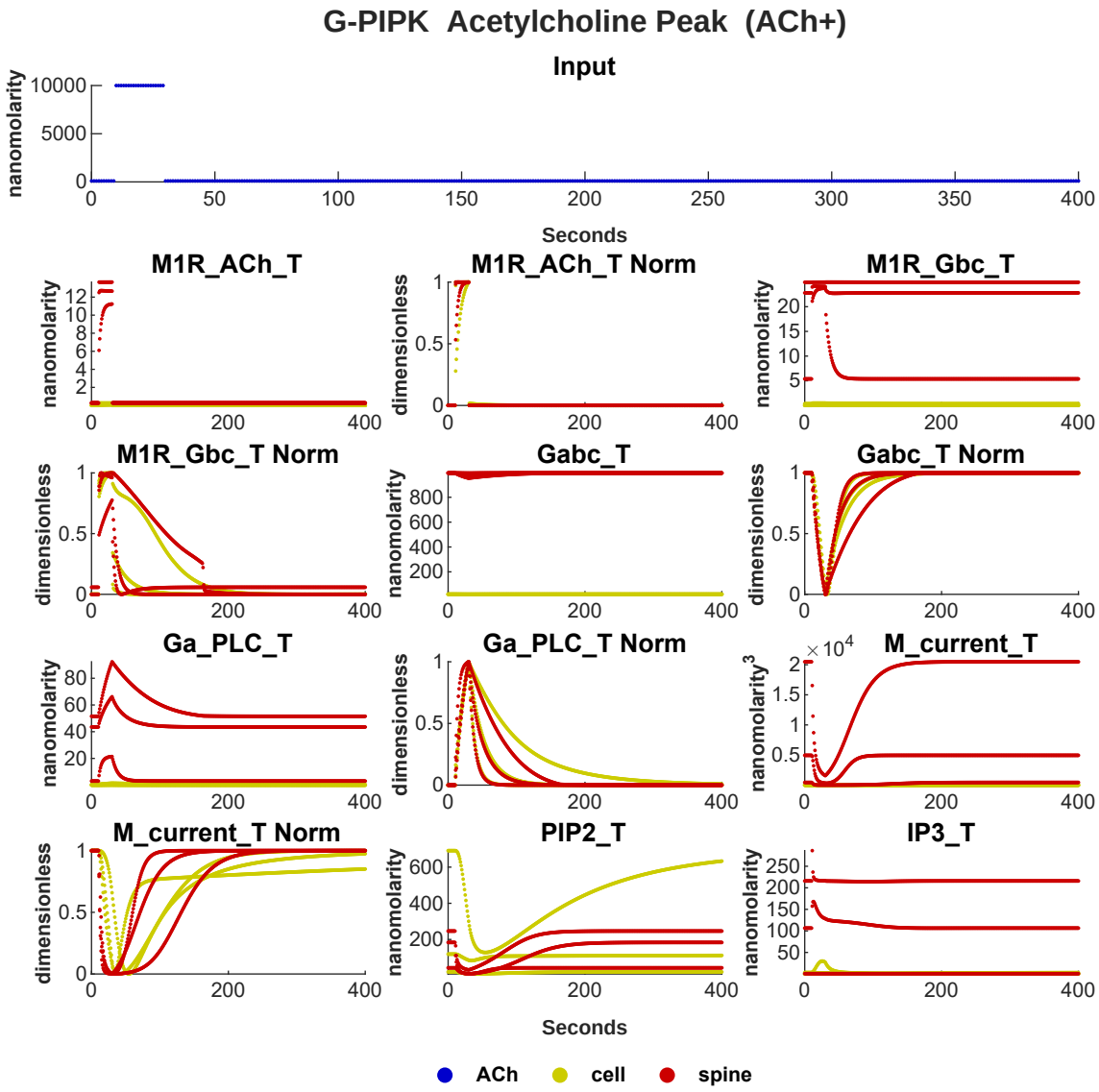


Figure D.17: G-PIPK prediction with ACh peak. The top graph shows the input (100 nM ACh with a peak of 10 mM between 10 and 30 seconds), while the bottom panels depict various outputs from the model: the total amount of M1R bound to ACh (M1R_ACh_T), and its normalised version (M1R_ACh_T_Norm), the total amount of M1R bound to $G_{\beta\gamma}$ (M1R_Gbc_T), and its normalised version (M1R_Gbc_T_Norm), the total amount of G-protein trimer concentration (Gabc_T), and its normalised version (Gabc_T_Norm), the total amount of G_{α} bound to PLC (Ga_PLC_T), and its normalised version (Ga_PLC_T_Norm), the PIP_2-dependent Kv7.2/3 membrane current (M_current), and its normalised version (M_current_Norm), the total amount of PIP_2 (PIP2_T) and the total amount of IP_3 (IP_3_T). The simulation was performed using initial conditions for cell and spine, as described in section 4.5.7 and table 4.9, with three optimized θ values.

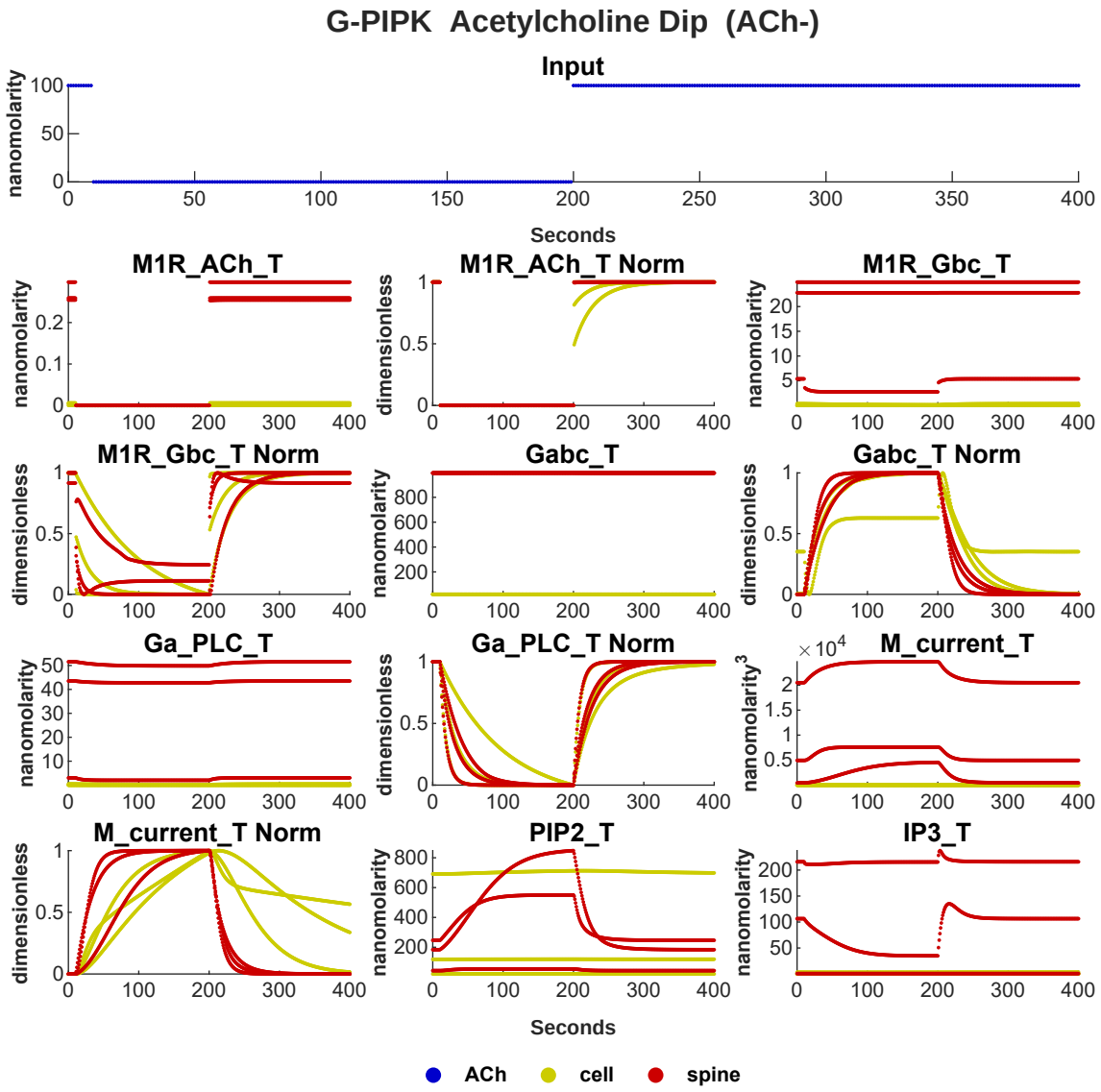


Figure D.18: G-PIPK prediction with ACh dip. The top graph shows the input (100 nM ACh with a dip of 0 M between 10 and 200 seconds), while the bottom panels depict various outputs from the model: the total amount of M1R bound to ACh (M1R_ACh_T), and its normalised version (M1R_ACh_T_Norm), the total amount of M1R bound to $G_{\beta\gamma}$ (M1R_Gbc_T), and its normalised version (M1R_Gbc_T_Norm), the total amount of G-protein trimer concentration (Gabc_T), and its normalised version (Gabc_T_Norm), the total amount of G_{α} bound to PLC (Ga_PLC_T), and its normalised version (Ga_PLC_T_Norm), the PIP_2-dependent Kv7.2/3 membrane current (M_current), and its normalised version (M_current_Norm), the total amount of PIP_2 (PIP2_T) and the total amount of IP_3 (IP_3_T). The simulation was performed using initial conditions for cell and spine, as described in section 4.5.7 and table 4.9, with three optimized θ values.

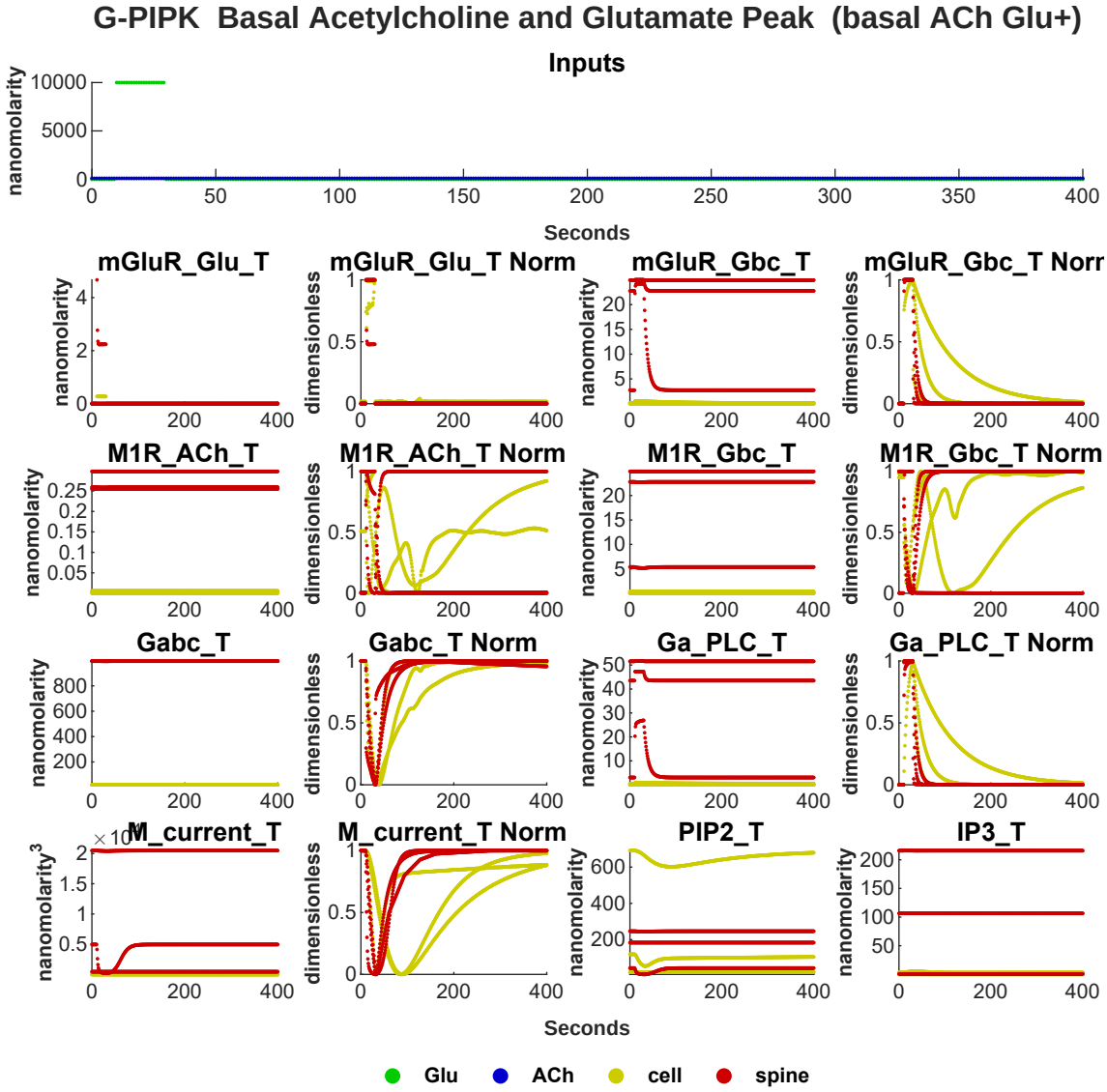


Figure D.19: G-PIPK prediction with basal ACh plus glutamate peak. The top graph shows the input, 100 nM ACh and 10 mM of glutamate between 10 and 30 seconds, while the bottom panels depict various outputs from the model: the total amount of MGluR bound to Glu (MGluR_Glu_T), and its normalised version (MGluR_Glu_T_Norm), the total amount of MGluR bound to $G_{\beta\gamma}$ (MGluR_Gbc_T), and its normalised version (MGluR_Gbc_T_Norm), the total amount of M1R bound to ACh (M1R_ACh_T), and its normalised version (M1R_ACh_T_Norm), the total amount of M1R bound to $G_{\beta\gamma}$ (M1R_Gbc_T), and its normalised version (M1R_Gbc_T_Norm), the total amount of G-protein trimer concentration (Gabc_T), and its normalised version (Gabc_T_Norm), the total amount of G_{α} bound to PLC (Ga_PLC_T), and its normalised version (Ga_PLC_T_Norm), the PIP_2-dependent Kv7.2/3 membrane current (M_current), and its normalised version (M_current_Norm), the total amount of PIP_2 (PIP2_T) and the total amount of IP_3 (IP_3_T). The simulation was performed using initial conditions for cell and spine, as described in section 4.5.7 and table 4.9, with three optimized θ values.

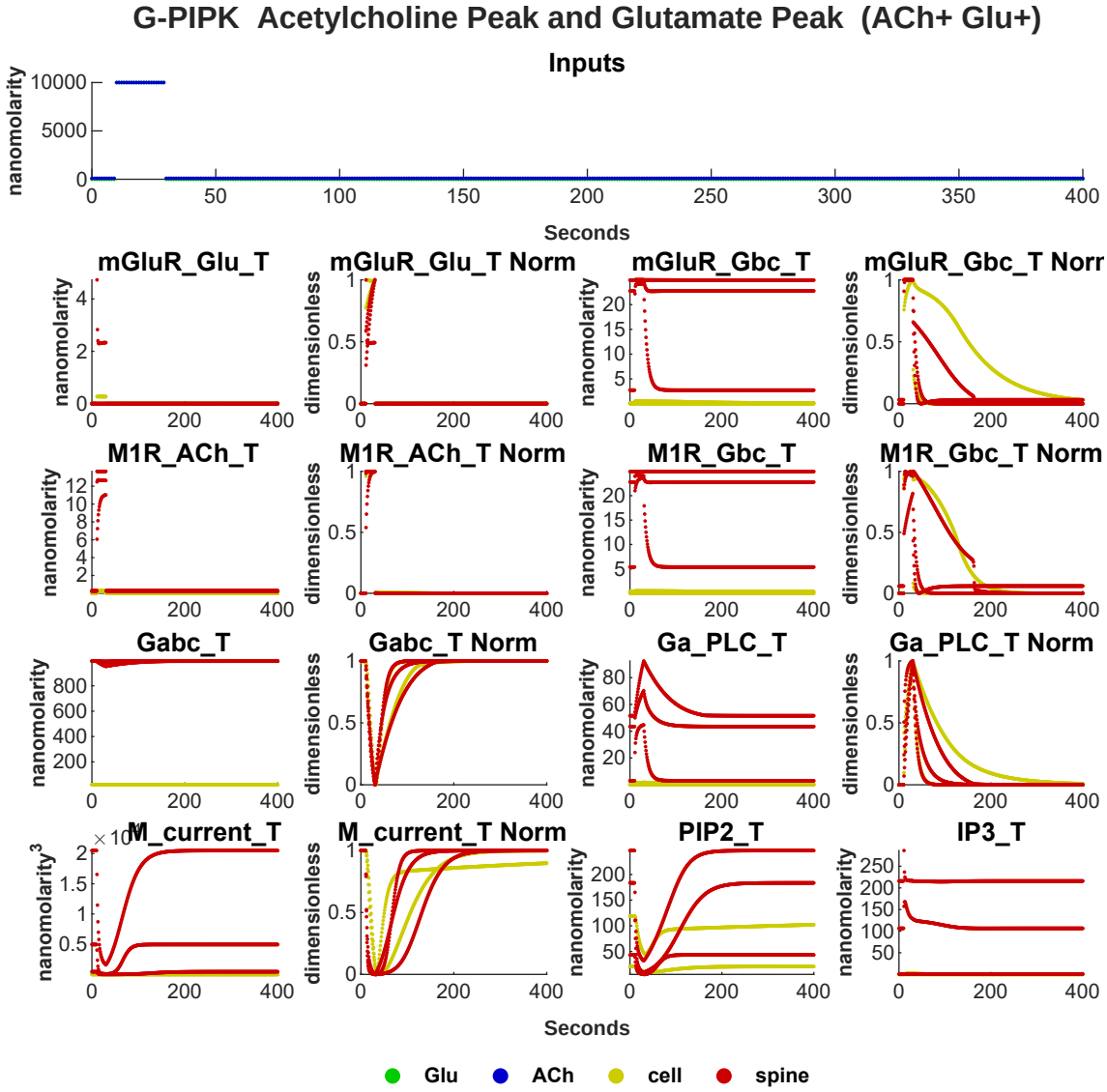


Figure D.20: G-PIPK prediction with ACh peak plus glutamate peak. The top graph shows the input, 100 nM ACh with a peak of 10 mM between 10 and 30 second and 10 mM of glutamate between 10 and 30 seconds, while the bottom panels depict various outputs from the model: the total amount of MGluR bound to Glu (MGluR_Glu_T), and its normalised version (MGluR_Glu_T_Norm), the total amount of MGluR bound to $G_{\beta\gamma}$ (MGluR_Gbc_T), and its normalised version (MGluR_Gbc_T_Norm), the total amount of M1R bound to ACh (M1R_ACh_T), and its normalised version (M1R_ACh_T_Norm), the total amount of M1R bound to $G_{\beta\gamma}$ (M1R_Gbc_T), and its normalised version (M1R_Gbc_T_Norm), the total amount of G-protein trimer concentration (Gabc_T), and its normalised version (Gabc_T_Norm), the total amount of G_{α} bound to PLC (Ga_PLC_T), and its normalised version (Ga_PLC_T_Norm), the PIP_2-dependent Kv7.2/3 membrane current (M_current), and its normalised version (M_current_Norm), the total amount of PIP_2 (PIP2_T) and the total amount of IP_3 (IP_3_T). The simulation was performed using initial conditions for cell and spine, as described in section 4.5.7 and table 4.9, with three optimized θ values.

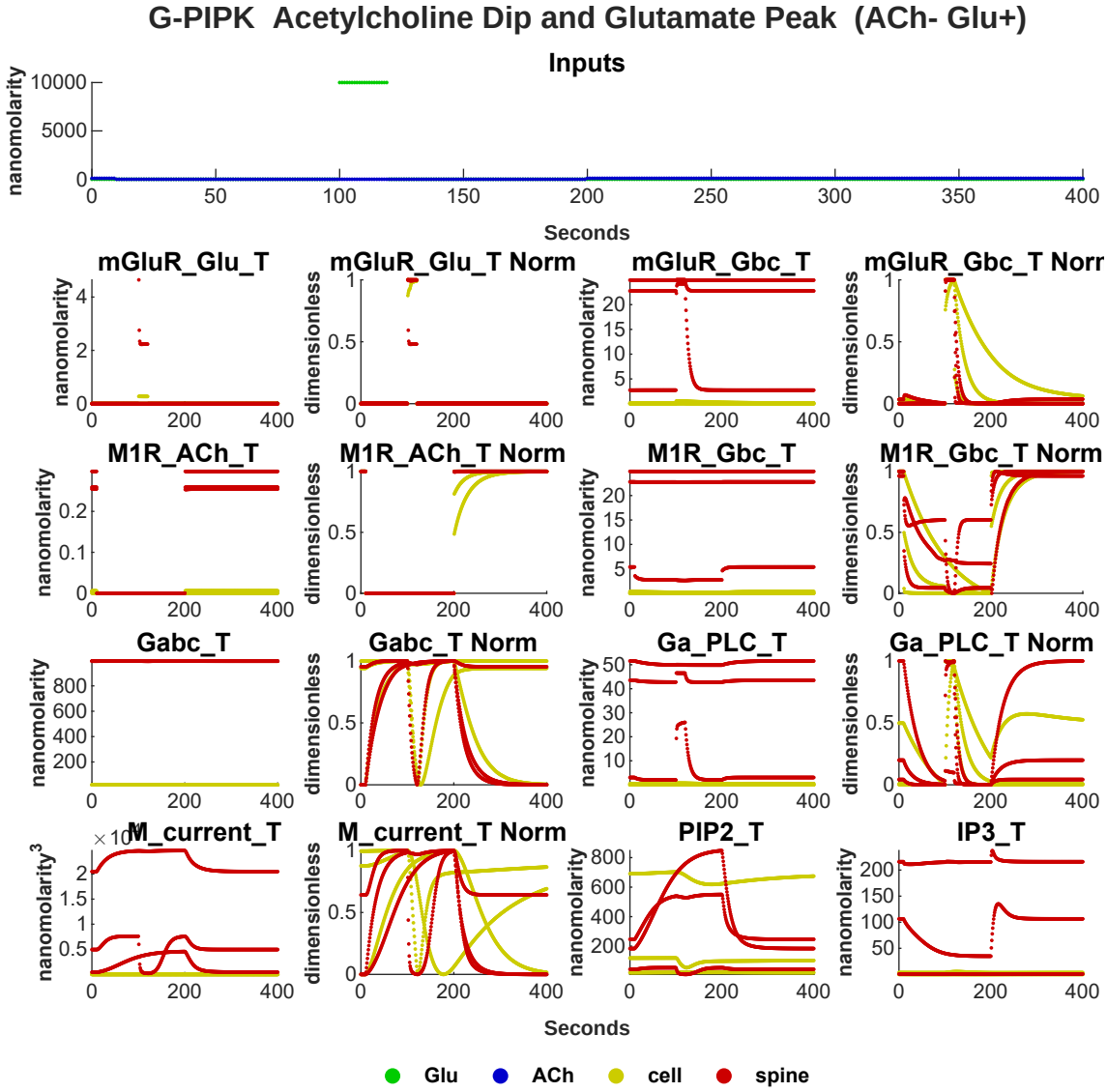


Figure D.21: G-PIPK prediction with ACh dip plus glutamate peak. The top graph shows the input, 100 nM ACh with a dip of 0 M between 10 and 200 seconds and 10 mM of glutamate between 10 and 30 seconds, while the bottom panels depict various outputs from the model: the total amount of MGluR bound to Glu (MGluR_Glu_T), and its normalised version (MGluR_Glu_T_Norm), the total amount of MGluR bound to $G_{\beta\gamma}$ (MGluR_Gbc_T), and its normalised version (MGluR_Gbc_T_Norm), the total amount of M1R bound to ACh (M1R_ACh_T), and its normalised version (M1R_ACh_T_Norm), the total amount of M1R bound to $G_{\beta\gamma}$ (M1R_Gbc_T), and its normalised version (M1R_Gbc_T_Norm), the total amount of G-protein trimer concentration (Gabc_T), and its normalised version (Gabc_T_Norm), the total amount of G_{α} bound to PLC (Ga_PLC_T), and its normalised version (Ga_PLC_T_Norm), the PIP_2-dependent Kv7.2/3 membrane current (M_current), and its normalised version (M_current_Norm), the total amount of PIP_2 (PIP2_T) and the total amount of IP_3 (IP_3_T). The simulation was performed using initial conditions for cell and spine, as described in section 4.5.7 and table 4.9, with three optimized θ values.