

Modelling the Intracellular Environment under Uncertainty, from Post-translational Modification to Metabolic Networks

Présentée le 11 juin 2021

Faculté des sciences de base
Laboratoire de biotechnologie computationnelle des systèmes
Programme doctoral en chimie et génie chimique

pour l'obtention du grade de Docteur ès Sciences

par

Robin Alexander DENHARDT-ERIKSSON

Acceptée sur proposition du jury

Dr A.-S. Chauvin, présidente du jury
Prof. V. Hatzimanikatis, directeur de thèse
Dr L. Hutchinson, rapporteuse
Dr M. Machacek, rapporteur
Prof. D. Iber, rapporteuse

ABSTRACT

Biological systems are complex and often only partially observable. These systems also often exhibit emergent behaviours which are central to their functioning. One of these is regulation, how a system reacts to its environment in a dynamic manner. In the case of biological systems the aim of regulation is often to ensure proper functioning and ultimately survival. By better understanding how the basic building blocks of biological systems fit together to create these emergent behaviours we can tune them to our advantage or fix them when they malfunction. For example, we may try to optimise the internal layout of cells that produce insulin for medical use, or try to block pathogens from infecting healthy cells. A hurdle to studying these systems is that experimental methods are often only able to measure certain aspects such as relative changes in the amounts of proteins. Because of these limitations it isn't possible to exhaustively probe every detail of these systems, only specific readouts.

This becomes a problem when trying to come up with a hypothesis to better describe the system in question. Because it is only possible to measure a limited number of aspects, it is not straightforward to integrate their interaction with the larger system and with each other. Without a mechanistic description of how the system works it is not possible to holistically understand its functioning nor its response to new perturbations. In other words, there is data but no knowledge has been created from it.

In this thesis we will show that using the partial information from experimental data combined with mathematical modelling, we can still extract knowledge of how these systems work. We will incorporate expert knowledge from the appropriate fields as well as multiple different types of experimental data in order to create models which provide a holistic understanding of how specific proteins are regulated.

We will demonstrate that mathematical models of Post-Translational Modification (PTM) can elucidate how this mechanism regulates various proteins. Given a set of dynamic time-series experiments and relative changes in protein levels, we will propose a tailored workflow that incorporates these data into a model. This model can then be used as a tool for understanding the mechanisms by which cells regulate their proteins. Creating a model makes it possible to place multiple sources of experimental data into a single unified system, as well as predicting how the protein would react to conditions which are not experimentally feasible.

A second aspect we will deal with is that of uncertainty in kinetic models. We will use Global Sensitivity Analysis (GSA) in order to better understand how different parameters affect biological behaviours. For example, how

multiple parameters interact and influence the half-life of a protein. We will then also apply this method to large scale models of metabolism. In this case, we will show that GSA can be used as a method to guide experimental efforts in order to refine model predictions while measuring a minimal number of parameters.

ZUSAMMENFASSUNG

Biologische Systeme sind komplex und oft nur teilweise beobachtbar. Diese Systeme zeigen oft Verhalten, welche sich nicht auf die einzelnen Komponenten zurück führen lässt sondern kollektive oder »emergente« Eigenschaften des gesamten Systems sind. Solche emergenten Eigenschaften sind von zentraler Bedeutung das Funktionieren dieser biologischen Systeme. Eine davon ist die Regulation des dynamischen Verhaltens nach dem das System durch Umwelt einflüsse beeinflusst wurde. Im Falle biologischer Systeme besteht das Ziel der Regulation oft darin, das ordnungsgemäße Funktionieren und damit letztlich das Überleben zu sichern. Wenn wir besser verstehen, wie die Grundbausteine biologischer Systeme zusammenpassen, um diese emergente Verhaltensweisen zu erzeugen, können wir sie zu unserem Vorteil abstimmen oder bei Störungen reparieren.

Wir können zum Beispiel versuchen, die innere Anordnung von Zellen optimieren, die Insulin für die medizinische Verwendung produzieren, oder Krankheitserreger daran hindern, gesunde Zellen zu infizieren. Eine Hürde bei der Untersuchung dieser Systeme besteht darin, dass experimentelle Methoden oft nur bestimmte Größen messen können, wie z.B. relative Veränderungen in den Mengen der Proteine. Aufgrund dieser Einschränkungen ist es nicht möglich, sämtliche Aspekte solcher Systeme erschöpfend zu untersuchen. Es können nur bestimmte Messwerte erfasst werden.

Dies wird zum Problem sobald versucht wird, eine Hypothese zur besseren Beschreibung des betreffenden Systems aufzustellen. Da es nur möglich ist, eine begrenzte Anzahl von physikalischen Größen zu messen, ist es schwierig die Interaktion von Teilsystemen mit dem gesamten System und untereinander zu verstehen. Ohne eine mechanistische Beschreibung der Funktionsweise des Systems ist es nicht möglich, seinen Funktionieren und seine Reaktion auf neue Störungen ganzheitlich zu verstehen. Anders ausgedrückt, obschon Daten erfasst wurden, haben diese noch zu keinem besseren Verständnis beigetragen.

In dieser Arbeit zeigen wir auf, dass es möglich ist, aus einer Kombination von Teilinformationen aus experimentellen Daten und mathematischer Modellierung ein Verständnis der Funktionsweise dieser Systeme zu gewinnen. Wir ziehen Expert*innenwissen aus den entsprechenden Bereichen und eine Vielzahl verschiedener Arten von experimentellen Daten bei, um Modelle zu erstellen, die mittels eines ganzheitlichen Verständnis aufzeigen, wie bestimmte Proteine reguliert werden.

Wir zeigen auf, dass mathematische Modelle des Posttranslationale Modifikation (PTM) erklären können, wie dieser Mechanismus verschiedene Proteine reguliert. Ausgehend von einer Reihe dynamischer Zeitreihenexperi-

mente und Angaben zu relativen Veränderungen der Proteinmengen schlagen wir einen maßgeschneiderten Arbeitsablauf vor, der diese Daten in ein Modell einbezieht. Dieses Modell kann dann als Werkzeug zum Verständnis der Mechanismen verwendet werden, mit denen Zellen ihre Proteine regulieren. Die Erstellung eines Modells macht es möglich, Daten aus verschiedenen Experimenten in ein ganzheitliche System zu integrieren und damit prediktive Aussagen bezüglich der Reaktionen von Proteinen auf spezifische Bedingungen zu machen, die auf diese Art nicht in Experimenten umgesetzt werden könne.

Des Weiteren befassen wir uns in dieser Arbeit mit der Unsicherheit in kinetischen Modellen. Wir werden eine Globale Sensitivitäts Analyse (GSA) durchführen, um besser zu verstehen, wie verschiedene Parameter das biologische Verhalten beeinflussen. Diese Methode macht es z.B. möglich zu verstehen, wie der Interaktion verschiedener Parameter die Halbwertszeit eines Proteins beeinflusst. Wir wenden diese Methode auch auf großmaßstäbliche Modelle des Stoffwechsels an. In diesem Fall zeigen wir auf, dass die GSA als Methode im Aufbau von Experimenten eingesetzt werden kann, um so die Modellvorhersagen bei gleichzeitiger Messung einer minimalen Anzahl von Parametern zu verfeinern.

ACKNOWLEDGEMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 686070.

Thank you to Professor Vassily Hatzimanikatis for warmly welcoming me into the LCSB, as well as his time and guidance.

Thank you to Professor Gisou Van der Goot for the stimulating exchanges and collaboration on building models of protein palmitoylation.

Thank you to the jury for taking the time to review this thesis.

Thanks to the LCSB family for the superb research environment.

CONTENTS

List of Figures	x
List of Tables	xxv
List of Acronyms	xxvi
1 INTRODUCTION	1
1.1 Systems Biology	1
1.2 Cell Regulation and Signalling	1
1.3 Studying Palmitoylation in cells	3
1.4 Mathematical Modelling of Palmitoylation	3
1.5 Uncertainty in Kinetic Models of Metabolism	6
1.6 Thesis Overview	7
2 CALNEXIN	8
2.1 Reaction kinetics	8
2.1.1 Chemical Master Equation	8
2.1.2 Law of Mass action	10
2.1.3 Enzyme kinetics	11
2.2 Experimental data	13
2.2.1 Palmitoylation Background	13
2.2.2 Radiolabelling	14
2.2.3 siRNA	16
2.2.4 Mutants	17
2.2.5 Cell Lines	17
2.2.6 Fluorescence Microscopy	18
2.3 Model structure	18
2.3.1 Calnexin Background	18
2.3.2 Model overview	20
2.3.3 Calibration and Results	22
2.4 Global Sensitivity Analysis	22
2.4.1 Introduction to Sensitivity Analysis	23
2.4.2 Sobol Sensitivity indices	24
2.4.3 Computing Sobol Sensitivity indices	26
2.4.4 Sampling Methods	28
2.4.5 Interpreting the Sobol indices	32
2.4.6 Application to the Calnexin model	33
2.5 Conclusion and Future Directions	40
3 CLIMP63	43
3.1 CLiMP63 Background	43
3.2 Experimental data	44
3.2.1 Cross-talk with Phosphorylation	44
3.2.2 Identifying palmitoylating enzymes	45

3.2.3	Determining palmitoylated fraction	45
3.2.4	Quantifying surface population	46
3.2.5	Fluorescence Microscopy	48
3.3	Modelling goals	48
3.4	Model construction	51
3.5	Model calibration	51
3.5.1	Rule-based modelling	51
3.5.2	Initial Calibration attempts	54
3.5.3	Optimisation algorithms	55
3.5.4	Alternative model structures	57
3.5.5	Integration using Julia	59
3.6	Results	61
3.7	Discussion	63
3.8	Conclusion	72
4	CMG2	73
4.1	CMG2 Background	73
4.2	Experimental data	74
4.2.1	Identifying the palmitoylating enzymes	76
4.2.2	Characterising the third palmitoylation site	77
4.2.3	Subcellular distribution	79
4.2.4	Radiolabelling	80
4.2.5	Comparing Retinal Pigment Epithelial 1 (RPE1) and HeLa cell lines	81
4.2.6	Palmitoylation and anthrax endocytosis	83
4.3	Model Structure	83
4.4	Model Calibration	86
4.5	Results	88
4.5.1	Calibration and Validation	88
4.5.2	Biosynthetic Flux	91
4.5.3	Half-life of surface subpopulation	97
4.5.4	Effect of palmitoylation on surface subpopulation	97
4.6	Discussion	99
4.7	Conclusion	100
4.7.1	Future Outlook	101
5	GMCA	103
5.1	Abstract	103
5.2	Author summary	104
5.3	Introduction	104
5.4	Results and Discussion	106
5.4.1	Illustrative example	106
5.4.2	Kinetic model	111
5.4.3	Uncertainty in control of the pentose phosphate path- way	111
5.4.4	Future opportunities and limitations	119

5.5	Conclusions	120
5.6	Materials and methods	120
5.6.1	MCA, sampling saturations	120
5.6.2	GSA, calculating sensitivity indices	122
6	CONCLUSION	125
6.1	Future Outlook	126
6.1.1	Palmitoylation models	126
6.1.2	Uncertainty in Metabolic models	128
I	APPENDIX	129
A	ENZYME KINETICS	130
A.1	Equilibrium approximation	130
A.2	Quasi Steady-State Approximation	131
A.3	Total Quasi Steady-State Approximation	131
B	CLIMP63 MODEL	133
B.1	Model structure	133
B.2	Parameter values	141
C	CMG2 MODEL	144
C.1	Model structure	144
C.2	Parameter values	164
D	E. COLI NETWORK	168
	BIBLIOGRAPHY	170

LIST OF FIGURES

Figure 1	Illustration of a thioester bond between Cysteine and a fatty acid. The cysteine residue is in black, the thioester bond in red and the fatty acid in blue.	13
Figure 2	Illustration of a pulse-chase experiment. The labelled compound is made available through the medium at $t = 0$, for a pulse duration of about 500 a.u. The chase starts at $t = 500$ when the labelled compound is removed from the system's environment. During the pulse, the radioactive signal increases as radioactive atoms are incorporated into newly synthesized proteins. When the culture medium is changed back to being non-radioactive, newly synthesized protein are once again made with non-radioactive species. Hence the radioactive signal decreasing during the chase. Note that the plateau during the pulse phase means that all protein has been replaced with the labelled variant. Depending on the protein turnover and the pulse duration this isn't necessarily the case.	15
Figure 3	The different atoms in a palmitoylated protein which can be radiolabelled. The sulfur atom in red can be replaced by ^{35}S . The hydrogen atoms in the palmitate chain can be replaced by tritium atoms, ^3H	15
Figure 4	The amino acids Cysteine and Alanine. They are identical except for the thiol group that is absent from Alanine. By replacing a Cysteine with Alanine, it is possible to disable palmitoylation at a specific site. . .	17
Figure 5	Fluorescence microscopy image of Calnexin in HeLa cells, reproduced with permission from Sandoz <i>et al.</i> [74]. The nuclei can be seen as grey oval shapes around which Calnexin, in the Endoplasmic Reticulum (ER), is much whiter and extends towards the cell periphery.	19

Figure 6	Molecular Dynamics (MD) simulations showing the effect of palmitoylation on the conformation of Calnexin, figure reproducing with permission from Lakkaraju <i>et al.</i> [44]. The orientation of the cytosolic tail with respect to the transmembrane helix is palmitoylation-dependent. The effect of the two palmitoylation sites is different. Palmitoylation on both sites seems to have an additive effect.	20
Figure 7	Illustration of the structure of the Calnexin model, reproduced with permission from Dallavilla <i>et al.</i> [43]. Synthesis leads to an unfolded form, 'rCAL', which must first fold into the mature protein 'fCAL'. Either palmitoylation site can be palmitoylated first. All palmitoylation reactions are catalyzed by zDHHC6, and a corresponding Acyl-protein Thioesterases (APT) catalyzes the reverse reactions. Decay, palmitoylation and depalmitoylation all depend on palmitoylation state.	21
Figure 8	Variance propagation through a model. The variance of the input parameter is high, while the resulting variance in the model output is relatively low. This would indicate a parameter with low sensitivity. . . .	24
Figure 9	Representation of a Full Factorial design with 3 parameters and 2 levels, reproduced with permission from Saltelli <i>et al.</i> [82]	29
Figure 10	Combined Latin Hypercube (LH)/Fractional Factorial (FF) design, reproduced with permission from Saltelli <i>et al.</i> [82]. A Resolution III design is shown on the left and Resolution IV on the right. There are the same number of points in each level and corner between any two parameters (Resolution III) or any three parameters (Resolution IV).	31
Figure 11	Illustration of reaction rate v as a function of the normalised substrate concentration S/K_m . The grey line shows the effect of increasing V_{max}	34
Figure 12	Fitness scores as a function of time for Nondominated Sorting Genetic Algorithm II (NSGA-II) parameter sets and the wide sets generated in GSA. The parameter sets given in [43] were used as a reference. The solid lines show the mean, and the shaded area shows the first and third quartile. The GSA parameter sets were obtained by randomly sampling 100 parameters sets using the wide parameter bounds.	35

Figure 13	Fitness scores as a function of time for NSGA-II parameter sets and the constrained sets generated in GSA. The parameter sets given in [43] were used as a reference. The solid lines show the mean, and the shaded area shows the first and third quartile. The GSA parameter sets were obtained by randomly sampling 100 parameters sets using the constrained bounds.	36
Figure 14	In-silico pulse-chase experiment, with NSGA-II parameter sets or constrained GSA sets. The parameter sets given in [43] were used as a reference. The solid lines show the mean, and the shaded area shows the first and third quartile. The GSA parameter sets were obtained by randomly sampling 100 parameters sets using the constrained bounds.	37
Figure 15	GSA was performed on various characteristics of the Calnexin model using 16'000 samples generated with the LH/FF smapling method. The mean and standard deviation of the sensitivty indices are shown. A) First order and total effect sensitivity indices for the half-life of total ^{35}S -labelled and ^3H -labelled Calnexin. Constrained parameter bounds. Other parameters with sensitivity indices of 0 are omitted for clarity. B) First order and total effect sensitivity indices for the steady-state concentrations of $f\text{CAL}$ and $c12\text{CAL}$. Constrained parameter bounds. C) First order and total effect sensitivity indices for the half-lives of $r\text{CAL}$, $c1\text{CAL}$ and $c12\text{CAL}$. D) First order and total effect sensitivity indices for the half-life of total ^{35}S labelled Calnexin. The difference between wide and constrained parameters bounds is shown. Results from local Sensitivity Analysis (SA) are also shown for comparison.	38
Figure 16	Representation of a possible palmitoylation network of Calnexin, including both PAT6 (zDHHC6) and its corresponding APT. Other palmitoylation targets of PAT6 are also shown	42
Figure 17	^3H labelling for two hours showing the effect of promoted microtubule stabilization using taxol, or increased microtubule depolymerization using nocodazole. Reproduced with permission from [74]	45

Figure 18	Quantification of ^3H labelling of Cytoskeleton-linking Membrane Protein 63 (CLiMP63) (N=4) done for two hours in control conditions and with silencing RNA (siRNA) targeting either zDHHc6 or zDHHc2. Mean, standard deviation and p-values (**p < 0.01, ***p < 0.001) are shown. Reproduced with permission from [74].	46
Figure 19	Schematic representation of the Acyl-RAC experimental protocol. Immunoprecipitation is used to capture the protein of interest. The palmitoylated cysteines can then be cleaved using hydroxylamine (NH_2OH). Fluorescent labelling is done using iodoacetamide-oregon-green-488 (IAA-OG488). Comparing the fluorescent readout with that of proteins that weren't cleaved with hydroxylamine makes it possible to calculate the fraction of acylated protein. Reproduced with permission from [74].	47
Figure 20	Results of the Acyl-RAC experiment on CLiMP63 (N=3), showing the estimated palmitoylated (acylated) fraction. Error bars represent the standard deviation. Reproduced with permission from [74].	47
Figure 21	Quantification of CLiMP63 population at the surface using surface biotinylation (N=3). Mean, standard deviation and p-values are shown (**p < 0.001). The effect of silencing zDHHc6 and zDHHc2 are shown relative to Control conditions. Reproduced with permission from [74].	48
Figure 22	Confocal immunofluorescence microscopy images of CLiMP63 in HeLa cells. Reproduced with permission from [74]. Control conditions and zDHHc6 knock-out are shown. The nuclei can be identified by dark circle-like shapes, around which CLiMP63 shows up as white dots.	49
Figure 23	Initial CLiMP63 model structure, containing three compartments and a total of six species (not counting the unfolded form of CLiMP63). M denotes the monomeric CLiMP63 protein, the subscript its cellular localisation and the superscript its palmitoylation state, 0 for non-palmitoylated and 1 for palmitoylated. Synthesis, folding, transport and decay are modelled according to mass action kinetics. Palmitoylation and Depalmitoylation are modelled with Total Quasi Steady-State Approximation (tQSSA) kinetics.	52

Figure 24	Subset of the calibration experiments ($N=5$) obtained after the initial calibration attempts. Experimental data points are shown in red with the error bars showing standard deviation. Model output is shown with the blue curves. The shaded areas represent the first and third quartile of the model output.	54
Figure 25	Population distribution obtained during the initial model calibration. The mean of 100 parameter sets is shown. The error bars represent the standard deviation. Both normal (Wild type (WT)) conditions and silencing zDHHC6 are shown. In both cases, the majority of CLiMP63 is at the Plasma Membrane (PM), whereas when silencing zDHHC6 almost none is present in the ER. Reproduced with permission from [74]. . .	55
Figure 26	Error function used for comparing experimental data to model output. Experimental data is shown with red dots, and is used to create a piecewise linear curve. The area between the experimental and model curves is the output of the error function. Even though there is no data between five and twenty hours in this example, using the area ensures that the last data point is properly taken into account.	56
Figure 27	Illustration of the general functioning of a Genetic Algorithm (GA) such as NSGA-II. An initial population is first sampled from the parameter space. A user-defined fitness function is used to calculate the fitness score of every member of the population. Based on their fitness scores, the members of the population "reproduce" by crossover and mutation of their parameters. Based on the characteristics of this new population and the stopping criteria the process is either repeated or stopped.	58
Figure 28	Illustration of the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) algorithm functioning. The fitness landscape is shown by white concentric circles, with the optimum being shaded a lighter colour. Members of the population are shown by black dots. The dashed orange oval represents the current direction of the algorithm, which is described by the covariance matrix. Reproduced with permission from [99].	58

Figure 29	Illustration of the multiple model structures proposed. Each modification to the monomer CLiMP ₆₃ model is highlighted in a specific colour. A recycling exosome was added to simulate membrane recycling at the PM. No reactions could occur in this compartment. A palmitoylation/depalmitoylation catalysed by zD-HHC6 and its corresponding APT was added in the Cytoplasm (CP). Palmitoylation/depalmitoylation of the unfolded CLiMP ₆₃ catalysed by zDHHC6 in the ER was added.	60
Figure 30	Final CLiMP ₆₃ model structure, containing three compartments and eight species (excluding unfolded protein). M denotes the monomeric CLiMP ₆₃ protein, the subscript its cellular localisation and the superscript its palmitoylation state, 0 for non-palmitoylated and 1 for palmitoylated. D denotes a CLiMP ₆₃ dimer, the subscript its cellular localisation and the superscript its palmitoylation state, 0 for non-palmitoylated, 1 for single palmitoylation and 2 for double palmitoylation. Compared to the initial structure, a dimer was added and transport of the palmitoylated monomer was removed. Reproduced with permission from [74].	62
Figure 31	Experimental data set that was used for calibration of the CLiMP ₆₃ model. Red dots represent the experimental data points (N=5), and their error bars the standard deviation. Solid grey lines show the median of 100 models, with the shaded area representing the first and third quartile. Reproduced with permission from [74].	64
Figure 32	Validation set of the CLiMP ₆₃ model. Red dots represent the experimental data points (N=5), and their error bars the standard deviation. Solid grey lines show the median of 100 models, with the shaded area representing the first and third quartile. Reproduced with permission from [74].	65
Figure 33	Population distribution of species in the model under normal (WT) conditions and when silencing zD-HHC6. The mean of a 100 models is shown. Error bars represent the standard deviation. Reproduced with permission from [74].	65
Figure 34	HeLa cells were ran on either Blue Native gel or Sodium dodecyl sulphate–polyacrylamide gel electrophoresis (SDS-PAGE). Two different surfactants, Digitonon and Sodium dodecyl sulfate (SDS), were used.	66

Figure 35	Deconvolution of WT ^{35}S and ^3H chases into the individual labelled species. The model was used to predict the evolution of the individual species over time. Experimental results are shown in red dots (N=5). The error bars represent the standard deviation. Solid grey lines represent the calibrated model output, and the shaded areas represent the first and third quartiles. Solid coloured lines show the evolution of individual species over time, their quartiles are omitted for clarity. Reproduced with permission from [74].	67
Figure 36	Palmitoylation (blue) and Depalmitoylation (orange) rates at steady-state for the different species in the model. The mean of 100 models is shown. Error bars show first and third quartiles. The palmitoylation rates for the various dimer species are summed. The same summing is done for the depalmitoylation rates of all dimer species. Reproduced with permission from [74].	68
Figure 37	Half-lives of the various CLiMP63 species predicted with the model. The mean of a 100 models is shown. Error bars represent the first and third quartiles. Reproduced with permission from [74].	69
Figure 38	Quantification of CLiMP63 decay labelled with TMR-star (left) (N=3). Fluorescent and Western blot of CLiMP63 labelled with TMR-star (right). The error bars represent the standard deviation. Reproduced with permission from Sandoz <i>et al.</i> [74].	69
Figure 39	The effect of zDHHc6 silencing. Knockout and over-expression on total CLiMP63 levels is shown. Model predictions are shown on the left plot and experimental validation (N=8, ***p < 0.01, ****p < 0.001) on the right. Error bars represent standard deviation. Reproduced with permission from [74].	70
Figure 40	Computational prediction of reduced depalmitoylation. The top three figures show predictions of the effect of reduced depalmitoylation on ^3H labelling decay, dimer formation and ^{35}S labelling decay. Experimental data points (N=3) are shown by red dots and the error bars represent standard deviation. The bottom two graphs show the experimental data corresponding to the ^3H and ^{35}S decay (N=3). Error bars represent standard deviation. Reduced depalmitoylation was obtained experimentally by introducing a second cysteine adjacent to the endogenous one. Reproduced with permission from Sandoz <i>et al.</i> [74].	71

Figure 41	Representation of the structure of Capillary Morphogenesis Gene 2 (CMG2), reproduced with permission from Deuquet <i>et al.</i> [118]. On the left panel, cysteines are shown in yellow. The two palmitoylated cysteines C344 and C345 can be seen on the cytosolic side adjacent to the transmembrane domain. On the right panel, mutations linked to Hyaline Fibromatosis Syndrome (HFS) are shown in red.	75
Figure 42	Palmitoyl Acyltransferase (PAT) screening experiment for CMG2. ^3H palmitate incorporation was performed in WT conditions, and when silencing multiple different PATs in order to determine which ones palmitate CMG2. The notation 'R1' indicates that zDHHC1 was silenced. Error bars show standard deviation calculated from experimental repeats (N=8).	76
Figure 43	^3H incorporation experiment on CMG2. Different experimental conditions were used: silencing zDHHC7 (R7) or zDHHC3 (R3), or using the drugs Brefeldin (bref) or Cycloheximide (CHX). Brefeldin fuses the Golgi apparatus to the ER and Cycloheximide prevents protein synthesis. Error bars represent the standard deviation calculated from experimental repeats (N=4).	77
Figure 44	Experiment showing the influence of different mutants on CMG2 levels. Results are normalised to WT conditions. CMG2 has three palmitoylation sites. A 'C' indicates that the Cysteine is still present and the palmitoylation site active. An 'A' indicates the site was mutated to alanine so that palmitoylation is no longer possible. Error bars represent standard deviation calculated from experimental repeats (N=9). . . .	78
Figure 45	Various experiments showing how ^3H palmitate incorporation changes in response to silencing zDHHC enzymes and mutating the third palmitoylation site of CMG2. Error bars show standard deviation calculated from experimental repeats (N=4).	79
Figure 46	Comparison of Total Cell Extract (TCE) and surface biotinylation experiments when silencing various PATs or APTs. Total cell extract values are shown in blue bars, and the values of the surface biotinylation experiment are shown in orange. The top of the bars show the mean, and the error bars show the standard deviation calculated from experimental repeats (N=4).	80

Figure 47	Multiple CMG2 ^{35}S pulse-chase experiments showing the effect of silencing zDHHC7 and/or zDHHC3. Error bars represent standard deviation calculated from experimental repeats (N=3).	81
Figure 48	Multiple CMG2 ^3H pulse-chase experiments showing the effect of mutating the cysteines of different palmitoylation sites. Error bars represent standard deviation calculated from experimental repeats (N=4). . . .	82
Figure 49	^3H pulse-chase experiment in WT conditions and silencing APT2. Experiments are shown in both RPE1 and HeLa cell lines. Error bars represent standard deviation calculated from experimental repeats (N=4)	82
Figure 50	A subset of the TCE and surface biotinylation experiments showing the overlap of experiments performed in both RPE1 and HeLa cell lines. Error bars represent standard deviation calculated from experimental repeats (N=4).	83
Figure 51	Illustration of the entry mechanism of the anthrax toxin, reproduced with permission from Göttele <i>et al.</i> [124]. The Protective Antigen (PA) first binds to an anthrax receptor such as CMG2. Several of these complexes then form a heptamer. Both lethal factor (LF) and edema factor (EF) can then bind to this heptamer. Once in the cell, LF and EF are released and disrupt cell signalling pathways.	84
Figure 52	^3H palmitate pulse-chase experiment following CMG2 in WT conditions and in the presence of the Protective Antigen (PA) that is part of the anthrax toxin. Error bars show standard deviation calculated from experimental repeats (N=3).	84
Figure 53	Illustration of the CMG2 model showing its structure in the ER. The 'u' prefix denotes unfolded CMG2 and 'f' indicates the folded species. The numbers indicate the palmitoylation sites that are occupied. For example, 'f12' indicates folded CMG2 which is palmitoylated on both the first and second site.	85
Figure 54	Illustration of the CMG2 model showing its structure in the Golgi apparatus.	86
Figure 55	Illustration of the different error functions used for TCE and surface biotinylation experiments. Because the surface biotinylation experiments can not accurately measure small signals, any model output under the measured value is given a zero error.	87

Figure 56	Pulse-chase radiolabelling experiments performed on CMG2. Only the calibration experiments are shown. Experimental results are shown in orange circles, with error bars indicating the standard deviation (N=3 for ^{35}S experiments and N=4 for ^3H experiments). Solid blue lines show the median of 200 models, with the shaded area representing the first and third quartile. In this case, the models are all very tightly grouped so that the quartiles are not discernible. The type of labelling, the pulse time in hours, as well as any mutants or PAT silencing is indicated in the title of each experiment. The notation 'R3' indicates that zD-HHC3 is silenced by RNA interference (RNAi). While the 'R3R7' notation indicates that both zDHHC3 and zDHHC7 are silenced by RNAi.	89
Figure 57	% Error for Pulse-chase radiolabelling experiments performed on CMG2. Only the calibration experiments are shown. The % error is shown between the mean of each experimental data point and the median of the model output. The type of labelling, the pulse time in hours, as well as any mutants or PAT silencing is indicated in the title of each experiment. The notation 'R3' indicates that zDHHC3 is silenced by RNAi. While the 'R3R7' notation indicates that both zDHHC3 and zDHHC7 are silenced by RNAi.	90
Figure 58	Surface biotinylation and TCE experiments done on CMG2. Experimental results are shown with orange circles, the error bars represent standard deviation (N=4 for surface biotinylation, N=9 for TCE and N=4 for TCE ^3H experiments). Simulated results from the model are shown with blue bars, the error bars correspond to the interquartile range. Calibration experiments are highlighted in red, all other experiments belong to the validation set. Both normal TCE and ^3H -labelling TCE were performed.	91

Figure 59	Pulse-chase radiolabelling experiments performed in the HeLa cell line. Only validation experiments are shown. Experimental results are shown in orange circles, with error bars indicating the standard deviation (N=3). Solid blue lines show the median of 200 models, with the shaded area representing the first and third quartile. In this case, the models are all very tightly grouped so that the quartiles are not discernible. The type of labelling, the pulse time in hours, as well as any mutants or PAT silencing is indicated in the title of each experiment. The notation 'R3' indicates that zDHHC3 is silenced by RNAi. While the 'R3R7' notation indicates that both zDHHC3 and zDHHC7 are silenced by RNAi.	92
Figure 60	% Error for Pulse-chase radiolabelling experiments performed in the HeLa cell line. Only the validation experiments are shown. The % error is shown between the mean of each experimental data point and the median of the model output. The type of labelling, the pulse time in hours, as well as any mutants or PAT silencing is indicated in the title of each experiment. The notation 'R3' indicates that zDHHC3 is silenced by RNAi. While the 'R3R7' notation indicates that both zDHHC3 and zDHHC7 are silenced by RNAi.	93
Figure 61	Pulse-chase radiolabelling experiments performed in the RPE1 cell line. Only validation experiments are shown. Experimental results are shown in orange circles, with error bars indicating the standard deviation (N=3 for ^{35}S experiments and N=4 for ^3H experiments). Solid blue lines show the median of 200 models, with the shaded area representing the first and third quartile. In this case, the models are all very tightly grouped so that the quartiles are not discernible. The type of labelling, the pulse time in hours, as well as any mutants or PAT silencing is indicated in the title of each experiment. The notation 'R3' indicates that zDHHC3 is silenced by RNAi. While the 'R3R7' notation indicates that both zDHHC3 and zDHHC7 are silenced by RNAi.	94

Figure 62	% Error for Pulse-chase radiolabelling experiments performed in the RPE ₁ cell line. Only the validation experiments are shown. The % error is shown between the mean of each experimental data point and the median of the model output. The type of labelling, the pulse time in hours, as well as any mutants or PAT silencing is indicated in the title of each experiment. The notation 'R ₃ ' indicates that zDHHC ₃ is silenced by RNAi. While the 'R ₃ R ₇ ' notation indicates that both zDHHC ₃ and zDHHC ₇ are silenced by RNAi.	95
Figure 63	Sankey diagrams showing model predictions of how the synthesis flux of CMG ₂ is distributed throughout the cell. WT conditions as well as zDHHC ₇ silencing and zDHHC ₃ silencing are shown. The median and standard deviation of 200 models are shown on the graph.	96
Figure 64	<i>In silico</i> ³⁵ S labelling of surface-resident CMG ₂ . Both zDDHC ₇ and zDHHC ₃ were silenced and the experiment reproduced in order to estimate the apparent half-life of surface-resident CMG ₂ . For each of the four conditions shown, the values were normalised with respect to the initial value at t=0 hours. The solid lines represent the median of 200 models. The shaded area the interquartile range.	97
Figure 65	<i>In silico</i> ³⁵ S labelling of surface-resident CMG ₂ . Total amounts of CMG ₂ are shown. The solid lines represent the median of 200 models. The shaded area the interquartile range.	98
Figure 66	Distribution of palmitoylation states of CMG ₂ at the PM. WT conditions and silencing both zDHHC ₇ and zDHHC ₃ are shown. The mean and standard deviation of 200 models are shown.	98
Figure 67	<i>In silico</i> ³⁵ S labelling of surface-resident CMG ₂ . WT conditions and APT overexpression are shown. The solid lines represent the median of 200 models. The shaded area the interquartile range.	99
Figure 68	Distribution of palmitoylation states of CMG ₂ at the PM. WT conditions and APT overexpression are shown. The mean and standard deviation of 200 models are shown.	100

Figure 69	Global Sensitivity of Metabolic Control Analysis (GMCA) workflow for characterizing sources of uncertainty in large-scale kinetic models. Diagram providing details of the various steps required for the characterization of parameters responsible for variance in kinetic models and their outputs. 107
Figure 70	Illustration of applying the GMCA workflow to a branched pathway model. (A) Overview of the branched pathway used for illustrative purposes. The parameters controlling the reaction v_4 will be further studied. (B) Distribution of 3 of the total 6 saturations (σ) present in the model, the other 3 saturations were omitted for clarity. Beta distributions were chosen for illustration purposes. Their values are obtained by sampling according to the Optimization and Risk Analysis of Complex Living Entities (ORACLE) workflow. (C) Distribution of the control coefficient: $C_{V_{\max,2}}^{v_4}$, the control of the flux of reaction v_4 with respect to V_{\max} of reaction v_2 . A normal distribution with mean 1.5 and standard deviation of 0.5 was chosen for illustration purposes. The mean (f_0) is shown in red. (D) Global Sensitivity Analysis (GSA), showing the first order sensitivity index (Si) of $C_{V_{\max,2}}^{v_4}$ with respect to the 6 saturations present in the model. (E) After having determined that σ_2 had the highest Si, we assume to have determined its value exactly. (F) Resulting distribution of $C_{V_{\max,2}}^{v_4}$ once the value of σ_2 is determined. 108
Figure 71	E.coli network diagram illustrating the reactions of the kinetic model. Diagram does not include all the reactions of the model. (A) Overview of the reaction network, color-coding indicates to which subsystem a reaction belongs. Abbreviations are as follows: G/G - Glycolysis and Gluconeogenesis, Pentose Phosphate Pathway (PPP) - Pentose Phosphate Pathway, TCA - Tricarboxylic Acid Cycle, Pyr - Pyruvate Metabolism, Glyox - Glyoxylate Metabolism. The reactions indicated in blue correspond to the Pentose phosphate pathway (PPP) subsystem, which was the focus of this study. A larger version is available in the Appendix (Figure 75). (B) Focus on the PPP subsystem. (C) Single reaction belonging to the PPP, the kinetic mechanism of the reaction is shown, the two kinetic parameters to be sampled using the ORACLE workflow are highlighted in blue and green. 113

Figure 72	Sobol sensitivity indices for flux control coefficient with respect to subsystem enzyme saturation levels. First order (dark green) and total effect (light green) Sobol sensitivity indices of control coefficient $C_{\text{GND}}^{\text{EDA}}$, corresponding to the sensitivity of the flux of reaction EDA with respect to $V_{\text{max,GND}}$. All enzyme saturations within a subsystem were grouped when performing sensitivity analysis. The 200'000 samples were split into three groups, from which the mean and standard deviation were calculated. The whiskers indicate the standard deviation and the bars report the mean. (A) Saturations sampled over their entire feasible range. (B) Saturations outside of the 5 studied subsystems, ie. the study scope, are sampled between their mean +/- 10%.	115
Figure 73	Example of the proposed GMCA workflow, performing GSA at different levels: first subsystems, then reactions and finally individual saturations. First order (dark green) and total effect (light green) Sobol sensitivity indices of control coefficient $C_{\text{EDD}}^{\text{EDA}}$, corresponding to the effect of $V_{\text{max,EDD}}$ on the flux of reaction EDA. (A) GSA was first performed by grouping enzyme saturations into subsystems. (B) An illustration of the first order sensitivity indices is superimposed on the network diagram, darker shades indicate higher first order sensitivity indices. (C) GSA was then performed by considering only reactions from the PPP and grouping enzyme saturations according to reaction. (D) First order sensitivity indices are superimposed on an illustration of the PPP. Darker shades indicate higher values. (E) Finally, the contribution of individual kinetic parameters in the EDD reaction were analysed. (F) The EDD reaction is shown, including substrates and its kinetic expression.	117

Figure 74	Overview of GMCA in a metabolic engineering context and <i>in silico</i> validation of results. An overview of how GMCA fits into a continuous improvement cycle. A study scope is first defined, selecting the control coefficients and sources of uncertainty. These can correspond to a subsystem, a reaction or any desired set of parameters. GMCA is then performed, identifying which sources of uncertainty contribute the most to the uncertainty in the control coefficients. Once identified, these sources of uncertainty can be better estimated, for example by experimental kinetic measurements, literature mining or data mining. This ultimately leads to better kinetic model predictive power, as the uncertainty in the control coefficients has been reduced.	118
Figure 75	E.coli network diagram illustrating the reactions of the kinetic model. Diagram does not include all the reactions of the model. Overview of the reaction network, color-coding indicates to which subsystem a reaction belongs. Abbreviations are as follows: G/G - Glycolysis and Gluconeogenesis, PPP - Pentose Phosphate Pathway, TCA - Tricarboxylic Acid Cycle, Pyr - Pyruvate Metabolism, Glyox - Glyoxylate Metabolism. The reactions indicated in blue correspond to the Pentose phosphate pathway (PPP) subsystem, which was the focus of this study.	169

LIST OF TABLES

Table 1	Example of generating a LH design for two parameters, four levels and eight samples	29
Table 2	Parameter bounds for k_{decay1} and k_{decay3} with either constrained or wide bounds.	40
Table 3	Increase in complexity between the Calnexin model and the the complete CLiMP ₆₃ model, including all proposed structures	59
Table 4	Speed Comparison of Julia and MATLAB. One hundred CLiMP ₆₃ models were integrated to steady-state as an indication of performance.	61
Table 5	Comparison of number of model parameters and Ordinary Differential Equations (ODE)s for the different CLiMP ₆₃ model structures which were tested.	63
Table 6	Reactions in the Pentose Phosphate Pathway (PPP) and their corresponding Km parameters. The Km parameters are grouped together when studying the contribution of an entire reaction to the uncertainty of a control coefficient.	112
Table 7	143
Table 8	167

LIST OF ACRONYMS

ANOVA	Analysis of Variance
ANTXR ₂	Anthrax toxin receptor 2
APT	Acyl-protein Thioesterases
CC	Control Coefficient
CCC	Concentration Control Coefficient
CLE	Chemical Langevin Equation
CLiMP ₆₃	Cytoskeleton-linking Membrane Protein 63
CMA	Covariance Matrix Adaptation
CMA-ES	Covariance Matrix Adaptation Evolution Strategy
CME	Chemical Master Equation
CMG ₂	Capillary Morphogenesis Gene 2
CP	Cytoplasm
DGSM	Derivative-based Global Sensitivity Measures
DNA	Deoxyribonucleic Acid
DOE	Design of Experiments
ECM	Extracellular Matrix
ER	Endoplasmic Reticulum
FBA	Flux Balance Analysis
FCC	Flux Control Coefficient
FDM	Finite Difference Method
FF	Fractional Factorial
GA	Genetic Algorithm
GSA	Global Sensitivity Analysis
GMCA	Global Sensitivity of Metabolic Control Analysis

HDMR	High Dimensional Model Representation
HFS	Hyaline Fibromatosis Syndrome
LH	Latin Hypercube
MCA	Metabolic Control Analysis
MD	Molecular Dynamics
mRNA	messenger RNA
NSGA-II	Nondominated Sorting Genetic Algorithm II
ODE	Ordinary Differential Equations
ORACLE	Optimization and Risk Analysis of Complex Living Entities
PA	Protective Antigen
PAT	Palmitoyl Acyltransferase
PM	Plasma Membrane
PPP	Pentose Phosphate Pathway
PTM	Post-Translational Modification
RNA	Ribonucleic Acid
RNAi	RNA interference
RPE ₁	Retinal Pigment Epithelial 1
RRE	Reaction Rate Equation
RTC	Ribosome-Translocon Complex
SA	Sensitivity Analysis
SDS	Sodium dodecyl sulfate
SDS-PAGE	Sodium dodecyl sulphate–polyacrylamide gel electrophoresis
shRNA	short hairpin RNA
siRNA	silencing RNA
TCE	Total Cell Extract
TFA	Thermodynamics-based Flux Analysis
QSSA	Quasi Steady-State Approximation
tQSSA	Total Quasi Steady-State Approximation
WT	Wild type

1 INTRODUCTION

1.1 SYSTEMS BIOLOGY

Before diving into the central questions that this thesis strives to answer, a small background on the field of Systems Biology may place the work in context. Specifically that of understanding and engineering complex living systems. Systems Biology is a relatively young field that primarily uses mathematical and computational modelling in the study of complex biological systems. Its birth can be linked to the beginning of the genomics era, when increasingly large amounts of genetic information were being made available through Deoxyribonucleic Acid (DNA) sequencing. A natural question with all this information was how it fits together to give rise to living organisms and all their associated behaviours. As such, Systems Biology is characterised by its holistic approach of putting together building blocks to describe larger complex systems. The large amounts of information require the use of efficient mathematical and computational methods.

Some of the problems that Systems Biology can help address are important biological questions. Understanding the development of cancer cells in a healthy organism, for example [1–5]. Other applications are closely tied to Biotechnology, using living systems as a means of production [6, 7]. The production of the anti-malarial drug Artemisin and that of Insulin have been greatly improved by Systems Biology approaches [8–10]. Systems Biology can help design better treatment therapies [11, 12], where the amount and timing of drug administration are optimised to minimise side effects and cost while maximizing drug efficacy [13–16]. The human digestive system is a rich community of different bacteria working together [17, 18]. Understanding the relationship between these bacteria and the host requires a holistic, systems-level approach. But Systems Biology can also contribute to more conventional industries [19–21]. From producing biofuels, to upgrading waste biomass and replacing traditional petrochemical feedstocks, the applications are numerous.

1.2 CELL REGULATION AND SIGNALLING

A big question in cell biology is describing how cells regulate themselves. How do they sense the outside world for potential threats such as excessive heat or toxins? How do they put defense mechanisms into place? How do they reduce their defenses once a threat has passed? How do they adapt and regulate their internal composition to ensure their proper functioning?

To explain some of these behaviour we first need to understand how cells function.

All cells contain the complete instructions for their growth and reproduction, irrespective of whether it is a cell in a multi-cellular organism like a human, or a single-cell bacteria. This set of instructions is encoded in DNA. Simply put, DNA is a very long sequence of just four letters: C, G, A and T. These four letters correspond to the four molecules which are attached one after another to form DNA: Cytosine, Guanine, Adenosine and Thymine. These can be attached in any order, forming a long strand containing all instructions for the organism to grow and reproduce. In practice, DNA doesn't exist as a single strand, but a second "mirror image" strand bonds along the first and they are both twisted together to form a spiral-like shape. The code in the first strand is mirrored in the second according to a pairing of the four letters. C is always paired to G and A is always paired to T. If a C is in the first strand, it is bonded to a G in the second strand. Human DNA contains about 3.2 billion of these pairs, while a bacteria like *E. coli* contains 4-5 million.

The code contained in the DNA can be translated into proteins, a large family of biomolecules which are essential for life. For example, the protein keratin is found in hair, nails, claws and hooves. Proteins are found in the digestive system in the form of enzymes, where they break down food into smaller molecules that cells may directly use. Proteins are also key to the immune system, where they recognise and destroy disease-inducing bacteria. Proteins are themselves used to translate DNA into more proteins, as well as making sure this translation is free of errors.

Once synthesized, proteins have variable lifetimes, and must be continuously monitored for correct functioning. Depending on the environmental situation, more or less proteins may be needed. After a large meal, a lot of enzymes are needed to digest all the food. During a severe flu, many immune system proteins are needed to clear out the viruses causing the disease. One method of quickly regulating protein levels is Post-Translational Modification (PTM). Once a protein has been translated from the DNA code into a physical molecule, additional small molecules can be added and removed. These can function as flags that carry information. Depending on the PTM, this may cause a protein to be degraded, transported elsewhere or modified in other ways. Palmitate is one such small molecule that can be attached to proteins. This process is called S-Palmitoylation, often referred to as just Palmitoylation. The attachment of these molecules is generally done by an enzyme, and in cases such as Palmitoylation can also be reversed by another enzyme. This reversibility makes it possible for Palmitoylation to be used to dynamically regulate a protein in response to changing environmental conditions.

1.3 STUDYING PALMITOYLATION IN CELLS

In order to study how Palmitoylation can regulate proteins, there are several experimental methods at our disposal. Radiolabelling can be used to study the synthesis and degradation of proteins. This replaces Sulfur atoms in the external nutrients of the cells with the radioactive Sulfur isotope ^{35}S . The cells uptake these radioactive Sulfur atoms and incorporate them in their proteins. By measuring the radioactivity of a particular protein, we can obtain information on how fast this protein is synthesized and how fast it is degraded. A second method involves labelling not the protein but the palmitate attached to the protein during Palmitoylation. Similar to ^{35}S labelling, the palmitate available to the cell is labelled by replacing the Hydrogen atoms with the radioactive isotope ^3H . In this case, instead of following the synthesis and degradation of the protein itself, we can now follow the attachment and detachment of palmitate to the protein.

Using ^3H palmitate labelling, it is also possible to estimate the fraction of palmitoylated vs. non-palmitoylated protein using a method called acyl-RAC [22], which attaches a molecule to the sites on the protein that are not occupied by palmitate.

With Immunofluorescence microscopy [23] it is possible to observe the distribution of a protein in a cell, making it possible to estimate the fraction of protein in each cellular compartment. In addition to all these methods, it is possible to modify the normal functioning of the cell and repeat the previous experiments.

Palmitoylating enzymes can be removed using siRNA, a molecule similar to DNA that can be tailored to block the translation of a protein.

It is also possible to change the DNA describing a protein so that it may not be palmitoylated anymore, creating a mutant. By changing the site on the protein where palmitate is attached, palmitoylation is blocked without changing the structure of the protein.

The choice and variety of methods makes it clear there is a combinatorial increase in the amount of experimental data that can be collected. Furthermore, there may be several enzymes that palmitoylate a single protein, and multiple palmitoylation sites on a single protein, adding to the complexity of protein regulation.

1.4 MATHEMATICAL MODELLING OF PALMITOYLATION

The challenge with this large number of different experimental methods is an overflow of information. It becomes hard to reason about the regulation mechanisms while keeping in mind all the experimental data simultaneously. This is exacerbated in biological systems, where results are often partial observations of a complex system. An analogy is taking snapshots of an intricate machine from a few limited viewpoints in a dark room. These

snapshots correspond to the experimental data, and the machine to the biological system being studied. It is simply not feasible to exhaustively quantify all properties of this machine in this situation. Furthermore, the relationship between dynamic properties (how the machine functions over an entire day) and static ones (how it functions just when the snapshot is taken) is not easy to deduce, requiring careful design of experiments and hypothesis testing. In order to be able to make sense of the pile of snapshots, working hypotheses are formulated that need to be consistent with all available data. In turn we will be able to verify our understanding of the system against each piece of experimental data to hopefully reach a consistent hypothesis.

In order to verify the hypotheses that have been formulated, we may put them into a more rigorous mathematical form. In this way, we may use mathematical modelling as a tool for hypothesis building and verification. In building these models, we will borrow many methods and approaches from Systems Biology and related fields. A particular advantage of Systems Biology is its holistic approach, the consideration of entire systems and their interactions. This approach lends itself particularly well to our goal of building mathematical models that can describe an entire set of experimental data. In defining the model, we also define its boundaries, in particular: which behaviours we want to reproduced with a mathematical model and which behaviours are beyond the scope of the model.

Once the experimental data has been collected we will use several different modelling and numerical methods in order to build and calibrate a mathematical model that can adequately explain the experiments. The first task is to classify the different types of reactions in the model. In the models we will build there are generally four types of reactions:

1. Synthesis - producing a protein
2. Transport - moving a protein between cellular compartments
3. Enzymatic - for example, palmitoylation and depalmitoylation, which involve an enzyme attaching or removing palmitate
4. Degradation - breaking down a protein

Depending on the type of reaction, we will assign a mechanism to it, describing the rate of the reaction as a function of the relevant conditions. These conditions cover factors such as concentrations of enzymes and proteins in different compartments, temperature and catalytic activity of the enzymes.

Once we have determined the type of every reaction in the model and assigned it a mechanism we can move on to connecting these reactions together. Here we need to figure out where synthesis of the protein occurs, through what path it is transported to different parts of the cell and where

it can be degraded. Previous knowledge from the field is very useful at this stage. Information regarding which enzymes can modify a protein, or in which compartments a protein is present can greatly simplify the model. This results in a model structure, a map describing the possible evolution of the protein in the cell with corresponding mathematical equations.

The next step involves translating this into computer code so that these mathematical equations may be computationally solved. If a small number of mathematical equations describe the model, it may be feasible to translate these mathematical equations into code by hand. Otherwise, a computational tool to automatically generate this code from the mathematical equations is needed.

At this point, we can already use the model to simulate experiments such as ^{35}S labelling, but we don't have information on the model parameters yet. These characterise the catalytic activity of the enzymes, how fast the protein is synthesized, transported and degraded. These parameters will be estimated using heuristic optimisation algorithms with the goal that the model should accurately reproduce the experimental results. In essence, the model is calibrated using the experimental data.

After having completed this process of building and calibrating, we can use the resulting model to explain and better understand the behaviour observed during the experiments. Ideally, if the calibration was successful, all experimental data is well reproduced and the model is consistent with the system we are studying, in this case the regulation of a specific protein by palmitoylation. This gives a unifying view of how the protein is regulated through palmitoylation, while also pinpointing which mechanisms are responsible for different behaviour. Because the experimental data is a partial observation of multiple superposed behaviours, a model provides a clear picture of how the basic building blocks fit together to create the observed behaviours. A subsequent step is to use the model to simulate conditions which are not experimentally feasible, providing another layer of insight.

Although a successful calibration yields a model consistent with experimental data, there is still the problem of uncertainty. Firstly, by using heuristic optimisation methods, we obtain a population of models that explain the data equally well. Meaning that we obtain a range of suitable parameters, rather than exact values. This means there is uncertainty in the model parameters. The parameters may have multiple values that are consistent with the data. Secondly, heuristic optimisation does not guarantee finding all suitable solutions. Even though our population of models may seem to cover the parameter space, we cannot exclude that there are unexplored regions in the parameter space that would provide equally good or even better models. This is a problem with heuristic optimisation itself, and cannot be easily solved without reformulating the problem and optimisation method. The first issue of the population of models can however be characterized using uncertainty quantification. Using these methods we

can quantify the propagation of uncertainty in model parameters to model behaviours. In doing so, it is possible to identify parameters that greatly change model behaviour and others that don't have a significant effect within the parameter space.

1.5 UNCERTAINTY IN KINETIC MODELS OF METABOLISM

In the second part of this thesis, we will be studying models of bacterial metabolism. In doing so, we will tailor the same methods of uncertainty quantification to kinetic models of metabolism.

The process of building a metabolic model is vastly different from the protein regulation models previously discussed. The DNA of the organism to model first needs to be determined by sequencing. Once the DNA sequence is obtained, it is possible to analyze it in order to find all enzymes that are encoded within. As enzymes are the principal catalyzers of biological systems, a list of enzymes can be translated to a list of reactions. Using previous knowledge concerning which reactions an enzyme can catalyze, a reaction network of the organism's metabolism can be drafted. This model draft is often then curated to match experimental data. Using this draft model it is possible to predict the biomass yield, how efficiently a carbon source such as glucose can be converted to biomass. This can be compared to the experimentally measured biomass yield in order to identify missing reactions [24–26]. It is also possible to determine which genes in the DNA sequence are essential for the organism to grow. This can be compared between the draft model and experimental data in order to further refine the model [27, 28].

From this curated metabolic model it is then possible to build a kinetic model. Reaction fluxes and metabolite concentrations first need to be estimated with methods such as Flux Balance Analysis (FBA) [29, 30] and Thermodynamics-based Flux Analysis (TFA) [31–34]. The mechanism of each reaction then needs to be determined. Experimental data can be used where available, and educated guesses made using reaction stoichiometry otherwise. Then the kinetic parameters are either incorporated from the literature or sampled [35, 36]. At this point the model can be integrated in order to study dynamic response over time.

Another analysis that can be performed is Metabolic Control Analysis (MCA) [36–42], which is a type of local sensitivity analysis describing the change in fluxes and concentrations as a function of a change in kinetic parameters. The quantitative change in flux or concentration in response to a change in a kinetic parameter is described by the corresponding control coefficient [36–42].

We will focus on uncertainty quantification in the context of MCA. Specifically, what we are interested in is how to improve the precision of control coefficients calculated with MCA. Because kinetic parameters are sampled,

we end up with a population of kinetic models, and a population of control coefficients. This leads to the similar issue of uncertainty as that of the palmitoylation models. Although MCA can predict which kinetic parameters will increase or decrease fluxes and concentrations, the population of control coefficients describe a range of predictions, sometimes contradicting each other. By identifying how uncertainty propagates from the kinetic parameters to the control coefficients, we can rank which kinetic parameters to measure in order to reduce the uncertainty in the control coefficients. This avoids the need for experiments to exhaustively measure every kinetic parameter. Instead, those which have the most impact on the control coefficient of interest can be measured first, greatly reducing the experimental resources needed.

1.6 THESIS OVERVIEW

The chapters of this thesis are organised in the following way

Chapter 2 starts with a published model of the chaperone protein Calnexin [43]. This model is already calibrated to experimental data. The goal of this chapter is to introduce the various biological and modelling concepts that will be used throughout the thesis. We will also provide an example of how uncertainty quantification can be applied to a model and what additional information can be extracted.

Chapter 3 discusses the building and calibration of a model describing the regulation by palmitoylation of Cytoskeleton-linking Membrane Protein 63 (CLiMP63). CLiMP63 has several different functions in different cellular compartments, and two currently known palmitoylating enzymes. The biological background as well as the discoveries enabled by the model will be discussed here.

Chapter 4 details the building of a Capillary Morphogenesis Gene 2 (CMG2) model. Another protein which has multiple functions, and whose palmitoylation plays a key role in the infection of cells by the anthrax toxin. We will discuss the experimental data available, how the model was built and calibrated and how palmitoylation regulates CMG2.

Chapter 5 deals with implementing uncertainty quantification into the Metabolic Control Analysis (MCA) of a kinetic model of metabolism. In particular, we will show the challenges of applying Global Sensitivity Analysis (GSA) to these models, and propose an efficient approach tailored to large scale metabolic models.

2 CALNEXIN

In this chapter we will go over some of the fundamental concepts that will be used throughout the thesis. Reaction kinetics, and enzyme kinetics in particular, are central to all models in this thesis. These are the building blocks with which all subsequent models will be built. We will describe the experimental data on the protein Calnexin and the biological background necessary to understand how it can be used for modelling. We will also go through the basics of Global Sensitivity Analysis (GSA), how the sensitivity indices are calculated, what they mean and how to interpret them. Finally, a previously calibrated model of the protein Calnexin will be used to illustrate the kinds of information that can be extracted with GSA.

The Calnexin model mentioned and experimental methods described in this chapter has been developed by Dallavilla *et al.* [43]. The experimental data pertaining Calnexin was produced by various members of the Van der Goot lab, and has been published in two papers [43, 44].

2.1 REACTION KINETICS

Ordinary Differential Equations (ODE) are profusely used to model chemical and biochemical reactions in many different contexts. But before we dive into the particular kinetics used, we will discuss their origin and the assumptions necessary for their validity. A summary relevant to this thesis will be given here, for a more in-depth explanation of the mathematical derivation the reader is referred to the excellent explanations of Desmond Higham [45].

2.1.1 Chemical Master Equation

The following equations are widely used to model reactions in a variety of chemical and biological systems. The common assumptions on which they all rely on are homogeneity, thermal equilibrium and constant volume. The assumption of homogeneity is often what is meant when discussing a well-mixed system. This means that physical characteristics such as concentrations, pressure and temperature are homogeneously distributed throughout the system, without an accumulation in a subvolume.

Next we introduce the state vector \mathbf{x} , which describes the probability of the system being in a certain state at time t . It has a length n that corre-

sponds to the total number of possible states of a system. The evolution of the state vector can be described with the Chemical Master Equation (CME):

$$\frac{dP(\mathbf{x}, t)}{dt} = \sum_{j=1}^m \alpha_j(\mathbf{x} - \mathbf{v}_j) P(\mathbf{x} - \mathbf{v}_j, t) - \alpha_j(\mathbf{x}) P(\mathbf{x}, t) \quad (1)$$

$P(\mathbf{x}, t)$ describes the probability of being in the different system states at time t . Like \mathbf{x} it is also of length n . \mathbf{v}_j is the reaction vector corresponding to reaction j . It describes the change in the state vector \mathbf{x} when reaction j occurs. $\alpha_j(\mathbf{x})$ describes the propensity of reaction j occurring given the state of the system \mathbf{x} . The propensities for first order reactions and some second order reactions are given for illustration purposes:

$$\text{1st Order} \quad S_M \rightarrow \dots \quad \alpha_j(\mathbf{x}(t)) = c_j x_M(t) \quad (2)$$

$$\text{2nd Order} \quad S_M + S_N \rightarrow \dots \quad \alpha_j(\mathbf{x}(t)) = c_j x_M(t) x_N(t) \quad (3)$$

$$\text{Dimerisation} \quad S_M + S_M \rightarrow D_M \quad \alpha_j(\mathbf{x}(t)) = c_j \frac{1}{2} x_M(x_M - 1) \quad (4)$$

Although the CME is an ODE describing the probability of each state over time, the size of the state vector scales with respect to the number of molecules in the system. For a simple system where k molecules may exist in two different forms, there are a total of $k + 1$ states. This means that for many systems, the sheer size of the state vector \mathbf{x} makes analytical or computational analysis impractical or infeasible.

Instead, we can solve single realisations of the CME rather than the entire probability distribution. This algorithm is called the Stochastic Simulation Algorithm, also known as Gillespie's algorithm [46, 47]. Using the current state and the propensity functions, Gillespie's algorithm performs the equivalent of a random dice roll to estimate the time to the next reaction and selects a reaction using a weighting derived from the propensity functions.

In order to make Gillespie's algorithm faster, we can take larger time steps in which multiple reactions occur, called Tau-leaping. For this approximation to be valid, it is important that $\alpha_j(\mathbf{x}(t))$ changes only slightly during the time step τ .

A further development to Tau-leaping can be made when the number of molecules in the system is large. Making these assumptions leads to the Chemical Langevin Equation (CLE):

$$d\mathbf{y}(t) = \sum_{j=1}^m \mathbf{v}_j \alpha_j(\mathbf{y}(t)) dt + \sum_{j=1}^m \mathbf{v}_j \sqrt{\alpha_j(\mathbf{y}(t))} dW_j(t) \quad (5)$$

The state variable has been changed to $\mathbf{y}(t)$ to underline the fact that it is now a continuous variable describing the number of molecules in each

state. Each $W_j(t)$ is a normally-distributed variable used to represent an independent scalar Brownian motion. This is the random selection component of this time-stepping algorithm, similar to the dice roll in Gillespie's algorithm. The $W_j(t)$ term means that there is still a degree of stochasticity in the system. There is still a random process by which $W_j(t)$ must be selected. In practice, this means sequential simulations of the system may be different, so that multiple simulations are often run to obtain a distribution of possible outputs.

In order to finally obtain the deterministic description that we will use for the rest of this thesis, we need to take the 'thermodynamic limit' as Desmond [45] calls it. This is the limit where the system volume and molecule counts tend to infinity but the concentrations remain constant. This is equivalent to a large system that has a large number of molecules. Because $W_j(t)$ is normally distributed with mean 0, taking this limit yields the Reaction Rate Equation (RRE):

$$\frac{d\mathbf{y}(t)}{dt} = \sum_{j=1}^m \mathbf{v}_j \alpha_j(\mathbf{y}(t)) \quad (6)$$

And in this case the reaction propensities are given as follows:

$$\begin{array}{lll} \text{1st Order} & S_M \rightarrow \dots & \alpha_j(\mathbf{y}(t)) = k_j y_M(t) \end{array} \quad (7)$$

$$\begin{array}{lll} \text{2nd Order} & S_M + S_N \rightarrow \dots & \alpha_j(\mathbf{y}(t)) = k_j y_M(t) y_N(t) \end{array} \quad (8)$$

$$\begin{array}{lll} \text{Dimerisation} & S_M + S_M \rightarrow D_M & \alpha_j(\mathbf{y}(t)) = k_j y_M(t)^2 \end{array} \quad (9)$$

Note that the constants c_j have been replaced with k_j to underline the change of units from number of molecules to concentrations. In order to convert from one type of constants to the other, all that is needed is to compare the relevant propensity equations $\alpha_j(\mathbf{y}(t))$. Converting from molecule counts to concentrations is done using the volume and Avogadro's number:

$$y_M(t) = \frac{x_M(t)}{n_A \text{vol}} \quad (10)$$

For example, comparing Equation 2 and Equation 9 for a first order reaction implies $c_j = \frac{k_j}{n_A \text{vol}}$

2.1.2 Law of Mass action

The law of mass action states that the rate of a chemical reaction is a function of the concentration of reactants and a kinetic constant [48]. It is actually a specific case of the RRE for first order reactions.

For example, given the reaction:



In this example, the law of mass action describes the rate of reaction as $k[A]$ where $[A]$ denotes the concentration of species A and k is the kinetic constant of the reaction. The stoichiometric coefficients of A and B denote how many species are produced and consumed in the reaction. In this case the stoichiometric coefficients for A and B are 1 and -1 respectively. By multiplying the stoichiometric coefficient with the rate of reaction, one gets the rate of change of a species. For this example:

$$\frac{d[A]}{dt} = -k[A] \quad \frac{d[B]}{dt} = k[A] \quad (12)$$

This system of ODEs can then be solved using one of the many existing numerical integration algorithms. As mentioned earlier the law of mass action is valid for a homogeneous, well-mixed system [48]. Both A and B must be uniformly distributed, as well as other variables that may affect the rate of reaction, such as temperature, pressure, inhibitors or catalysts.

2.1.3 Enzyme kinetics

In biological systems, enzymes play a critical role as catalysts. A typical irreversible enzymatic reaction can be represented as follows:



Where S is the substrate of enzyme E , SE the enzyme-substrate complex that is formed when the two bind together, and P the product of the reaction. The final step of product formation is considered irreversible, an assumption that is justified if the reaction is highly exothermic, $\Delta G \ll 0$, or if there is much more substrate than product, $S \gg P$ [49, 50].

Using the law of mass action, we can write out the differential equations for the four species in the system:

$$\frac{d[S]}{dt} = -k_b[E][S] + k_{ub}[ES] \quad (14)$$

$$\frac{d[E]}{dt} = -k_b[E][S] + k_{ub}[ES] + k_{cat}[ES] \quad (15)$$

$$\frac{d[ES]}{dt} = k_b[E][S] - k_{ub}[ES] - k_{cat}[ES] \quad (16)$$

$$\frac{d[P]}{dt} = k_{cat}[ES] \quad (17)$$

Where k_b and k_{ub} are the binding and unbinding constants of the substrate to the enzyme. k_{cat} is the constant of the irreversible reaction step producing P.

Although accurate, having four differential equations to describe one reaction has a high computational cost, especially when considering bigger models with tens to hundreds of reactions. What is of interest is the overall rate of the reaction, i.e. $S + E \rightarrow P + E$. To this end there are several approximations that can be made:

1. Equilibrium approximation [51]. The Enzyme and substrate are immediately in equilibrium, i.e.

$$k_b[E][S] = k_{ub}[ES] \quad (18)$$

This approximation is valid when $k_{cat} \ll k_{ub}$. In other words, the rate constant of product formation must be much lower than that of dissociation. In this way, the equilibrium between free enzyme, substrate and the enzyme-substrate complex is maintained even as substrate is transformed into product.

2. Quasi Steady-State Approximation (QSSA) [52]. The concentration of the intermediate complex ES does not change on the time scale of product formation, i.e.

$$k_b[E][S] = k_{ub}[ES] + k_{cat}[ES] \quad (19)$$

This approximation is valid if the enzyme concentration is much less than the substrate $E \ll S$, or if the Michaelis constant $K_m = \frac{k_{ub} + k_{cat}}{k_b}$ is large.

3. Total Quasi Steady-State Approximation (tQSSA) [53, 54]. The total substrate $S_t = S + ES$ and total enzyme $E_t = E + ES$ are introduced. The concentration of the total substrate does not change on the time scale of product formation. This approximation is valid in cases where enzyme concentration is both high and low relative to substrate concentration.

Generally speaking, tQSSA has a wider range of validity [53], accounting for situations with both high and low amounts of enzyme relative to the substrate. Compared to the domain of validity of the QSSA, the tQSSA actually overlaps and extends it [54]. Furthermore, it is easy to account for competition between multiple substrates for the same enzyme using tQSSA. The general form of an enzymatic reaction using tQSSA is then:

$$\frac{dS_i}{dt} = \frac{k_{cat} E_t S_{i,t}}{K_{m,i} (1 + \sum \frac{S_{j,t}}{K_{m,j}}) + S_{i,t} + E_t} \quad (20)$$

Where S_i is one of the substrates of the enzyme E . The competition between substrates is accounted for in the $\sum \frac{S_{j,t}}{K_{m,j}}$ term. tQSSA is used to model all enzymatic reactions in palmitoylation models unless stated otherwise.

2.2 EXPERIMENTAL DATA

In this section we will explain the biological background and experimental data which are used to calibrate the Calnexin model, as well as the CLiMP63 model in [Chapter 3](#) and the CMG2 model in [Chapter 4](#). We explain how the PTM palmitoylation works on a molecular level, and what the experiments measure. We will then talk of the different types of experimental methods used and how they are combined to produce the experimental data we will later use.

Details of the experimental protocols are available in the paper published by Dallavilla *et al.* [\[43\]](#).

2.2.1 Palmitoylation Background

S-Palmitoylation is a PTM consisting in the attachment of a C16 acyl chain (palmitate) to a cysteine residue through a thioester bond ([Figure 1](#)) [\[55\]](#), often called simply "palmitoylation".

This process is a subclass of the more general S-Acylation. It is also the only known reversible lipid modification that proteins may undergo [\[56\]](#). The interconvertible behaviour of a reversible PTM such as palmitoylation may play a role in cellular signaling and regulation through a switch-like behaviour, like phosphorylation or ubiquitination. Depending on whether the protein is modified or not its behaviour can be altered in a tunable manner.

Several hundred proteins have already been identified as palmitoylation targets in mammals [\[57\]](#). The known roles of palmitoylation are diverse, it plays a role in protein localisation, bacterial and viral infections, and also interacts with other signalling mechanisms such as phosphorylation [\[58–62\]](#). The enzymes that catalyse palmitoylation belong to a family of proteins containing a DHHC-motif domain: named for the repetition of the sequence of the four amino acids Aspartic Acid (D), Histidine (H) and

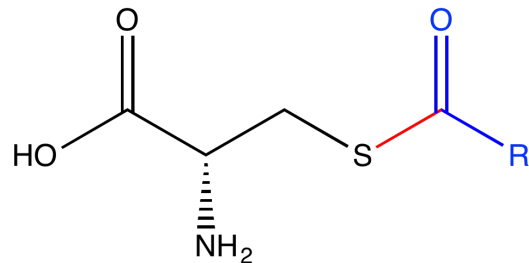


Figure 1: Illustration of a thioester bond between Cysteine and a fatty acid. The cysteine residue is in black, the thioester bond in red and the fatty acid in blue.

Cysteine (C) [63]. It is this DHHC domain that is the active site of the enzyme, catalysing the addition of a palmitate [55, 64]. These enzymes are also sometimes called Palmitoyl Acyltransferase (PAT). The corresponding class of de-palmitoylation enzymes are APT. Certain enzymes involved in palmitoylation, such as zDHHC6 and its corresponding APT, may be palmitoylated themselves [65], creating the possibility of a signalling cascade using interconvertible enzymes, similar to phosphorylation cascades [66].

2.2.2 Radiolabelling

The general technique of labelling proteins with radioactive atoms goes back at least to the early 1970s [67]. It has been more recently tailored to probe palmitoylation of specific proteins in cultured cells [68]. This involves using molecules which contain radioactive isotopes, such that the chemical properties are the same, but they can be identified from their radioactivity. Using a culture medium in which a specific molecule contains radioactive atoms, we can follow the incorporation and turnover of these in the cell's proteins. By replacing certain molecules in the culture medium by their radioactive or non-radioactive variants, radiolabelling can be turned on or off as shown in Figure 2.

Many different molecules in the culture medium can be replaced with a variant where some of the atoms are radioactive. In our experiments, two different kinds of radiolabelling will be used, as shown in Figure 3. The first replaces the non-radioactive hydrogen atoms in palmitate with radioactive ^3H atoms. This can be used to track the dynamics of palmitoylation and de-palmitoylation, the attachment and removal of palmitate from the protein. The second replaces the sulfur atoms in cysteine with radioactive ^{35}S atoms. This can be used to track the synthesis and decay of the protein itself.

In practice, radiolabelling is carried out in a cell culture where the medium is changed multiple times to turn labelling on and off. For a pulse-chase experiment, the medium will first be the radioactive variant, the pulse, before being changed to the non-radioactive variant, the chase. Several samples will be taken at different time points of the chase phase. These samples will be lysed and then ran on a western blot. The radioactivity of the band corresponding to the protein of interest is then measured. Because the half-lives of both ^3H and ^{35}S are in the order of years and months, this means relatively little will decay between sample collection and analysis. ^3H has a half-life of roughly 12 years [69] while ^{35}S has a half-life of about 87 days [70].

Because we are studying cells in an *in vitro* culture, this means that the observations are of a population of cells rather than a single cell. There are therefore not only many copies of the same protein within each cell, but also many cells in the culture. This means that the data is measuring the presence of a large number of proteins. Recall in Section 2.1.1, that a large

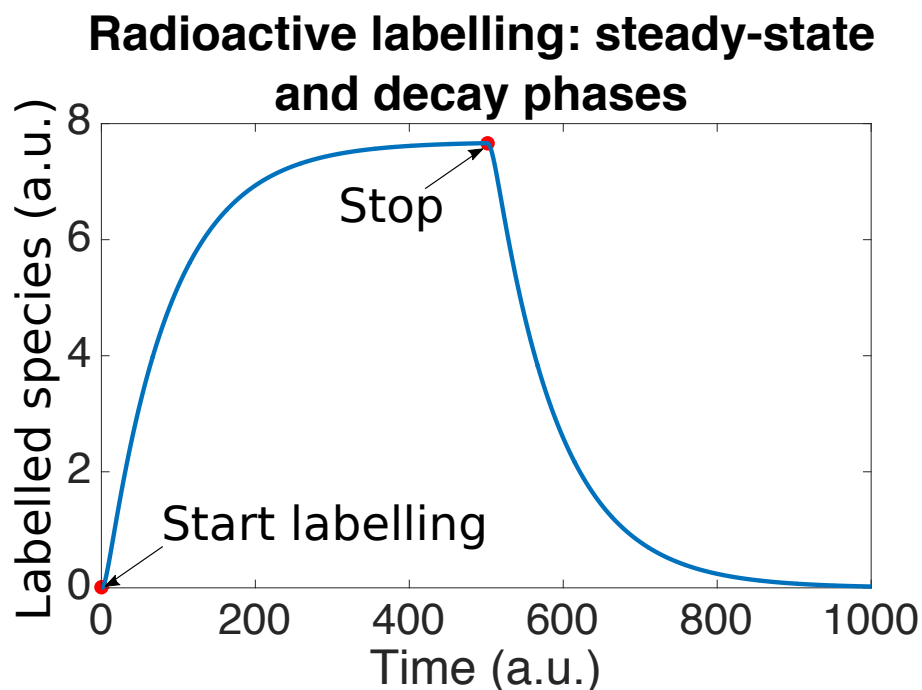


Figure 2: Illustration of a pulse-chase experiment. The labelled compound is made available through the medium at $t = 0$, for a pulse duration of about 500 a.u. The chase starts at $t = 500$ when the labelled compound is removed from the system's environment. During the pulse, the radioactive signal increases as radioactive atoms are incorporated into newly synthesized proteins. When the culture medium is changed back to being non-radioactive, newly synthesized protein are once again made with non-radioactive species. Hence the radioactive signal decreasing during the chase. Note that the plateau during the pulse phase means that all protein has been replaced with the labelled variant. Depending on the protein turnover and the pulse duration this isn't necessarily the case.

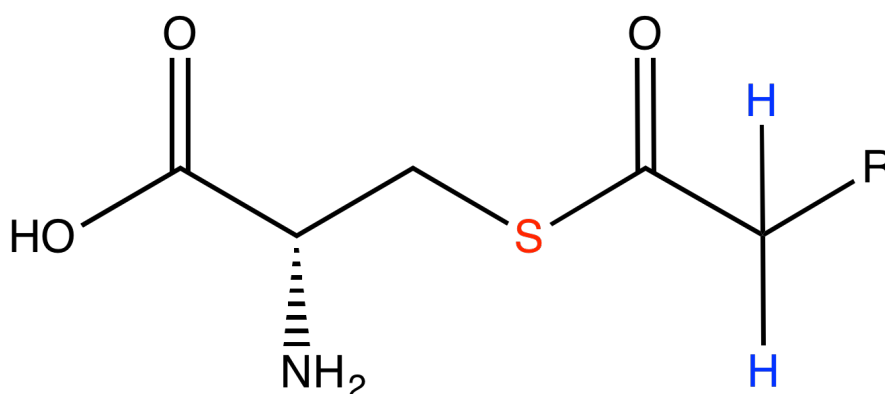


Figure 3: The different atoms in a palmitoylated protein which can be radiolabelled. The sulfur atom in red can be replaced by ^{35}S . The hydrogen atoms in the palmitate chain can be replaced by tritium atoms, ^3H .

number of molecules is important for the RRE approximation to be valid, and therefore also for validity of the tQSSA. Because of the many copies per cell and the many cells in the culture, we can therefore justify using the kinetics described in [Section 2.1.2](#) and [Section 2.1.3](#).

Another note of interest is that the experimental data itself is deterministic. When repeating the experiments they all reproduce a behaviour similar to that shown in [Figure 2](#). Further experimental data can be found in [Section 3.2](#) and [Section 4.2](#). Indeed, repeating the experiment is a deterministic measurement because it is a population of proteins within a population of cells that are being measured. Any single-cell noise will disappear in the culture of cells. Since we want to reproduce the deterministic exponential shape of the experiment in any case, the RRE is ideal. Furthermore, the dynamics observed are also relatively slow, on the scale of hours, whereas the time scale of most enzymatic reactions is less than a second [71].

2.2.3 *siRNA*

silencing RNA (siRNA) are short, double-stranded Ribonucleic Acid (RNA) molecules that are naturally occurring. They form part of the defense mechanism against viral infections. When viruses attempt to hijack the translation and transcription machinery of the host cell, siRNA are crucial in identifying foreign messenger RNA (mRNA) and eliminating them through RNA interference (RNAi). Once a foreign mRNA is identified by siRNA binding to it, it is cleaved. Subsequent proteins will recognise this mRNA as abnormal and degrade it, preventing expression of the protein it encodes.

Although naturally occurring, siRNA can also be used for experimental purposes. In the cell, it needs to efficiently bind to the mRNA of the gene that should be silenced. For each different gene to be silenced, an appropriate siRNA has to be tailored. If it binds sufficiently well, this siRNA can be used to alter protein levels in the cell. Although this is not a gene knockout, it has the similar effect of reducing the expression of a gene. In a gene knockout the encoding gene is actually removed from the cell's genome. However, in the case of siRNA the transcription of the gene will continue. In practice, an efficient siRNA can greatly reduce the levels of protein, achieving a similar effect to gene knockout.

siRNA can therefore be used to reduce the amount of a PAT. Given that there is significantly less enzyme present, the rate of palmitoylation catalyzed by the enzyme should also decrease. In practice, the enzyme should actually be active beforehand and the siRNA should have a strong enough effect on the enzyme levels for a difference to be observed. In this way, it is possible to study the effect that different PATs have on a protein. In a first step, this method can be used to identify which PATs influence protein stability. This method can also be used to characterize the effect of a PAT on palmitate turnover. By silencing a PAT and performing a ^3H pulse-chase

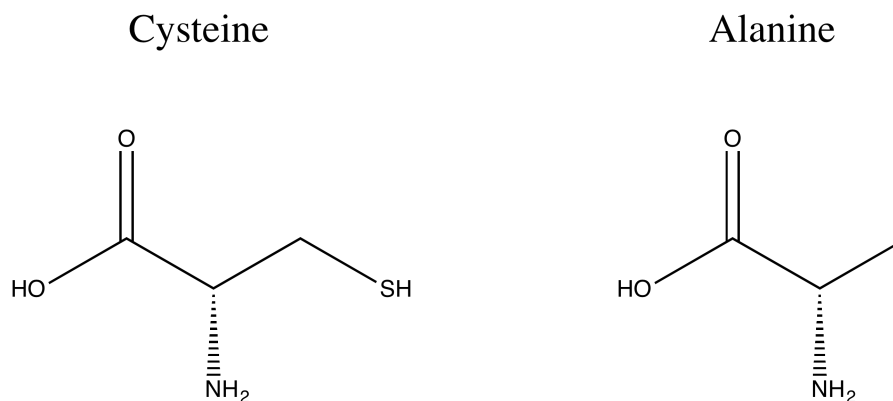


Figure 4: The amino acids Cysteine and Alanine. They are identical except for the thiol group that is absent from Alanine. By replacing a Cysteine with Alanine, it is possible to disable palmitoylation at a specific site.

experiment, it can be observed if palmitate turnover is affected. Silencing multiple PATs may also reveal additive or cooperative effects.

2.2.4 *Mutants*

Plasmids can be used to introduce genetic material into the cell that will be transcribed and translated into proteins. By modifying the coding sequence present in the plasmid, it is possible to introduce mutated proteins into the cell or overexpress endogenous proteins.

If the amino acid sequence and palmitoylation sites of a protein are known, mutation can be performed on these sites. By replacing a Cysteine with an Alanine, there is no longer a Sulfur atom available for the thioester bond to form and palmitoylation is no longer possible ([Figure 4](#)). By picking a cell line where the expression of a protein is low or absent, it is possible to study the behaviour of a mutant protein without the influence of the WT protein. Ideally, the plasmid and the transfection through which it is introduced into the cell are tuned so that the WT protein levels are reproduced. In this way, mutants can be prepared where one or several palmitoylation sites are disabled. This makes it possible to study the kinetics of palmitoylation and depalmitoylation of the individual sites. It also enables determining the effect of each palmitoylation site on properties such as stability and localization of the protein. This technique can be combined with siRNA to identify which enzymes palmitoylate which Cysteines.

2.2.5 *Cell Lines*

As mentioned in [Section 2.2.4](#), it is possible to pick different human cell lines in which to study proteins. In our case, the choice was made to gen-

erally use HeLa cells. This is because they have been successfully used in the past to study many human disease, such as polio [72] and cancer [73]. They have a long history of use in research and have therefore been well studied and characterised. Due to this history, the experimental methods for handling them in a laboratory setting are also mature and robust.

However, it mustn't be excluded that a protein may behave differently between human cell lines. As discussed in [Section 2.2.4](#), certain cell lines may not express the protein in question.

2.2.6 Fluorescence Microscopy

Fluorescence microscopy is a term regrouping several microscopy techniques which utilise fluorophores to perform imaging *in vivo* or *in vitro*. The general similarity between these methods is that of using fluorophores for observation. These may be specifically targeted to molecules of interest. This can be accomplished in multiple different ways. A mutant protein with an additional fluorescent sequence may be introduced into the cell as described in [Section 2.2.4](#). It is also possible to use antigens that strongly bind to the protein of interest. The fluorophore is then tagged onto the antigen. When a sample is illuminated with light of a specific frequency, the fluorophore first absorbs light and then re-emits it at a lower frequency. By filtering only the light emitted by the fluorophore, an image showing the presence and distribution of the targeted molecule is produced. An example is shown in [Figure 5](#), which shows the protein Calnexin in HeLa cells. From the image it can be observed that the majority of Calnexin is located in the ER.

By observing these images it is possible to get a rough estimate of the distribution of protein in a cell. In particular, which compartments contain the majority of a protein. Combining this technique with siRNA or performing it on mutant cells also makes it possible to verify the effects of these perturbations on protein localization.

2.3 MODEL STRUCTURE

2.3.1 Calnexin Background

Calnexin is a single-pass transmembrane protein of 67kDa. It is a part of the Ribosome-Translocon Complex (RTC) and thus aids in protein folding and quality control, an essential cellular function.

It has two palmitoylation sites which are situated close to its transmembrane domain. Palmitoylation of both these sites is performed by one PAT enzyme: zDHH6 [44]. The palmitoylation of Calnexin has been shown to be crucial for its association to the RTC, its localisation [44] and the confor-

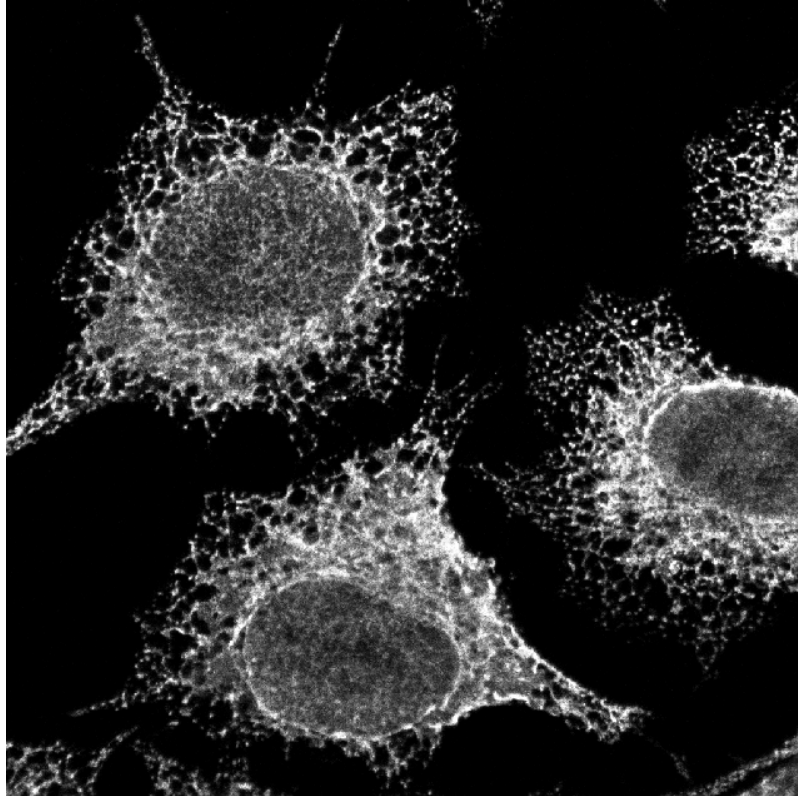


Figure 5: Fluorescence microscopy image of Calnexin in HeLa cells, reproduced with permission from Sandoz *et al.* [74]. The nuclei can be seen as grey oval shapes around which Calnexin, in the ER, is much whiter and extends towards the cell periphery.

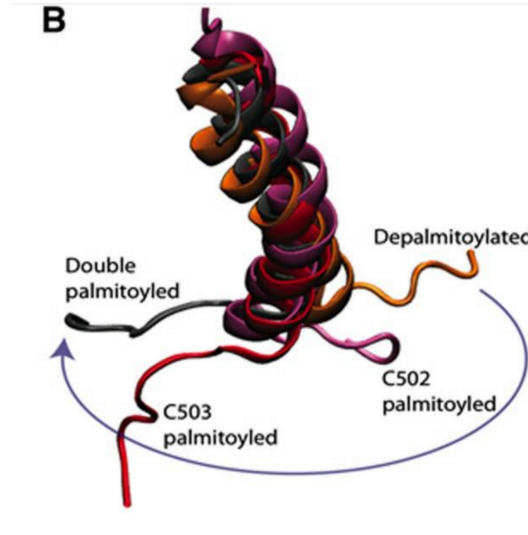


Figure 6: MD simulations showing the effect of palmitoylation on the conformation of Calnexin, figure reproducing with permission from Lakkaraju *et al.* [44]. The orientation of the cytosolic tail with respect to the transmembrane helix is palmitoylation-dependent. The effect of the two palmitoylation sites is different. Palmitoylation on both sites seems to have an additive effect.

mation of its cytosolic tail (Figure 6). It has also been shown that palmitoylation is used to shift Calnexin from its role in protein quality control to regulation of Calcium signalling [75]. In this way, the palmitoylation state of Calnexin is changed in response to ER stress [75], assigning more Calnexin to protein quality control.

2.3.2 Model overview

The model described here was previously published by Dallavilla *et al.* [43]. A schematic overview of the model structure is shown in Figure 7. The palmitoylation/de-palmitoylation cycles were setup similarly to the interconvertible cycles described by Goldbeter and Koshland [76]. This approach of building models from small interconvertible cycles has been successfully used to model signalling pathways such as the MAPK pathway [77, 78]. tQSSA kinetics described in Section 2.1.3 were used for all enzymatic reactions, in this case palmitoylation and de-palmitoylation. All other reactions, such as synthesis, transport and degradation, are modelled with mass action kinetics as described in Section 2.1.2.

In this model, Calnexin is first synthesized in the ER, producing an unfolded form *rCAL* shown in Figure 7. Folding then occurs following mass action kinetics, producing *fCAL*. At this point, the enzyme zDHHC6 can palmitoylate Calnexin on either the first or second site, producing *c1CAL* or

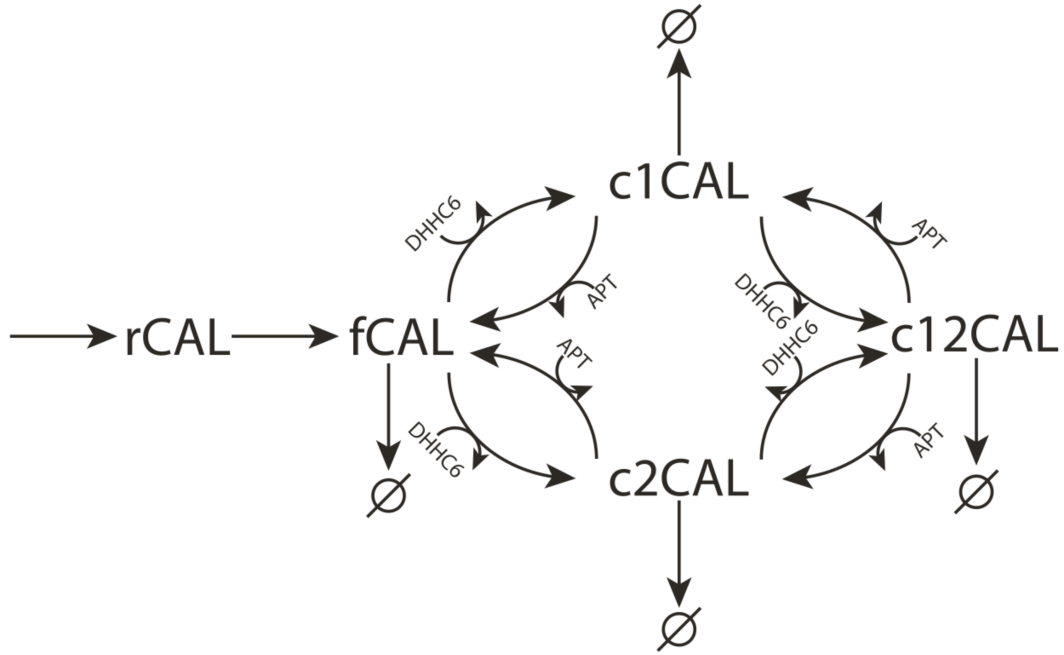


Figure 7: Illustration of the structure of the Calnexin model, reproduced with permission from Dallavilla *et al.* [43]. Synthesis leads to an unfolded form, 'rCAL', which must first fold into the mature protein 'fCAL'. Either palmitoylation site can be palmitoylated first. All palmitoylation reactions are catalyzed by zDHHC6, and a corresponding APT catalyzes the reverse reactions. Decay, palmitoylation and depalmitoylation all depend on palmitoylation state.

$c2CAL$ respectively. A final palmitoylation leads to the fully palmitoylated calnexin species $c12CAL$. De-palmitoylation can occur at any stage, and is also catalyzed by a single enzyme. Although the de-palmitoylating APT has not yet been identified, it is modelled as a single enzyme. In the case there are multiple APTs, this would merely increase the apparent enzyme activity, so the model would remain correct. Degradation can also occur at any time, except for the unfolded species, for which there is no degradation.

The species $fCAL$, $c1CAL$ and $c2CAL$ all compete for the $zDHHC6$ enzyme. While the species $c1CAL$, $c2CAL$ and $c12CAL$ all compete for the APT enzyme. This is easily taken into account using tQSSA kinetics ([Section 2.1.3](#)).

2.3.3 Calibration and Results

The Calnexin model was calibrated using ^{35}S and 3H radiolabelling experiments. These experiments used combinations of palmitoylation site mutation, $zDHHC6$ silencing and $zDHHC6$ overexpression. A Genetic Algorithm (GA) was used to calibrate the model, yielding a population of kinetic parameters. Specifically, NSGA-II was used [79]. The experiments were split into calibration and validation sets. The model was calibrated on the calibration experiments and the result was verified to reproduce the validation experiments. More details are available in the published paper [43].

Several interesting findings were obtained from the calibrated model. Most importantly it was shown that the fully palmitoylated form was the most stable, significantly more so than the other three forms. Furthermore, it was also shown that at steady-state, around 70% of the Calnexin was in this fully palmitoylated form. The turnover of palmitate was found to be surprisingly slow, on the order of 32 hours. This showed that 3H radiolabelling experiments were mostly describing the turnover of palmitate on the single palmitoylated species, which were much less stable.

2.4 GLOBAL SENSITIVITY ANALYSIS

The goal of GSA in the context of the Calnexin model is to extract as much knowledge as possible. Although useful insights were already obtained from simulating the model in conditions that were infeasible *in vitro*, we will show that additional knowledge can be gained using GSA. In this section, we will give an introduction to SA in general, how the Sobol indices used in GSA are derived, how to interpret them, and finally the application of GSA to the Calnexin model.

2.4.1 Introduction to Sensitivity Analysis

Sensitivity Analysis (SA) refers to a broad array of methods that may be used in the design and analysis of mathematical models. SA methods make it possible to study the relation between uncertainty in model outputs and uncertainty in input parameters [80]. Depending on the exact nature of the SA performed, it is possible to attribute model characteristics back to input parameters. SA can be used to verify current understanding of the model, in terms of seeing which input parameters are responsible for an observed output, as well as leading to further understanding of the model or directing attention towards areas of interest.

Historically, the early uses of SA took the local approach [80], also known as local sensitivity. These methods use small perturbations in input parameters around a certain value, for example the mean of an input parameter, and observe the resulting perturbation in model output. Local Sensitivity Analysis (SA) is very similar to the concept of elasticities and metabolic control coefficients [40], which are performed around steady-state. This local approach can be useful in elucidating which input parameters have the most impact in a certain regime.

Some background as to how local SA is performed is helpful to interpreting GSA results. Let Y be a scalar model output that is continuous and continuously differentiable. Local Sensitivity indices $s(Y, X_i)$ can be calculated by taking the derivative of a model output Y in relation to a single input parameter X_i , also known as first-order local sensitivity [81]:

$$s(Y, X_i) = \frac{\partial Y}{\partial X_i} \quad (21)$$

This describes the change in model output vs the change in model input. The larger the absolute value of the derivative, the more a small change in input parameter X_i will have a large effect on model output Y . Local sensitivity analysis is performed at a specific point in the input parameter space. Therefore its value will depend on the point that is chosen. An issue that may be encountered in more complex systems is the feasibility of analytically calculating the derivative in Equation 21. To circumvent this problem, the Finite Difference Method (FDM) may be used to approximate the derivative by a ΔX_i [81]:

$$s(Y, X_i) = \frac{\partial Y}{\partial X_i} \approx \frac{\Delta Y}{\Delta X_i} = \frac{Y(\Delta X_i + X_i) - Y(X_i)}{\Delta X_i} \quad (22)$$

The limit of this expression as ΔX_i tends to 0 gives the same result as the analytical derivative. The obvious issue here has to do with the accuracy of choosing a sufficiently small ΔX_i . If the derivative is not linear within

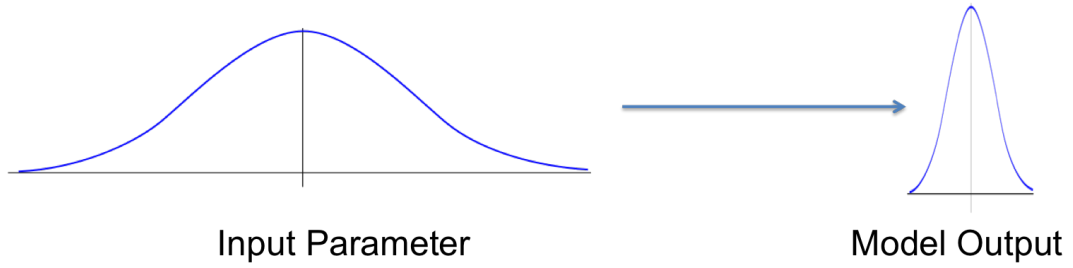


Figure 8: Variance propagation through a model. The variance of the input parameter is high, while the resulting variance in the model output is relatively low. This would indicate a parameter with low sensitivity.

ΔX_i of the reference point, different ΔX_i values lead to different results [81]. Although still useful, the local sensitivity approach has the limitation of not considering input parameters over their entire range [80]. From the late 1980s a new class of SA known as GSA was developed with roots in statistics [80]. Another definition of SA in this context is the study of how variance in the input parameters is propagated to variance in the model output [82]. An example of this interpretation is shown in Figure 8. As variance travels through the model from an input parameter to a model output, an amplification of this variance signifies a critical parameter, while a dampening is synonymous with a less important parameter.

Contrasting to local SA, the advantages GSA offers are a consideration of input parameters over their entire range, and thus an avoidance of possible local behaviour. GSA also avoids linearity assumptions [80], and may indicate more subtle effects such as interactions between different input parameters. It is for these reasons that GSA was chosen as a tool for further analysing the previously developed Calnexin model, both for the purpose of validation and attempting to extract any additional understanding of the underlying phenomena.

2.4.2 Sobol Sensitivity indices

Let Y be the model output of square-integrable function f acting on independent input parameters X_1 to X_k .

$$Y = f(X_1, X_2, \dots, X_k) \quad (23)$$

Because the function f is square-integrable, an expansion into terms of increasing dimensionality can be performed, also called a High Dimensional Model Representation (HDMR):

$$f = f_0 + \sum_i^k f_i + \sum_i^k \sum_{j>1}^k f_{ij} + \cdots + f_{1,2,\dots,k} \quad (24)$$

In this expansion each term is also square-integrable. The term f_0 corresponds to the mean of the model output Y . The $\sum_i^k f_i$ term corresponds to additive effects of single input parameters on the model output, i.e. no interactions with other parameters. Terms of higher order correspond to interactions between parameters, for example $\sum_i^k \sum_{j>1}^k f_{ij}$ corresponds to all two-way effects, that is, effects between exactly two parameters.

The Russian mathematician Ilya M. Sobol proved that if all the terms of the decomposition, apart from f_0 , have a zero mean then these terms are orthogonal in pairs and conditional expected values may be used to compute them [82]. For example, in the case of two input parameters i and j :

$$\begin{aligned} f_0 &= E(Y) \\ f_i &= E(Y|X_i) - E(Y) \\ f_j &= E(Y|X_j) - E(Y) \\ f_{ij} &= E(Y|X_i, X_j) - f_i - f_j - E(Y) \end{aligned} \quad (25)$$

In order to obtain the Analysis of Variance (ANOVA) HDMR decomposition, all that is needed is to take the variance of each side of Equation 24 [82]. The mean of all terms apart from f_0 has to be 0 for this step to be valid, in this case the terms are independent:

$$\text{Var}(Y) = \sum_i^k \text{Var}(f_i) + \sum_i^k \sum_{j>1}^k \text{Var}(f_{ij}) + \cdots + \text{Var}(f_{1,2,\dots,k}) \quad (26)$$

The so-called Sobol' indices are then obtained by normalising each term in this expansion by the total variance $\text{Var}(Y)$ of the model output. Combining this with the identities shown in Equation 25 yields:

$$S_i = \frac{\text{Var}(f_i)}{\text{Var}(Y)} = \frac{\text{Var}_{X_i}(E_{X_{\sim i}}(Y|X_i))}{\text{Var}(Y)} \quad (27)$$

The notation $E_{X_{\sim i}}$ indicates that the model output Y is averaged across all values of $X_{\sim i}$, i.e. all parameters other than X_i . The notation Var_{X_i} means that the variance is calculated across all values of X_i . S_i is known as the

main effect or first order sensitivity index of input X_i . It represents the fraction of the total output variance $\text{Var}(Y)$ that is attributed to variance in the input parameter X_i . The first order index neglects interaction effects of X_i with other parameters.

Interactions between parameters can be characterised by applying the same procedure to the higher order terms in Equation 24. However, as the number of input parameters increases, the number of terms in Equation 24 increases exponentially and it becomes very costly to calculate all the possible interactions effects individually. In order to avoid this issue, the total effect index was introduced. The total effect is the sum of an input parameter's first order effect as well as all higher-order effects in which it appears [82]. The total effect for input parameter i :

$$S_{t,i} = S_i + \frac{\sum_i^k \sum_{j>1}^k \text{Var}(f_{ij})}{\text{Var}(Y)} + \dots + \frac{\text{Var}(f_{1,2,\dots,k})}{\text{Var}(Y)} \quad (28)$$

In order to simplify calculations, the total effect of X_i can be considered as the total variance of the model output minus all other first order effects and interactions between parameters excluding X_i , all of this normalised with the output variance. This is equivalent to [82]:

$$S_t = \frac{\text{Var}(Y) - \text{Var}_{X_{\sim i}}(E_{X_i}(Y|X_{\sim i}))}{\text{Var}(Y)} = 1 - \frac{\text{Var}_{X_{\sim i}}(E_{X_i}(Y|X_{\sim i}))}{\text{Var}(Y)} \quad (29)$$

This makes computation faster by avoiding the need to individually calculate all interaction terms of the HDMR. The value $S_t - S_i$ can serve as a good indicator for the interaction effects of parameter X_i [82]. In models with a large number of parameters, it is more economical to only compute the first order and total effect indices first. If additional insight is required, interactions between specific parameters can then be selected for computation based on previous results, i.e. selecting parameters that are already shown to have some kind of interaction effects.

Furthermore, since both the first order index S_i and the total effect index S_t are normalised by the variance of the model output $\text{Var}(Y)$, this means that they are both in the interval $[0, 1]$.

2.4.3 Computing Sobol Sensitivity indices

Although we now know how the Sobol sensitivity indices are derived, the next step is their numerical calculation. If we look at the equations for the first order and total effect sensitivity indices (Equation 27, Equation 29), we can see they both involve the variance of a conditional expectation:

$$\text{Var}_{X_i}(E_{X_{\sim i}}(Y|X_i)) \quad (30)$$

This expression can be calculated by:

1. Generating N samples of all input parameters
2. Fixing X_i to its value in the first sample
3. Calculating the expected model output $E_{X_{\sim i}}(Y|X_i)$ by generating another N samples to assign values to the remaining parameters $X_{\sim i}$.
4. Calculating the variance of the model output Y by moving through the rest of the sampled values of X_i . Re-calculating the expected model output $E_{X_{\sim i}}(Y|X_i)$ each time
5. Once an array of expected model outputs is obtained, the variance of this can be calculated using $\text{Var}(Y) = E(Y^2) - E^2(Y)$

Although this would work, because there are kN^2 samples, the model producing the output Y from input parameters X would need to be evaluated kN^2 times. The quadratic term comes from having the same number of samples to calculate the inner expectation and the outer variance of the expectation. This is of course a problem when increasing the sample size, as the number of model evaluations will scale quadratically, quickly becoming very computationally intensive. *Saltelli et al.* appropriately called this a "brute force method" [82].

In order to address this issue the Saltelli method [82] can be used. Three matrices of size (N, k) are generated, A , B and C_i . Where N is the number of samples and k the number of parameters. Each row of these matrices corresponds to a parameter set generated by sampling. Each column represents a parameter, namely its values generated by sampling.

A and B are independently generated by sampling, while C_i is constructed from all columns of B , except column i that is taken from A . There is therefore one matrix C_i for each model parameter. The model outputs are then evaluated and stored in vectors of length N :

$$A = \begin{pmatrix} a_{1,1} & \cdots & a_{1,k} \\ \vdots & \ddots & \vdots \\ a_{N,1} & \vdots & a_{N,k} \end{pmatrix} \quad B = \begin{pmatrix} b_{1,1} & \cdots & b_{1,k} \\ \vdots & \ddots & \vdots \\ b_{N,1} & \vdots & b_{N,k} \end{pmatrix} \quad (31)$$

$$C_i = \begin{pmatrix} b_{1,1} & \cdots & a_{1,i} & \cdots & b_{1,k} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ b_{N,1} & \vdots & a_{N,i} & \cdots & b_{N,k} \end{pmatrix}$$

$$\begin{aligned} y_A &= f(A) \\ y_B &= f(B) \\ y_{C_i} &= f(C_i) \end{aligned} \quad (32)$$

The first order and total effect sensitivity indices are calculated in the following way:

$$\begin{aligned} S_i &= \frac{\text{Var}_{X_i}(E_{X_{\sim i}}(Y|X_i))}{\text{Var}(Y)} = \frac{y_A y_{Ci}/N - f_0^2}{y_A y_A/N - f_0^2} \\ S_t &= 1 - \frac{\text{Var}_{X_{\sim i}}(E_{X_i}(Y|X_{\sim i}))}{\text{Var}(Y)} = 1 - \frac{y_B y_{Ci}/N - f_0^2}{y_A y_A/N - f_0^2} \end{aligned} \quad (33)$$

Where terms such as $y_A y_{Ci}$ are vector products and f_0 is the mean of the model output calculated from y_A and y_B .

An approximate explanation of these formulae is that if X_i is an influential parameter, the model output between y_A and y_{Ci} will not change significantly, thus high values will be preferentially multiplied together, resulting in a large first order sensitivity index [82]. This approximate explanation is also valid for the total effect index, however as it is now only the values of X_i that change between the matrix B and C_i , if X_i is an influential parameter this will yield low values for the numerator and a high total effects index.

Compared to the brute force method, $2N$ evaluations would be needed for matrices A and B , followed by kN evaluations for matrices C_i if the sensitivity indices of all parameters were to be calculated. This gives a total of $N(2 + k)$ evaluations, with the significant advantage of being linear with respect to N . There is therefore no longer an increasingly large computational cost as the sample size is increased.

2.4.4 Sampling Methods

Although the Saltelli method for calculating Sensitivity Indices has previously been shown and discussed, the method by which samples are generated is an entirely independent choice. Random number generators are easily available; however, these do not always provide satisfactory results. Particularly when using crude Monte Carlo sampling, clustering may occur if the sample size is insufficient. The consequence being that the input space is not uniformly sampled and the results obtained not necessarily reliable. One way to avoid the issue of clustering is to use Design of Experiments (DOE) methods in order to ensure that sampling is performed more uniformly within the input space. The most simple and well known sampling method is the Full Factorial design. In the case of three input parameters that take values of either 1 or -1, there are a total of 2^3 possible combinations. As shown in Figure 9, the DoE in this case would correspond to evaluating the model at each corner of the cube. In this case the values 1 and -1 represent two different levels that the parameter values may take. To generalise: when performing a full factorial design for k parameters with s different levels, the total number of combinations is s^k [82].

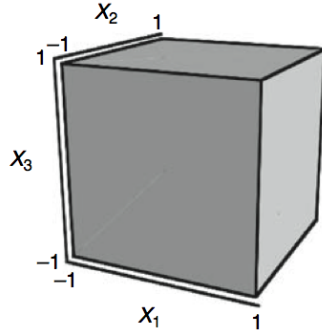


Figure 9: Representation of a Full Factorial design with 3 parameters and 2 levels, reproduced with permission from Saltelli *et al.* [82]

X_1	X_2		X_1	X_2
0	0	Randomise \rightarrow	0	0
1	1		3	1
2	2		2	3
3	3		1	2
0	0		0	3
1	1		2	1
2	2		3	2
3	3		1	0

Table 1: Example of generating a LH design for two parameters, four levels and eight samples

Full Factorial designs can be performed with more than two levels in order to capture non-linear effects of input parameters. However, as the number of parameters and levels increases, the cost associated with model evaluations rises exponentially. In order to reduce the computational cost, a fraction of these points can be selected. A Latin Hypercube (LH) design assures uniform sampling of the input space by first dividing each parameter range into a given number of sub-intervals or levels, an equal number of points is then sampled from each level. An example of this is shown in Table 1.

The advantage of randomising the two groups of simulations independently is that each group can be an independent estimate of the mean model output, making it possible to calculate a variance of the mean. This is similar in nature to performing several LH designs and calculating the uncertainty of the mean model output between them. Furthermore, these LH designs tend to converge faster than simple Monte Carlo sampling.

When considering only two levels that parameter values may take, FF methods make use of Hadamard matrices in order to quickly generate designs. Elements in a Hadamard matrix are either 1 or -1. These matrices are orthogonal and can be generated recursively from the smallest Hadamard matrix H_2 .

$$\begin{aligned} H_2 &= \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \\ H_{2^n} &= \begin{pmatrix} H_{2^{n-1}} & H_{2^{n-1}} \\ H_{2^{n-1}} & -H_{2^{n-1}} \end{pmatrix} \end{aligned} \quad (34)$$

These matrices have the interesting property of any two columns representing perpendicular vectors. Note also that when comparing two columns, half of the entries are the same, while the other half is different. These two properties also hold when looking at the rows of the matrix. In DOE terms, this is equivalent to a design of Resolution III [82]. It is possible to improve this using the following matrix concatenation:

$$M_{2^n} = \begin{pmatrix} H_{2^n} \\ -H_{2^n} \end{pmatrix} \quad (35)$$

By concatenating the original Hadamard matrix with its negative, additional combinations are created. This leads to a design of Resolution IV [82]. This can be related to the Full Factorial design in Figure 9. The cube represents the input space formed by any three parameters chosen from the total k input parameters. When picking any three parameters, there are an equal number of points in each corner of the cube.

It is possible to combine FF and LH designs in order to benefit from the advantages of both. In the LH, the main point of interest is that the input parameters are divided into levels along their range, and that there are an equal number of points in each level. Using a FF method it is possible to attain a Resolution IV design, which is equivalent to having the same number of points in each corner of the cube formed by any three input parameters. An illustration of these design criteria is shown in Figure 10.

As it is not immediately evident how to combine these two designs, the following outline adapted from [82] is a good starting point. For the case of four parameters, four levels and four simulations:

1. Generate the Hadamard matrix:

$$H_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \quad (36)$$

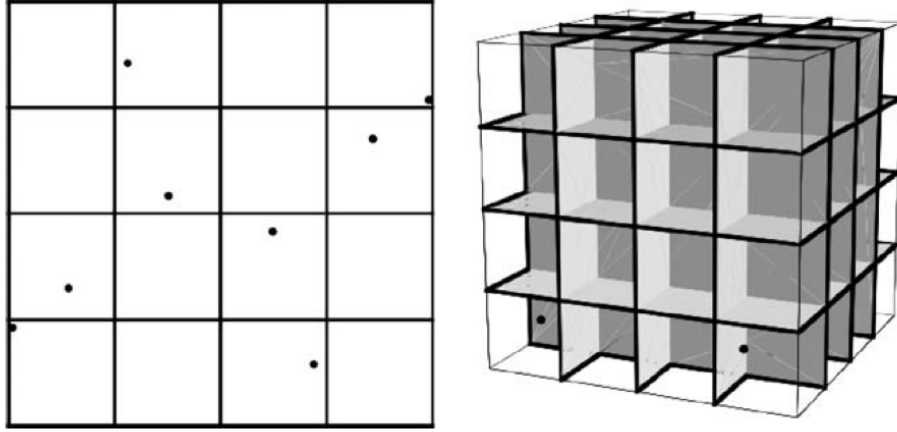


Figure 10: Combined LH/FF design, reproduced with permission from Saltelli *et al.* [82]. A Resolution III design is shown on the left and Resolution IV on the right. There are the same number of points in each level and corner between any two parameters (Resolution III) or any three parameters (Resolution IV).

2. Generate a LH design for four parameters, four levels and four simulations:

$$\text{lh} = \begin{pmatrix} 1 & 0 & 3 & 3 \\ 3 & 2 & 1 & 0 \\ 2 & 1 & 0 & 2 \\ 0 & 3 & 2 & 1 \end{pmatrix} \quad (37)$$

All that is needed here is to generate random permutations of the integer sequence from one to four

3. The two designs are combined in the following way
 - If the entry in the Hadamard matrix is 1, the value in the LH design is added to the number of levels, 4 in this case.
 - If the entry in the Hadamard matrix is -1, the value in the LH design is subtracted from (number of levels -1), 3 in this case

Applying this to the previously generated LH design gives:

$$\text{lh} = \begin{pmatrix} 5 & 4 & 7 & 7 \\ 7 & 1 & 5 & 3 \\ 6 & 5 & 3 & 1 \\ 4 & 0 & 1 & 5 \end{pmatrix} \quad (38)$$

4. The number of rows is doubled by calculating the complement: each value in the previous matrix is subtracted from $(2 \times \text{number of levels} - 1)$, 7 in this case. This gives the final combined LH/FF design:

$$l_h = \begin{pmatrix} 5 & 4 & 7 & 7 \\ 7 & 1 & 5 & 3 \\ 6 & 5 & 3 & 1 \\ 4 & 0 & 1 & 5 \\ 2 & 3 & 0 & 0 \\ 0 & 1 & 2 & 4 \\ 3 & 7 & 6 & 2 \\ 5 & 4 & 7 & 7 \end{pmatrix} \quad (39)$$

The initial LH design had 4 levels and 4 simulations; these numbers are doubled when calculating the complementary matrix, yielding a design with 8 levels and 8 simulations.

Although this example has a lower number of parameters and simulations for illustrative purposes, steps one and two can easily be modified or repeated to yield the much larger designs that are used in practice. Namely, the Hadamard matrix generated in step one simply needs to have as many columns as there are input parameters. The LH design in step 2 can be repeated and the matrixes concatenated to increase the number of simulations up to the desired amount.

There are many different ways to perform sampling, but the combined Latin Hypercube and Fractional Factorial design was chosen based on the description by *Saltelli et al.* [82].

2.4.5 Interpreting the Sobol indices

The first order sensitivity index S_i represents the average reduction in model output variance $\text{Var}(Y)$ when the parameter X_i is fixed. This is of particular interest when we want to measure parameters in order to improve model predictions. Because X_i is unknown, we have a distribution of its possible values. This distribution approximates the 'true' value of X_i . If X_i is measured, it could fall anywhere in an interval described by its distribution. Depending on the measured value, this may give larger or smaller reductions in variance. Hence S_i doesn't give a guarantee of the variance reduction of Y , but only a mean reduction that takes into account the distribution of X_i .

Considering only S_i doesn't describe the interactions that may occur between parameters. Let us consider [Figure 11](#) as an example. This figure describes irreversible Michaelis-Menten kinetics ([Equation 40](#)). It shows the reaction rate as a function of substrate concentration. The 'model' in this

case, has one output: the reaction flux v . There are three input parameters: the maximum reaction rate V_{\max} , the substrate concentration S and the Michaelis constant K_m .

$$v = V_{\max} \frac{S}{K_m + S} \quad (40)$$

Let us first consider the sensitivity of the reaction flux v with respect to the maximum reaction rate V_{\max} . From Equation 40 we can see that any change in V_{\max} will always have an effect on the reaction rate v . In other words, the sensitivity of v with respect to V_{\max} is independent of the values of the other parameters. Translating this into the Sobol sensitivity indices means that $S_i = S_t$ for the sensitivity of v with respect to V_{\max} . There are no other parameters in the model that will amplify or dampen the effect of V_{\max} on v .

However, the effect of substrate concentration S on reaction rate v is not independent of other parameters. Depending on the value of K_m , the reaction will be operating close to or far from saturation. When K_m is large compared to S , the enzyme is unsaturated. Therefore, more substrate will accelerate the rate of reaction. Conversely, when K_m is low compared to S , the enzyme is saturated, and increasing S will no longer increase the reaction rate. In the saturated regime $S_t > S_i$ for the sensitivity of v with respect to S . This means that the effect of S on v is no longer independent, but depends on the value of K_m .

Looking only at the first order sensitivity index S_i makes it possible to quickly identify single parameters that will reduce uncertainty in a model. In this case, uncertainty is the variance of the model output Y . A high value of $S_t - S_i$ indicates strong interaction effects. This means two or more parameters interact in order to influence the model output. A parameter that has a very high S_t but S_i of zero still indicates a parameter critical for model output, but it is through interactions with other parameters that it modulates the output. If S_t is high but S_i is zero it means there is no reduction in model uncertainty on average when the parameter is measured. But because S_t is high, model uncertainty can be reduced if other interacting parameters are also measured.

2.4.6 Application to the Calnexin model

When calibrating the Calnexin model *Dallavilla et al.* [43] assigned a fitness score to each parameter set based on its deviation from experimental data. This score is calculated in the following manner:

$$\text{score} = \frac{(y_{\text{simulated}} - y_{\text{experimental}})^2}{\sigma_{\text{experimental}}} \quad (41)$$

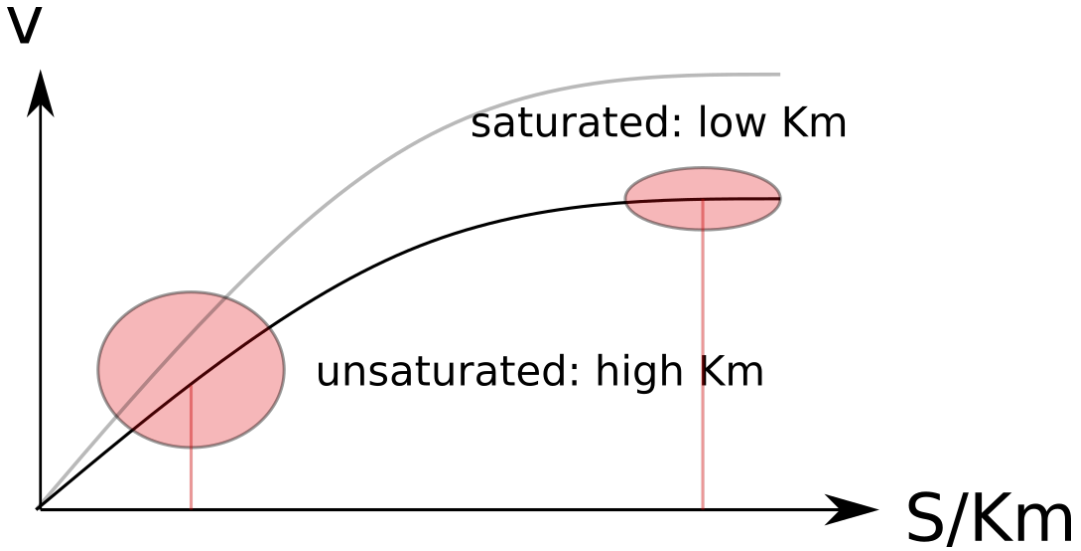


Figure 11: Illustration of reaction rate v as a function of the normalised substrate concentration S/K_m . The grey line shows the effect of increasing V_{max} .

The values of y correspond to the amount of labelled Calnexin present at different time points of the pulse-chase experiments. The standard deviation of the experimental results is obtained by repeating the in-vitro pulse-chase experiments [43]. The difference between the in-silico and the in-vitro experiments is therefore a value to be minimised when fitting the model parameters. A higher score indicates a worse fit with experimental data, and a lower one a better fit.

When performing GSA, the bounds of each input parameter must be specified. Two different ways of setting these bounds were tested:

1. Constrained: the minimum and maximum of each parameter were the same as those in the population obtained with NSGA-II.
2. Wide: The maximum and minimum of each parameter from the constrained bound are either multiplied or divided by 10 in order to define the new maximum and minimum. This generates a larger input space.

It is critical to underline that the parameter sets generated in GSA and those from the NSGA-II are not the same. When using the constrained parameter bounds, the only similarities are the maxima and minima of all parameters. The input space defined by these bounds will be sampled during GSA, as described in [Section 2.4.4](#). This means that new combinations of parameter values will be generated that are not present in those produced by the NSGA-II.

[Figure 12](#) shows the effect on the fitness score when using the wide bounds. The solid lines represent the mean of all parameter sets, while

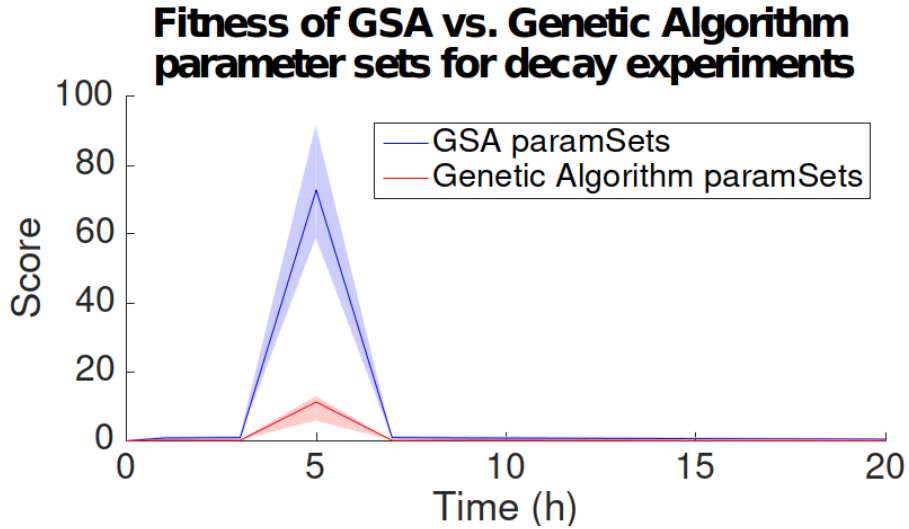


Figure 12: Fitness scores as a function of time for NSGA-II parameter sets and the wide sets generated in GSA. The parameter sets given in [43] were used as a reference. The solid lines show the mean, and the shaded area shows the first and third quartile. The GSA parameter sets were obtained by randomly sampling 100 parameters sets using the wide parameter bounds.

the edges of the lighter areas show the first and third quartiles of the model output. Increasing the upper and lower bounds for all parameters has the general effect of worsening the fit in relation to experimental data.

This worsening of fit when choosing wider parameter bounds is something to be expected. NSGA-II is an optimisation method that attempts to find a region in the input space where the fit is the best. It is therefore unsurprising that the fit be worsened when the bounds of this region are widened. The difference in fit is particularly pronounced at the 5-hour mark of the pulse-chase experiments. Indicating a very good point to start should the fit need to be further improved.

Figure 13 shows the same graph as Figure 12, except that the constrained parameter bounds are used for the GSA. When choosing the same parameter bounds as those of the NSGA-II optimisation results, the fitness score is not noticeably affected. This demonstrates a non-trivial aspect of the NSGA-II results. As long as the bounds of the input space are kept the same, any combination of parameter values within this space will not significantly impact the fitness score. In other words, the NSGA-II has identified an entire input space with an adequate fit, rather than just individual parameter sets. The results of the NSGA-II can therefore be reduced to the maximum and minimum values of each parameter, rather than the multitude of sets that are returned.

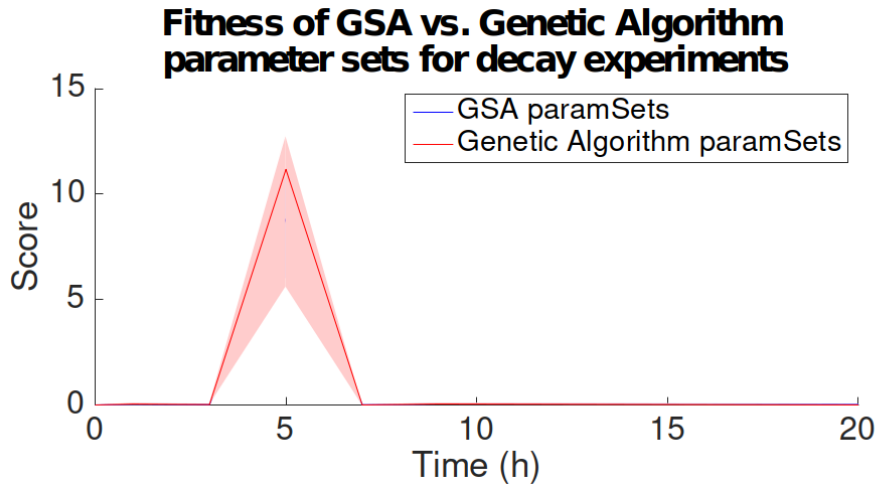


Figure 13: Fitness scores as a function of time for NSGA-II parameter sets and the constrained sets generated in GSA. The parameter sets given in [43] were used as a reference. The solid lines show the mean, and the shaded area shows the first and third quartile. The GSA parameter sets were obtained by randomly sampling 100 parameters sets using the constrained bounds.

Figure 14 supports this conclusion. When looking at the in-silico pulse-chase experiments, the results produced with parameter sets from the NSGA-II or constrained GSA are virtually the same; the NSGA-II is robust in the constrained space. This explains why the fitness scores are also the same, as they are based on the pulse-chase curves. It can be noted that the shift between the two regimes of the biphasic decay occurs around the 5-hour mark. It is also at the 5-hour mark that the fit with experimental data is the worst. This suggests that the model struggles to accurately capture this shift in decay phases, although the fit once firmly in either regime is much better.

The following results are generated by taking 16'000 samples using the previously described combined LH/FF sampling method. All samples are taken from the constrained parameter space. The sensitivity indices are calculated using the Saltelli method. The entire process of GSA is repeated three times in order to calculate the standard error of all sensitivity indices.

Figure 15A shows first order and total effect sensitivity indices of the four most important model parameters with respect to half-life of the total amount of ^{35}S -labelled protein. The total amount of ^{35}S -labelled protein is calculated by simply adding the amounts of all labelled species. This metric therefore does not take into account the relative amounts of each Calnexin species and the change in population during the decay.

As can be immediately seen, there are only two parameters that impact the half-life of the total amount of ^{35}S -labelled protein. Furthermore, the

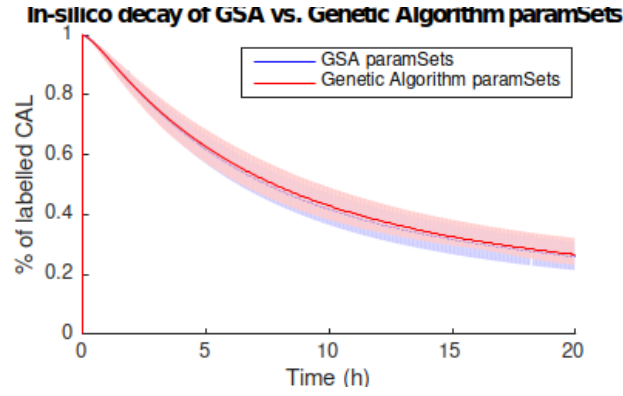


Figure 14: In-silico pulse-chase experiment, with NSGA-II parameter sets or constrained GSA sets. The parameter sets given in [43] were used as a reference. The solid lines show the mean, and the shaded area shows the first and third quartile. The GSA parameter sets were obtained by randomly sampling 100 parameters sets using the constrained bounds.

first order and total effect sensitivity indices have similar values, and their sums are close to one. This indicates that within the constrained input space, the model behaves close to perfectly additive. There is little to no interaction between parameters; the effect of a change in one parameter does not depend on the value of other parameters. $k_{\text{decay}3}$ is the decay constant of the double palmitoylated species $c_{12}\text{CAL}$. It is not surprising that this is the most influential parameter; at steady-state it is mostly this species which is present [43], therefore most of the decay flux is controlled by $k_{\text{decay}3}$. The second most important parameter is the maximum speed of palmitoylation, $V_{\text{max}_{\text{fwd}}}$. This is related to the smaller pool of $f\text{CAL}$ that is also present at steady-state [43]. It appears that in the parameter space that was studied, the decay constant associated with $f\text{CAL}$, $k_{\text{decay}1}$, does not have much of an effect on the total protein half-life, although $V_{\text{max}_{\text{fwd}}}$ definitely does. This could be explained by the parameter $V_{\text{max}_{\text{fwd}}}$ having a large effect on the species present, a higher value would mean more flux going from $f\text{CAL}$ to $c_{12}\text{CAL}$, leading to a bigger pool of $c_{12}\text{CAL}$ with a different decay constant. Although this result may be somewhat surprising, it is not incomprehensible, as the input space that is chosen can have a large effect on GSA results.

Figure 15B shows sensitivity indices for the steady-state concentrations of both $f\text{CAL}$ and $c_{12}\text{CAL}$, supporting the previous explanation that $V_{\text{max}_{\text{fwd}}}$ changes the species present. Although not the most important parameter, $V_{\text{max}_{\text{fwd}}}$ does change the steady-state concentrations of both $f\text{CAL}$ and $c_{12}\text{CAL}$. It is thus possible that changing this parameter can change the distribution of Calnexin species, changing the relative amount of $c_{12}\text{CAL}$ and thus the stability of the labelled Calnexin pool.

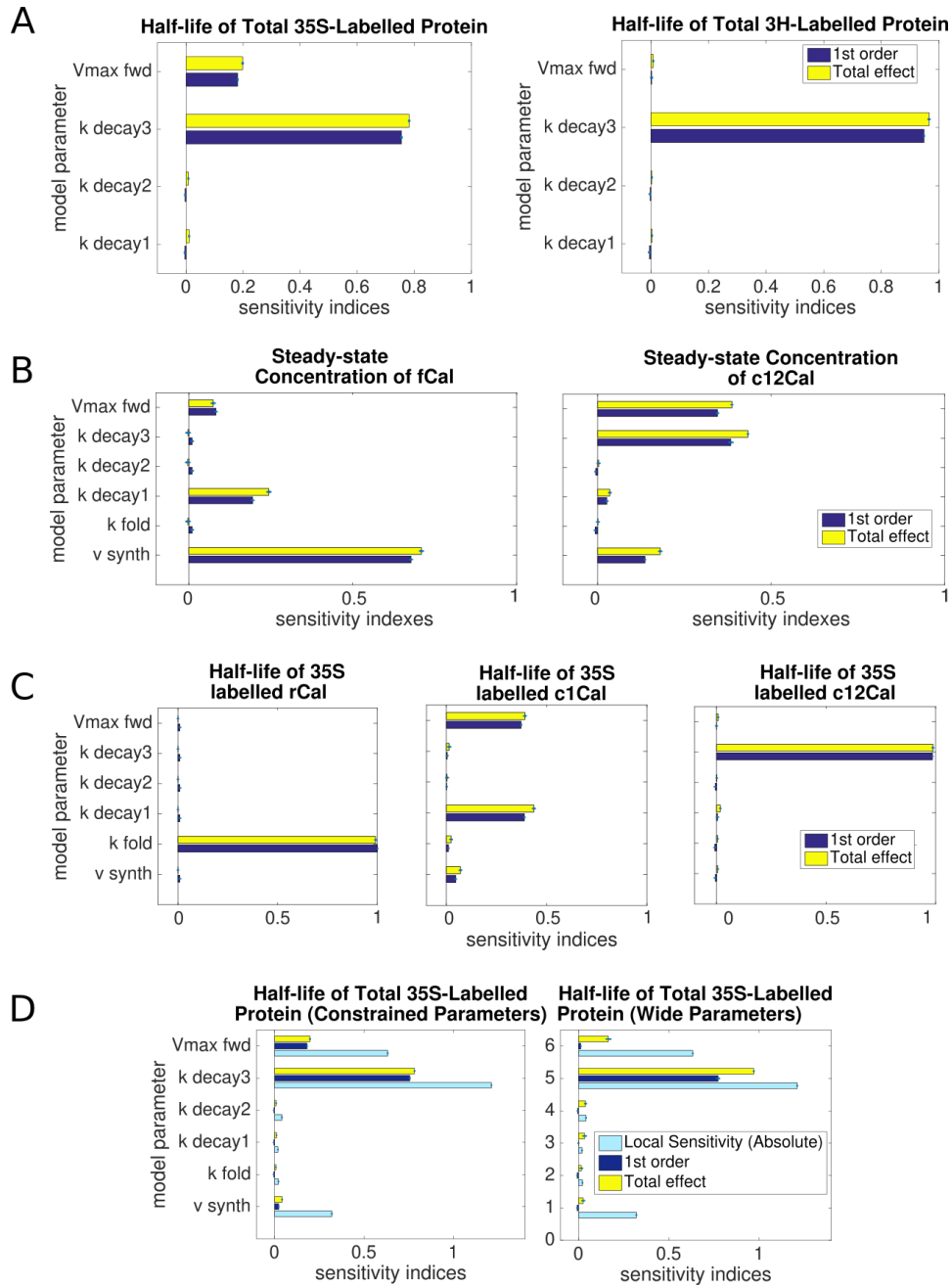


Figure 15: GSA was performed on various characteristics of the Calnexin model using 16'000 samples generated with the LH/FF sampling method. The mean and standard deviation of the sensitivity indices are shown. A) First order and total effect sensitivity indices for the half-life of total ^{35}S -labelled and ^3H -labelled Calnexin. Constrained parameter bounds. Other parameters with sensitivity indices of 0 are omitted for clarity. B) First order and total effect sensitivity indices for the steady-state concentrations of *fCAL* and *c12CAL*. Constrained parameter bounds. C) First order and total effect sensitivity indices for the half-lives of *rCAL*, *c1CAL* and *c12CAL*. D) First order and total effect sensitivity indices for the half-life of total ^{35}S labelled Calnexin. The difference between wide and constrained parameters bounds is shown. Results from local SA are also shown for comparison.

Figure 15A is similar to Figure 15A, however only the ^3H -labelled protein is now considered. As $f\text{CAL}$ is non-palmitoylated, it is not labelled with ^3H -palmitate and therefore not a part of the total ^3H -labelled protein. This can be seen from the close to null sensitivity indices of $V_{\max_{\text{fwd}}}$; now that $f\text{CAL}$ is not counted in total labelled protein this parameter no longer has an effect. It is virtually only labelled $c12\text{CAL}$ that is present, and its decay constant is the only significant parameter for half-life. This is a good example of the usefulness of the two different labelling methods; different sub-sections of the Calnexin palmitoylation model can be targeted for study.

Figure 15C shows another key property of the half-lives of various species in the model; these are often determined by only one or two parameters. As long as the parameters stay within the defined input space, it is possible to easily tune the half-life of specific species without necessarily affecting that of others. If these parameters would need to be tuned beyond their constrained bounds, it would be easy to perform GSA with new parameter bounds and study the results, although the change in parameter space may mean the model no longer accurately reproduces the experiments.

The previous GSA results were generated using the constrained parameter space defined by the NSGA-II. The following section widens this space in order to identify potential changes in model behaviour. Figure 15D shows the same objective function as previously discussed; the half-life of the total amount of ^{35}S labelled protein. However, the difference in GSA results with wide and constrained parameter bounds is shown. The results give a good insight of what may happen when the input space is enlarged. According to the total effects index, the most important parameters remain the same and the ranking does not change. However, the first order effect of $V_{\max_{\text{fwd}}}$ entirely disappears. As there are only two significant parameters, this means that there is an interaction between $k_{\text{decay}3}$ and $V_{\max_{\text{fwd}}}$; the impact that either one has will depend on the value of the other one. This intuitively makes sense, as the speed of palmitoylation determines how much Calnexin ends up in the fully palmitoylated $c12\text{CAL}$ form, which then decays with rate constant $k_{\text{decay}3}$. Although some very slight interaction was visible previously, it is much more pronounced in the wider parameter space. Local sensitivity was also performed in the middle of the input space (the mean of every parameter) in order to have a comparison. Absolute values are taken to simplify comparison when the derivative is negative. The local sensitivity produces a similar ranking to GSA, although it does not capture any interaction effects. It also assigns a high sensitivity to the protein synthesis rate v_{synth} , which GSA reports as being negligible.

Table 2 shows the different parameter bounds for the decay constants $k_{\text{decay}1}$ and $k_{\text{decay}3}$, associated with $f\text{CAL}$ and $c12\text{CAL}$ respectively. The important aspect here is that in the constrained space they do not overlap. This means that $f\text{CAL}$ will systematically decay faster than $c12\text{CAL}$. However, once the bounds are widened, these parameters do overlap. When

Parameter	Constrained		Wide	
	Maximum	Minimum	Maximum	Minimum
$k_{\text{decay}1} [\text{h}^{-1}]$	0.1589	0.0760	1.589	0.00760
$k_{\text{decay}3} [\text{h}^{-1}]$	0.0268	0.0101	0.268	0.00101

Table 2: Parameter bounds for $k_{\text{decay}1}$ and $k_{\text{decay}3}$ with either constrained or wide bounds.

$V_{\text{max}_{\text{fwd}}}$ is thus changed, it is entirely possible that the $f\text{CAL}$ and $c12\text{CAL}$ species have similar rates of decay, depending on the value of $k_{\text{decay}3}$. This means that the effect of $V_{\text{max}_{\text{fwd}}}$ will depend on whether $k_{\text{decay}3}$ is higher or lower than $k_{\text{decay}1}$. In the wider parameter space, the total effect sensitivity index of $k_{\text{decay}1}$ is slightly higher than in the constrained space, but still too small to be significant. This is another surprising result, and may be due to the vast majority of Calnexin being in the $c12\text{CAL}$ state, therefore very little decay flux goes through $k_{\text{decay}1}$. This explanation does not exclude the possibility of interaction between $V_{\text{max}_{\text{fwd}}}$ and $k_{\text{decay}3}$ either, the population will change, but its half-life is still mostly dependent on the decay constant of $c12\text{CAL}$.

2.5 CONCLUSION AND FUTURE DIRECTIONS

Many additional aspects of the previously built Calnexin model have been further investigated. These include:

- Half-lives of various Calnexin species
- Steady-state concentrations
- Parameter interactions
- Effects of parameter bounds on model behaviour

In the process of performing GSA, the fitness functions of different parameter sets were studied, revealing some of the more subtle qualities of NSGA-II. Namely that, in this case, the result returned by the NSGA-II is in fact an entire parameter space in which the fit to experimental data is good, not just a population of isolated parameter sets.

One particular point of the pulse-chase experiments was highlighted in which the fit of the simulated data was significantly poorer than elsewhere. This point coincided with a shift between the two regimes of biphasic decay. If the fit of the parameters would need to be further improved, starting at this point would be the most logical choice.

The results concerning half-life show that it would be very easy to engineer the half-life of different species in the model by acting on only a handful of parameters. This is also true of other model properties, such as steady-state concentrations. GSA can identify parameters and subsets that determine different model properties. It is even possible to modify the half-life of certain species, such as *c1CAL* or *rCAL*, without affecting any of the others.

Finally, it was shown that the chosen input space can have a significant effect on the results generated by GSA. In this particular case, the ranking of the parameters according to total effect did not change, but interactions between parameters appeared and some first order indices were reduced to negligible values. These results were compared to those of Local Sensitivity. Although in both cases the ranking was the same, local sensitivity identified important parameters that GSA classified as unimportant. This is most likely because GSA takes into account the entire parameter space, determining on average if a parameter has an effect on a model output. Furthermore, GSA supplies additional information in terms of the interactions between parameters. This is one of its key strengths, being able to distinguish between first order effects of a parameter and interactions with other parameters. If desired, it is possible to compute two-way, three-way or higher parameter interactions. As models become more complex and the potential for interactions increases, GSA can yield significantly improved holistic understanding of how models work.

As was mentioned in the introduction; Calnexin is an interesting protein not only because of its twin palmitoylation sites, but also because it is palmitoylated by the enzyme zDHHC6, which is itself palmitoylated. Building a more complex model that includes zDHHC6 and a corresponding APT is an exciting step on the way to creating a system which more closely resembles what occurs in the cell. [Figure 16](#) shows what a palmitoylation network could resemble when including zDHHC6 and APT in addition to Calnexin. The number of input parameters as well as the model outputs that may be of interest increases significantly in this more complex model. There may also be a higher potential for interactions between different parameters of the model due to this increased complexity, making it a perfect candidate for the GSA algorithms already used for Calnexin. The ability to distinguish parameter interactions may become even more valuable in this context. Further refining and improving the efficiency of the algorithms will also be necessary in order to deal with the much larger number of input parameters, which increases about one order of magnitude. Although efforts have been made towards creating such a model, it remains a work in progress.

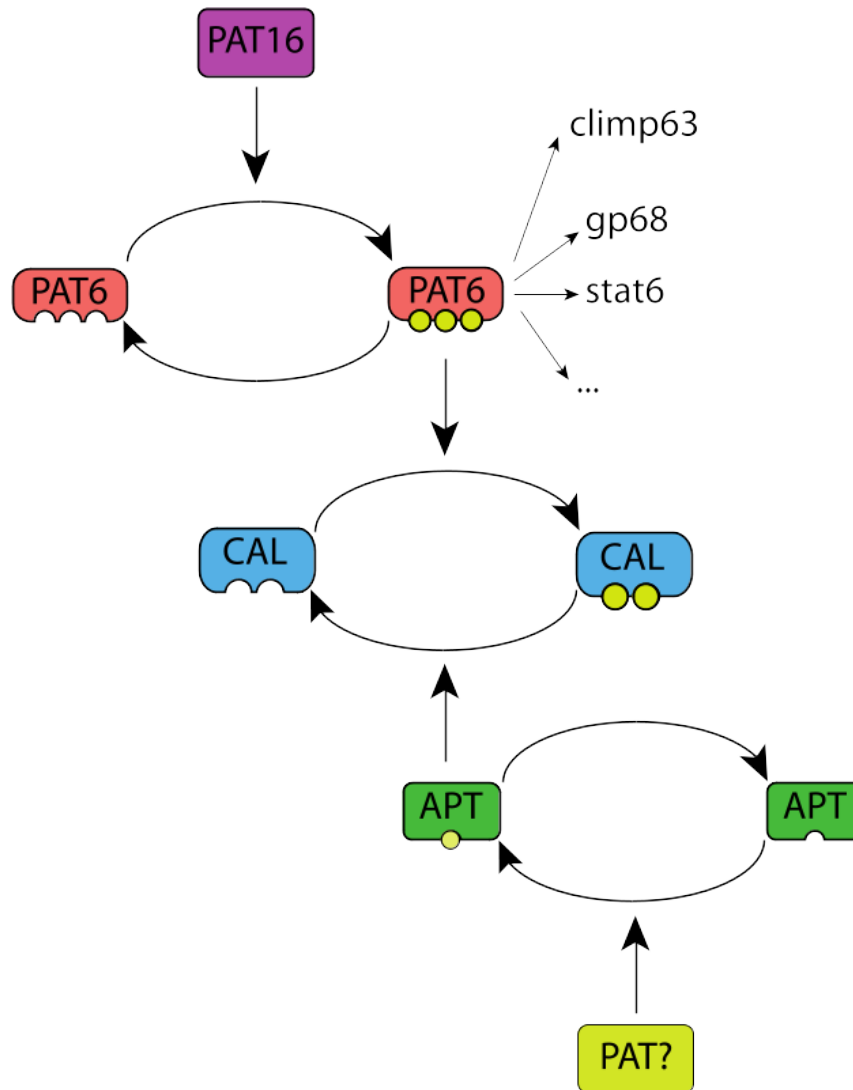


Figure 16: Representation of a possible palmitoylation network of Calnexin, including both PAT6 (zDHHC6) and its corresponding APT. Other palmitoylation targets of PAT6 are also shown

3 CLIMP63

The results in this chapter were used to prepare a manuscript in collaboration with several authors. The manuscript was submitted to a preprint server [74] and is currently awaiting re-submission to a journal. All figures in this chapter are either taken from or based on this manuscript. The experiments were performed by various members of the Van der Goot Lab, namely: Patrick Sandoz, Laurence Abrama, Sylvia Ho, Béatrice Kunz.

In this chapter we will create a kinetic model of the Cytoskeleton-linking Membrane Protein 63 (CLiMP63). We will show the different types of experimental data that were successfully integrated into the model and the difficulties in properly calibrating the model. Although we started with a similar approach as that used to create the Calnexin model in Chapter 2, we encountered several difficulties and had to radically rethink the model construction and calibration phases. After an overhaul of the methodology, we obtained a model that could accurately reproduce all of the experimental data. We were able to estimate half-lives of the seven different forms of CLiMP63, and describe how this protein was regulated through palmitoylation. Most importantly, our approach showed that the multimers formed by CLiMP63 are crucial to its function.

3.1 CLIMP63 BACKGROUND

Cytoskeleton-linking Membrane Protein 63 (CLiMP63) is 63 kDa single-pass transmembrane protein. It was discovered in the 1990s [83, 84], where it was both identified as an antigen for specific monoclonal antibodies and a highly palmitoylated protein during cell mitosis.

It's main roles include generation and maintenance of ER sheets, where it acts as a spacer [74], and binding microtubules in order to anchor the ER to the cytoskeleton. It also has other roles which are less well studied. These include being recruited to the ribosome-translocon complex [74], as well as acting as a receptor for multiple different ligands at the PM [85–87].

Disregulation of CLiMP63 has been linked to multiple different types of cancer, where its upregulation has been linked to poor prognoses [88–90]. It has also been found that CLiMP63 is downregulated in patients that have undergone surgery for abdominal aortic aneurysm [91].

In addition to palmitoylation, CLiMP63 can also undergo phosphorylation, but not glycosylation, being devoid of any glycan modifications [84]. Human CLiMP63 has a cytosolic tail that is 106 residues long, and has a single palmitoylated cysteine just six residues away from its transmembrane do-

main. Furthermore, this cytosolic tail is predicted to be intrinsically disordered by the IUPred *in silico* computational tool [92].

It was also found that CLiMP63 is an abundant protein, at least in the ER, where its copy numbers were estimated between 100,000 and 400,000 [2, 93, 94].

CLiMP63 is therefore a protein with many different roles, most of them at the ER, where it is primarily located. A small subpopulation is also located at the PM, where it is a receptor for multiple different ligands.

3.2 EXPERIMENTAL DATA

In this section, we will talk about the experimental data that was available to build and calibrate the CLiMP63 model. The experimental data can be broadly classified into two categories. The first is used in order to inform choices when defining the model structure, whereas the second is used in determining parameter values during model calibration.

Initial experiments were performed in order to determine if any cross-talk between phosphorylation and palmitoylation existed, and also to identify the PATs targeting CLiMP63. These experiments belong to the first category, used for defining model structure.

All the experimental methods that were described in Section 2.2 were also used for studying CLiMP63. Namely, radiolabelling with either ^{35}S or ^3H , siRNA to reduce enzyme levels, overexpression to increase enzyme levels and mutants to disable palmitoylation sites. However, in this case there were two PATs targeting CLiMP63: zDHHC6 and zDHHC2. In contrast with Calnexin, CLiMP63 has only one palmitoylation site. These series of experiments belonged to the second category, being used during calibration to determine the values of model parameters.

Details concerning the experimental protocols can be found in the preprint by Sandoz *et al.* [74].

3.2.1 Cross-talk with Phosphorylation

Because CLiMP63 is also phosphorylated, it was important to determine whether there was a cross-talk between phosphorylation and palmitoylation. Especially as palmitoylation occurs on the cytosolic domain, which is where microtubules attach. As phosphorylation prevents microtubules from attaching to CLiMP63, microtubules were either stabilized using the drug taxol or their depolymerization promoted with nocodazole. Using ^3H labelling, the incorporation of palmitate after two hours was compared under these three conditions. It was observed that palmitate incorporation wasn't affected (Figure 17), and therefore that it appears to be independent from microtubule interaction and phosphorylation.

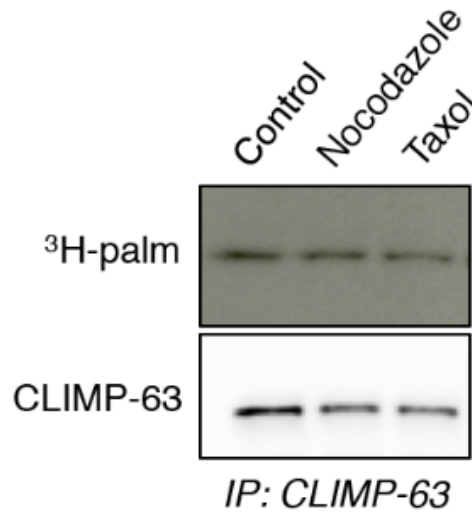


Figure 17: ^3H labelling for two hours showing the effect of promoted microtubule stabilization using taxol, or increased microtubule depolymerization using nocodazole. Reproduced with permission from [74]

3.2.2 Identifying palmitoylating enzymes

It was previously reported that CLiMP63 is a target of the PAT zDHHC2 [95]. However, as the majority of CLiMP63 is localised to the ER and zDHHC2 is at the PM [74], it was thought that there must be another PAT targeting CLiMP63. Experiments were thus performed in order to confirm the activity of zDHHC2 and identify any other PATs targeting CLiMP63. This was done by first performing ^3H palmitate labelling for two hours in control conditions, and then silencing zDHHC6 or zDHHC2, for a total of three experiments. The results in Figure 18 clearly show that both these enzymes contribute to CLiMP63 palmitoylation, although zDHHC6 appears to be more important for palmitate incorporation.

3.2.3 Determining palmitoylated fraction

The Acyl-RAC experimental protocol was used to determine the fraction of palmitoylated CLiMP63 at steady-state. This protocol works by first collecting the protein from the cell at steady-state. The free cysteines are then labelled with a fluorescent chemical. The experiment is then repeated by adding hydroxylamine, which cleaves the thioester bonds, effectively depalmitoylating occupied cysteines. By comparing the fluorescent readout from both these experiments, it is possible to estimate the fraction of palmitoylated cysteines. If no fluorescence is observed when hydroxylamine is not added, this means that all cysteines are palmitoylated. On the other hand, if the fluorescence readout is the same with or without hydroxy-

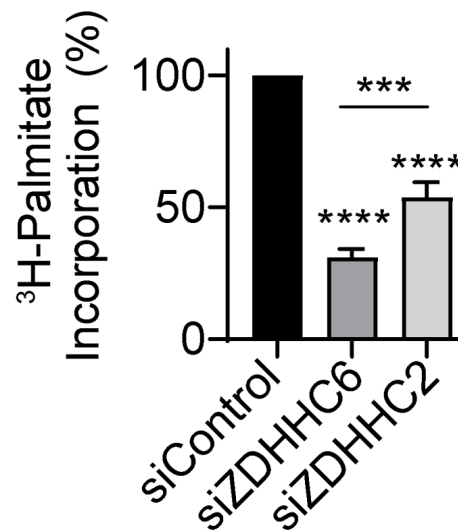


Figure 18: Quantification of ^3H labelling of CLiMP6₃ (N=4) done for two hours in control conditions and with siRNA targeting either zDHHHC6 or zDHHHC2. Mean, standard deviation and p-values (**p < 0.01, ****p < 0.001) are shown. Reproduced with permission from [74].

lamine, it means that none of the cysteines are palmitoylated. A schematic of the process is shown in Figure 19.

The Acyl-RAC protocol was applied to CLiMP6₃, the results are shown in Figure 20. With an acylated fraction of 87 percent, the vast majority of CLiMP6₃ is palmitoylated in cells.

3.2.4 Quantifying surface population

In order to observe how localisation of CLiMP6₃ is affected by palmitoylation, surface biotinylation was used. This method works by covalently attaching biotin to the protein of interest. Using streptavidin beads, for which biotin has a particularly high affinity, the tagged proteins can be purified. A western blot can then be performed in order to identify the different proteins and their relative amounts.

One advantage is the particularly small size of the biotin molecule, which decreases the possibility for interference in the functioning of the tagged protein.

Many different biotinylation agents exist. Some require enzymes in order to be attached to proteins, while chemical biotinylation agents have a reactive group that is attached to the biotin. This reactive group influences the solubility of the biotinylation agent. NHS-Biotin is soluble in water but

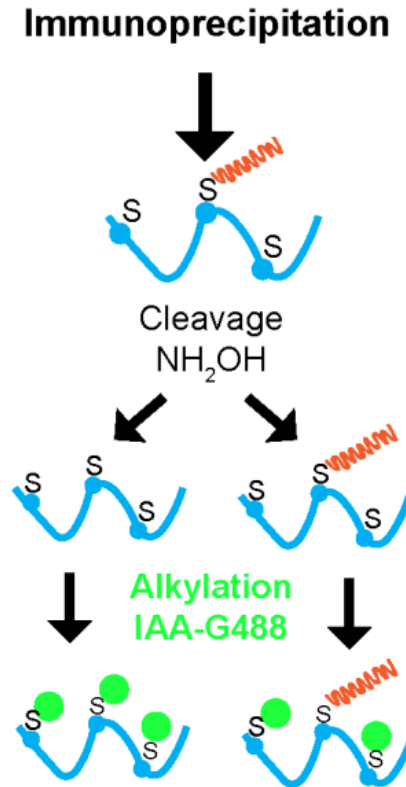


Figure 19: Schematic representation of the Acyl-RAC experimental protocol. Immunoprecipitation is used to capture the protein of interest. The palmitoylated cysteines can then be cleaved using hydroxylamine (NH_2OH). Fluorescent labelling is done using iodoacetamide-oregon-green-488 (IAA-OG488). Comparing the fluorescent readout with that of proteins that weren't cleaved with hydroxylamine makes it possible to calculate the fraction of acylated protein. Reproduced with permission from [74].

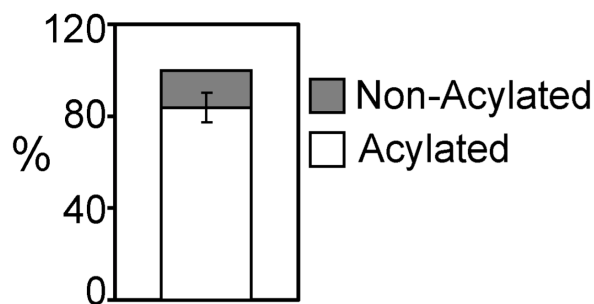


Figure 20: Results of the Acyl-RAC experiment on CLiMP63 (N=3), showing the estimated palmitoylated (acylated) fraction. Error bars represent the standard deviation. Reproduced with permission from [74].

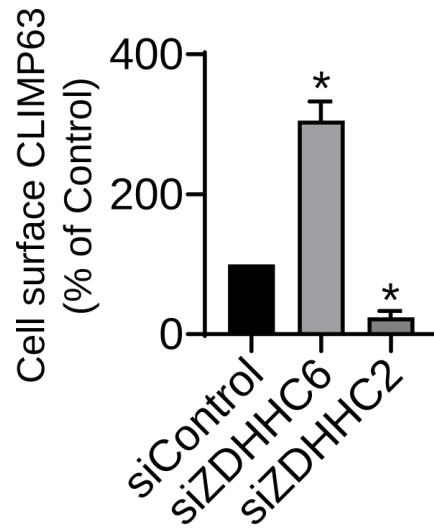


Figure 21: Quantification of CLiMP₆₃ population at the surface using surface biotinylation (N=3). Mean, standard deviation and p-values are shown (**p < 0.01). The effect of silencing zDHHc6 and zDHHc2 are shown relative to Control conditions. Reproduced with permission from [74].

does not penetrate the cell membrane, meaning that it will biotinylate all proteins on the cell membrane, but no intracellular proteins. In this way, it is possible to estimate the relative change of CLiMP₆₃ at the surface under different conditions, namely when silencing zDHHc6 or zDHHc2, as shown in Figure 21.

3.2.5 Fluorescence Microscopy

As with Calnexin (Section 2.2.6), fluorescence microscopy was also used to study CLiMP₆₃. From the images shown in Figure 22, it can be seen that CLiMP₆₃ is also predominately localised in the ER. However, what can also be noted is that the distribution is not significantly affected when one of the palmitoylating enzymes, zDHHc6, is knocked out. This information can in fact be used either for model calibration or validation. This can be done by assigning a penalty when the fraction of CLiMP₆₃ in the ER is under a certain threshold.

3.3 MODELLING GOALS

Given the previous knowledge described in Section 3.1 and the experimental data in Section 3.2, several questions could be formulated.

One of these involves localisation of CLiMP₆₃, which has important roles in two distinct cellular compartments, the ER and the PM. This raises the question of how both roles in the different compartments are balanced. An-

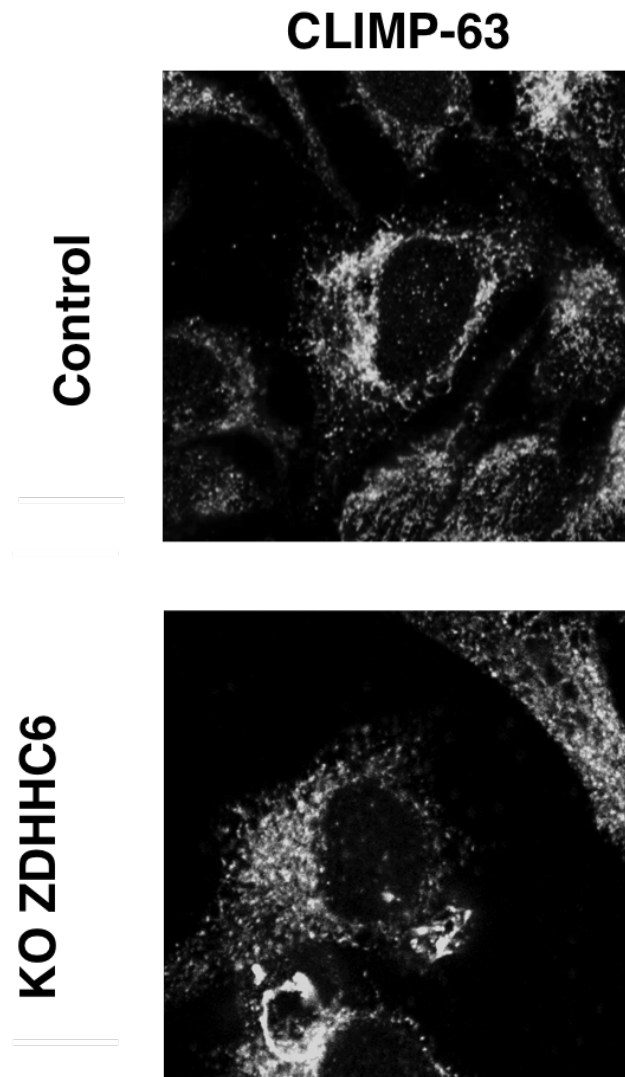


Figure 22: Confocal immunofluorescence microscopy images of CLiMP₆₃ in HeLa cells. Reproduced with permission from [74]. Control conditions and zDHHHC6 knockout are shown. The nucleii can be identified by dark circle-like shapes, around which CLiMP₆₃ shows up as white dots.

other question is how this distribution is controlled and regulated by zDHHC6 and zDHHC2. Experimental data shown in [Figure 21](#) suggests that both zDHHC6 and zDHHC2 modulate the localisation of CLiMP63. However, the exact mechanism by which this happens isn't clear. Furthermore, it is not easy to verify that a proposed mechanism is consistent with the rest of the experimental data. A model can give quantitative answers to these questions. Not only can it propose a mechanism by which localisation is controlled, but it can also quantitatively estimate how the different enzymes affect the stability and trafficking of CLiMP63. As multiple sets of experimental data can be integrated in the calibration process, it is possible to create a model that is consistent with the entirety of the experimental data.

As is shown in [Figure 31](#), the apparent stability of ^{35}S labelled CLiMP63 is high, while palmitate turnover is also high. This would suggest that palmitoylation of CLiMP63 is highly dynamic, while the majority is palmitoylated, it is constantly undergoing depalmitoylation-palmitoylation cycles. This begs the question of how this dynamic palmitoylation fits in with regulation and localisation of CLiMP63. In particular, which species are present and how do they evolve over time. Given that CLiMP63 is present in two compartments and has one palmitoylation site that may be occupied or not, there are thus at least four different CLiMP63 species.

Although ^{35}S radiolabelling experiments give a general sense of how stable a protein is, in practice they only monitor newly synthesized proteins. These must undergo a folding and maturation process, meaning that they do not necessarily have the same properties as the bulk of the protein. This problem of partial observability is the same in the case of ^3H palmitate labelling, it is only newly attached palmitate which is monitored. If there is a large bulk of the protein that undergoes very little palmitoylation, its palmitate turnover will not be measured by these experiments. This is a recurring challenge when building these models of palmitoylation, the experimental data is only in fact a partial observation of the system. In order to better understand how a protein is regulated, it is very useful to estimate the properties of the individual CLiMP63 species. For example, using the model in order to estimate *bona fide* half lives of the different CLiMP63 species helps in understanding how regulation works, and is not possible using only experimental data and current knowledge.

In addition to estimating half-lives and the evolution of different species, we will use the model to estimate the palmitoylation and depalmitoylation fluxes of different species. We aim to identify which species are preferentially palmitoylated and depalmitoylated, and how this differs between compartments.

3.4 MODEL CONSTRUCTION

As was discussed in [Section 3.1](#), it was already known that CLiMP63 is present in the ER and PM. The results in [Section 3.2.2](#) identified two enzymes that palmitoylate CLiMP63: zDHHC6 and zDHHC2. Given that these enzymes were localised in the ER [\[74\]](#) and PM, respectively, a rough model structure could already be constructed, as shown in [Figure 23](#).

This model included the two compartments in which CLiMP63 was known to be present, as well as the CP through which it needs to be transported to reach the PM. In this model, a constant synthesis flux first leads to an unfolded form of CLiMP63, from which it folds to a mature form according to first order kinetics. Palmitoylation can only occur once CLiMP63 has folded. All transports and decay reactions are modelled according to first order kinetics. All palmitoylation and depalmitoylation reactions are modelled with tQSSA kinetics, as was the case in the Calnexin model ([Section 2.3.2](#)).

Although no APT targeting CLiMP63 has been identified, we assigned an APT enzyme in each compartment where palmitoylation by a PAT occurred. Furthermore, as no other PAT apart from zDHHC6 and zDHHC2 has been identified so far, it was assumed that no palmitoylation or depalmitoylation occurs in the CP. Since the CP was only a transitory compartment, no decay reactions were included, according to the principle of parsimony.

3.5 MODEL CALIBRATION

In this section we will go over the process of calibrating the CLiMP63 model. We will talk about some of the different numerical methods and software tools that were used. We will also show the unsuccessful attempts and the solutions implemented to reach a satisfactory model calibration.

3.5.1 *Rule-based modelling*

Once the initial model structure had been established, we then needed to translate this into a computational form in order to be numerically tractable. In the Calnexin model the code for the system of ODEs was written by hand. One of the difficulties in writing code for these models is keeping track of the different labelling states. Because there are several different radioactive labels, the number of combinations rises significantly. Every label, and whether it is on or off, must be accounted for in the model. For a protein with a single palmitate attached to it, this means there are $2^2 = 4$ different labelling combinations possible, arising from the combination of the ^{35}S and ^3H labels. For a protein with three palmitates attached, this rises exponentially to $2^4 = 16$.

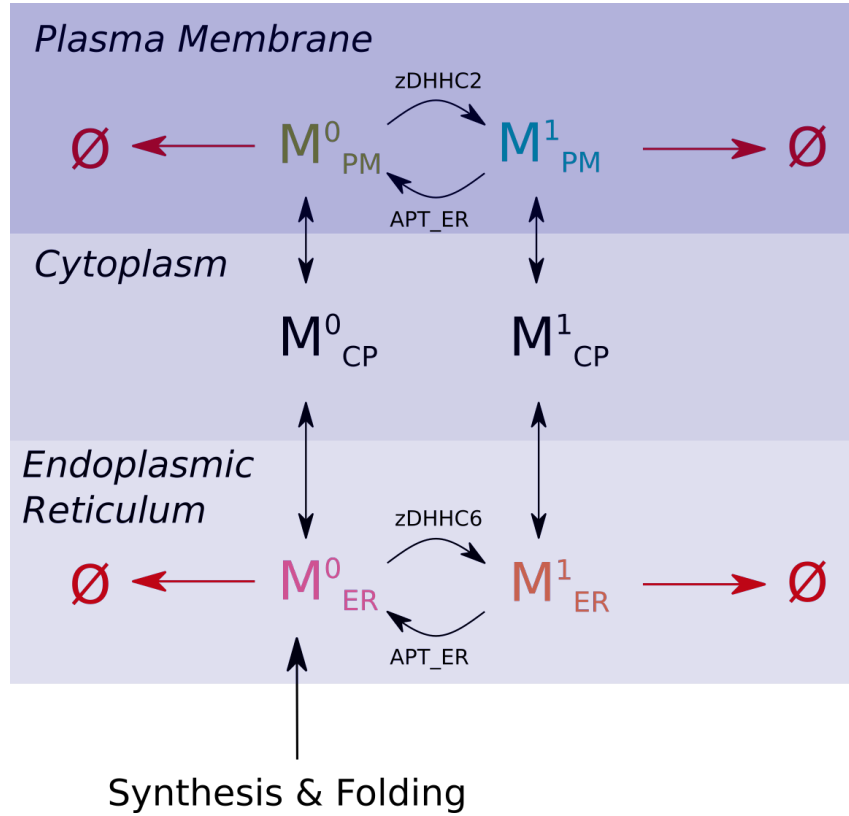


Figure 23: Initial CLiMP63 model structure, containing three compartments and a total of six species (not counting the unfolded form of CLiMP63). M denotes the monomeric CLiMP63 protein, the subscript its cellular localisation and the superscript its palmitoylation state, 0 for non-palmitoylated and 1 for palmitoylated. Synthesis, folding, transport and decay are modelled according to mass action kinetics. Palmitoylation and Depalmitoylation are modelled with tQSSA kinetics.

In order to better manage this combinatorial complexity, we opted for a rule-based modelling approach using RuleBender [96, 97]. In this approach rules are used instead of writing out each reaction multiple times to take into account all the different labelling combinations. Rules are similar to reactions in that they specify products, substrates and a reaction rate, but they also match multiple substrates and products based on the expressions used. For example, a palmitoylation site is represented by C for the Cysteine to which a palmitate is attached. This Cysteine can either be unoccupied $C \sim 0$, palmitoylated $C \sim P$ or palmitoylated with a labelled palmitate $C \sim PL$. It may also have the sulfur present in its amino acids labelled $^{35}\text{S} \sim 1$ or not labelled $^{35}\text{S} \sim 0$. Writing a reaction rule, for a generic molecule or protein M:

$$M(C \sim 0) \rightarrow M(C \sim P) \quad (42)$$

Corresponds to attaching a non ^3H labelled palmitate to a protein. The $M(C \sim 0)$ expression will match any proteins that are non-palmitoylated. Because the ^{35}S labelling property is not specified, it will match both ^{35}S labelled and non-labelled proteins and treat them equivalently. In fact this rule corresponds to writing out two reactions, one with $^{35}\text{S} \sim 1$ and the other with $^{35}\text{S} \sim 0$.

Writing the rules for protein decay:

$$\begin{aligned} M(C \sim 0) &\rightarrow \emptyset \\ M(C \sim P \sim PL) &\rightarrow \emptyset \end{aligned} \quad (43)$$

The first rule matches any protein that is non-palmitoylated and the second any protein that is palmitoylated (P) or palmitoylated with a labelled palmitate (PL). In this example, 2 rules are written instead of 6 explicit reactions. In this way, rules can be written which are independent of the labelled state of the protein, while generating a system of ODEs which still keeps track of the labelled and non-labelled population in all their combinatorial complexity.

Combining these rules with observables allows easy interfacing with the complex system of ODEs that is created. For example:

$$M(^{35}\text{S} \sim 1) \quad (44)$$

Would match all proteins that are ^{35}S labelled, i.e. contain a ^{35}S Cysteine, irrespective of their palmitoylation state and ^3H labelling. Various software packages exist which implement this sort of rule-based model building, for example BioNetGen[96] or PySB[98].

As we have seen, rule-based modelling is very well suited to describing radioactive labelling experiments. In particular, it facilitates handling the combinatorial explosion that occurs due to the large amount of labelling combinations.

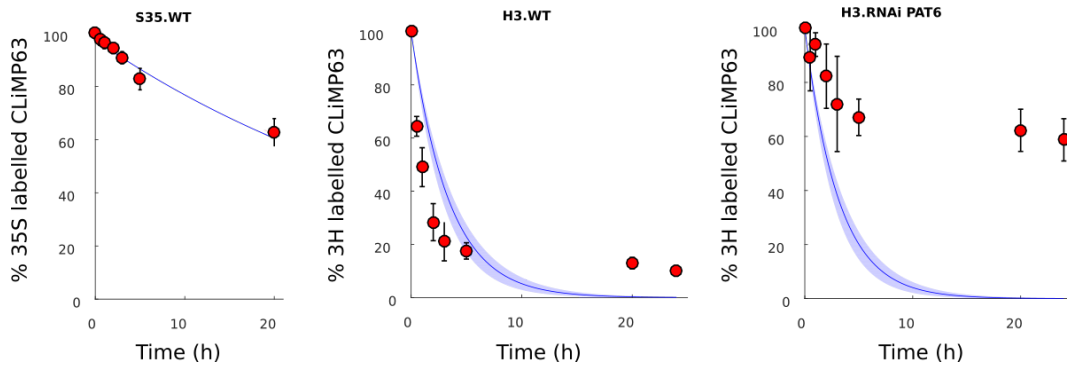


Figure 24: Subset of the calibration experiments ($N=5$) obtained after the initial calibration attempts. Experimental data points are shown in red with the error bars showing standard deviation. Model output is shown with the blue curves. The shaded areas represent the first and third quartile of the model output.

3.5.2 Initial Calibration attempts

The same general approach to model calibration was taken as that used for the Calnexin model [43]. Specifically, the NSGA-II optimisation algorithm was used to minimise a sum of squares objective function that was designed to minimise the difference between experimental data and model output. The biggest difference was that a rule-based modelling approach was used (Section 3.5.1) in order to facilitate model generation and make it more tractable.

Although this approach had been successful for the Calnexin model, initial results for the CLiMP63 model were not satisfactory. Reproducing the pulse-chase experiments was not successful, as shown in Figure 24. It was not only that the decay curves were not reproduced, but some of these experiments were missing the typical biphasic shape which was evident in the data. This pointed to a larger underlying issue than just a failure to find a suitable parameter space. Indeed, in order to be able to reproduce a biphasic decay like that of the ^3H WT chase experiment (Figure 24), there need to be at least two different CLiMP63 species that are decaying with different rate constants. The model outputs in Figure 24 are not a biphasic decay, meaning that there is only one predominant species decaying with a single rate constant.

Furthermore, the distribution obtained with the model did not agree with microscopy experiments (Section 3.2.5), which indicated that a majority of the protein was localised in the ER both in WT conditions and upon zD-HHC6 knockout. Figure 25 shows that the model had less than half of the CLiMP63 at the ER in WT conditions. Upon zDHHC6 knockout, the model had only a negligible fraction of the protein left in the ER. Clearly, such a drastic change would have been visible in the microscopy experiments.

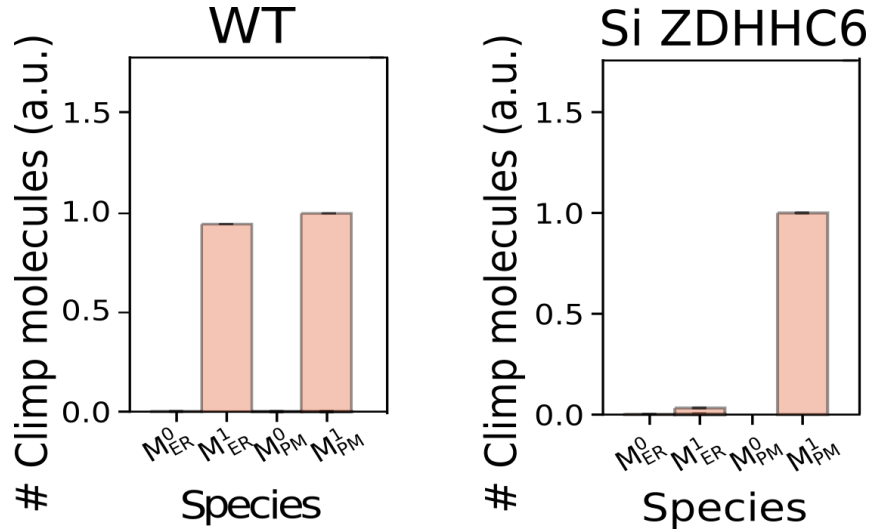


Figure 25: Population distribution obtained during the initial model calibration. The mean of 100 parameter sets is shown. The error bars represent the standard deviation. Both normal (WT) conditions and silencing zD-HHC6 are shown. In both cases, the majority of CLiMP₆₃ is at the PM, whereas when silencing zDHHHC6 almost none is present in the ER. Reproduced with permission from [74].

Additionally, the surface biotinylation experiments (Figure 21) showed that although the majority of the protein stays in the ER when zDHHHC6 is silenced, there is a significant increase of the amount at the PM. The model, however, showed no sensitivity of the PM subpopulation to zDHHHC6 silencing.

Given these large disparities between the model and experimental data, it was decided that an in-depth overhaul of the approach should be undertaken. Although the core structure of the model would be kept the same, multiple changes would be individually tested, as well as trying out a different optimisation algorithm and error function.

3.5.3 Optimisation algorithms

The most obvious place to start improving model calibration is the optimisation algorithm. Generally, there are two parts to a heuristic optimisation algorithm like NSGA-II. The first is the fitness function, also called error function. This defines a certain fitness score based on the model parameters. In our case, the model parameters are used to simulate a system of ODEs with the goal of reproducing experimental data. The fitness function then produces a score based on the similarity to experimental data. The second part is the heuristic optimisation algorithm itself. This determines how the iterative optimisation takes place, how the next iteration is chosen based on the

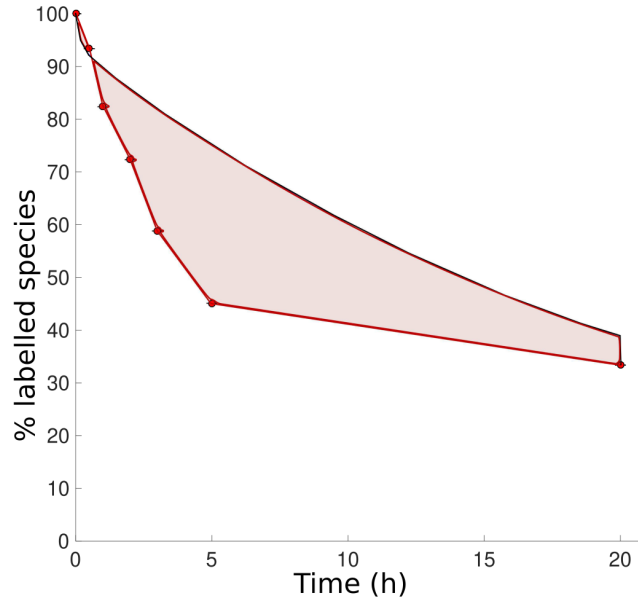


Figure 26: Error function used for comparing experimental data to model output. Experimental data is shown with red dots, and is used to create a piecewise linear curve. The area between the experimental and model curves is the output of the error function. Even though there is no data between five and twenty hours in this example, using the area ensures that the last data point is properly taken into account.

current one, and generally how the parameter population and its associated fitness scores are taken into account when exploring the parameter space.

Although a sum of squares error function worked well for calibrating the Calnexin model, this approach breaks down when the experimental data points are not evenly spread out in time. As can be seen in [Figure 26](#), there is a grouping of data points between one and five hours, followed by a single data point at twenty hours. When using a sum of squares error function, a disproportionate amount of weight is given to the data points grouped at the beginning of the experiment. If the model reproduces the data accurately before the five hour mark, and then diverges afterwards, the fitness score will not be significantly affected. Although there is only one data point at the twenty hour mark, it provides crucial information on the rate of decay between five and twenty hours. Instead, we can use the experimental data points to construct a piecewise linear curve and then calculate the area between this curve and the model output curve. In this way, all experimental data points will be taken into account even if they are not evenly spread out in time, as the area in [Figure 26](#) illustrates.

We can also change the heuristic optimisation algorithm itself, hopefully improving the results obtained. Calibration of the Calnexin model used the NSGA-II algorithm, which belongs to the general class of GAs. These work in a way inspired by the natural selection of individuals in a population, il-

illustrated in [Figure 27](#). The idea is to mimic evolution as it occurs in nature to find an optimal parameter space. Each set of parameters is synonymous with an individual of the larger population. Using the previously defined fitness function, we can assign a numerical fitness score to each individual. Based on this fitness score, only a fraction of the population survives an iteration of the algorithm. The surviving individuals then reproduce, creating new individuals by exchanging some of their parameters. Random mutations also occur at this point. The fitness score of the new population is evaluated and the algorithm repeats until a stopping criteria is met. This can be a target fitness, or merely a fixed number of iterations of the algorithm.

Algorithms based on Covariance Matrix Adaptation (CMA) work in a similar way as GAs. They also use a population that is iteratively evolved based on the fitness value of the individuals. However, instead of randomly crossing over parameters during reproduction, CMA builds a covariance matrix from the fittest individuals of the population and updates it every iteration. This matrix is used instead of the reproductive step. In this way, the algorithm has some concept directionality. Although there is not a clear vector indicating the direction of optimality, new individuals are sampled based on the covariance matrix. This ensures that parameters which must have certain values relative to other parameters stay in this required range. Whereas parameters which are free to vary irrespective of other parameters may do so. An illustration of what this looks like is shown in [Figure 28](#). We chose to use the CMA-ES algorithm for calibrating the CLiMP63 model.

3.5.4 *Alternative model structures*

In order to investigate additional reasons why the initial calibration was not successful, it was decided to try different model structures. Based on the literature and background of the CLiMP63 protein ([Section 3.1](#)), three additional model structures were explored. These are all shown in [Figure 29](#).

The first alternate structure introduces a recycling compartment at the PM. This is to simulate the constant sorting and renewal of transmembrane proteins in the cell membrane. In this compartment, CLiMP63 is merely taken up and then released back to the PM. Transport to and from the recycling compartment follow first order kinetics and have their own rate constant. The rate constant is also different based on palmitoylation state of CLiMP63. No decay or palmitoylation is possible in this compartment, as it is only meant to simulate CLiMP63 being erroneously included in the sorting process of membrane proteins.

The second introduces zDHHC6 in the CP. Although zDHHC6 is primarily localised in the ER, it is possible that it may be transported in the same vesicles as CLiMP63, due to their both being transmembrane proteins. The concentrations of zDHHC6 in the CP and ER were made independent, as

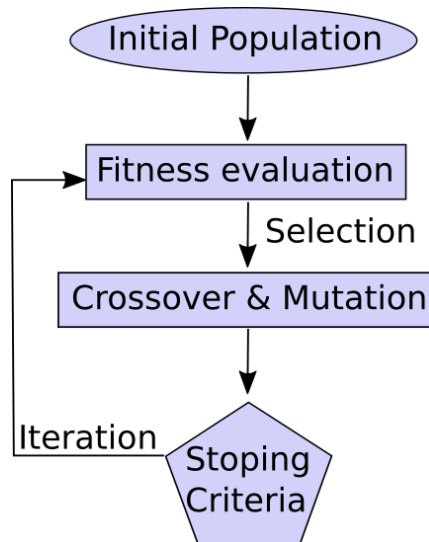


Figure 27: Illustration of the general functioning of a GA such as NSGA-II. An initial population is first sampled from the parameter space. A user-defined fitness function is used to calculate the fitness score of every member of the population. Based on their fitness scores, the members of the population "reproduce" by crossover and mutation of their parameters. Based on the characteristics of this new population and the stopping criteria the process is either repeated or stopped.

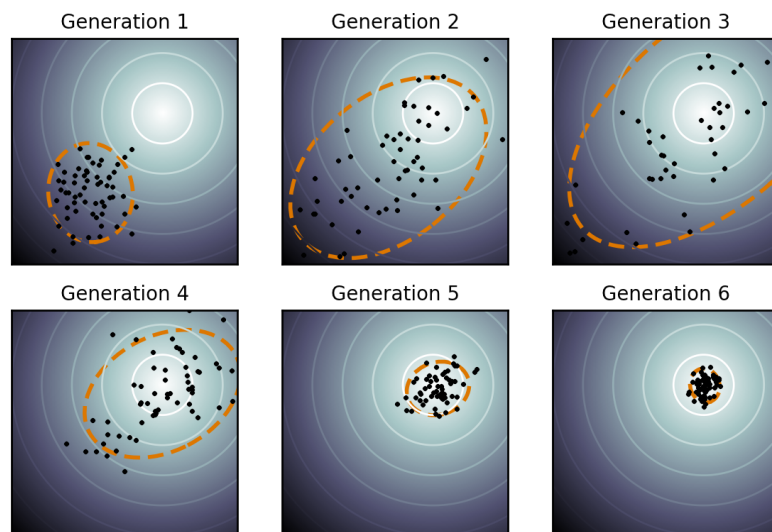


Figure 28: Illustration of the CMA-ES algorithm functioning. The fitness landscape is shown by white concentric circles, with the optimum being shaded a lighter colour. Members of the population are shown by black dots. The dashed orange oval represents the current direction of the algorithm, which is described by the covariance matrix. Reproduced with permission from [99].

	Calnexin	CLiMP63
# of compartments	1	3
# of parameters	17	60
# of species	19	58

Table 3: Increase in complexity between the Calnexin model and the the complete CLiMP63 model, including all proposed structures

there was no prior knowledge in this respect. The kinetic parameters, however, were kept the same, as there was no obvious reason for them to change between the two compartments.

The last structure that was tested involved expanding the unfolded form of CLiMP63. The difference in this case was that the unfolded form could get palmitoylated before folding. There is currently no clear experimental evidence of folding or palmitoylation occurring first or fastest, and it is most likely to depend on the protein. Including palmitoylation before folding in the model therefore accounts for both possibilities. As palmitoylation of CLiMP63 occurs close to the transmembrane domain, this may alter its interaction with the membrane and possibly affect the folding process. For this reason, two folding constants were given, one for the palmitoylated form and one for the non-palmitoylated form.

3.5.5 Integration using Julia

Although rule-based modelling facilitated generating increasingly large models of palmitoylated proteins, actually integrating the systems of ODEs also becomes computationally more expensive as model size grows. Table 3 shows the increase in model size going from Calnexin to CLiMP63.

In order to make the full CLiMP63 dimerisation model more computationally tractable it was decided to try and improve the computational efficiency of the model calibration. Refactoring the MATLAB code in order to improve performance and legibility only yielded small improvements. Because a well established ODE integrator was already being used (SUNDIALS [100]) it was decided that larger changes would be beneficial, namely using the Julia programming language [101–104]. The advantages of Julia are numerous: it is a high-level, high-performance programming language that is compiled, in the likes of C. At the same time it can directly interface with C, while there are also well tested packages for interfacing with Python. All this is free and open source published under an MIT license. Furthermore, a Julia interface to the SUNDIALS package was already available in the DifferentialEquations.jl package [105], making comparison easy.

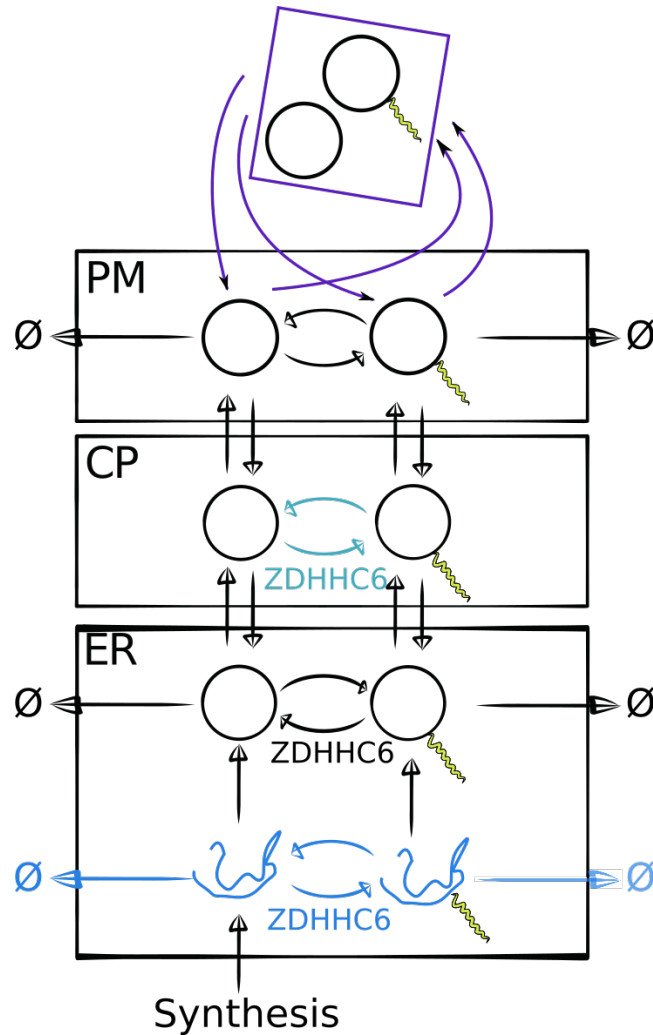


Figure 29: Illustration of the multiple model structures proposed. Each modification to the monomer CLiMP63 model is highlighted in a specific colour. A recycling exosome was added to simulate membrane recycling at the PM. No reactions could occur in this compartment. A palmitoylation/depalmitoylation catalysed by zDHHC6 and its corresponding APT was added in the CP. Palmitoylation/depalmitoylation of the unfolded CLiMP63 catalysed by zDHHC6 in the ER was added.

# of Workers	Julia 0.6 s	MATLAB R2016a s
2	1.15	44
4	0.3	14
8	0.2	6.5
12	0.17	6.5
16	0.15	<i>additional licenses needed</i>

Table 4: Speed Comparison of Julia and MATLAB. One hundred CLiMP6₃ models were integrated to steady-state as an indication of performance.

Table 4 shows the difference in computation time when using Julia or MATLAB. Results are also shown for different levels of parallelisation, where the number of workers denotes how many models are being solved simultaneously. Remarkably, a speedup of factor of 30-40 was obtained merely by switching languages while keeping the same ODE integrator backend. The reasons for this won't be discussed in depth here, as this is not the main focus of the thesis. It is most likely the ODE evaluation step which is the bottleneck in this case, as the integrator has to still evaluate the system of ODEs in either Julia or MATLAB. As one is compiled and the other interpreted, this most probably where the large increase in performance came from.

Using the easy C-interfacing capabilities of Julia, it was straightforward to use the C implementation of the CMA-ES optimisation algorithm described in Section 3.5.3.

3.6 RESULTS

After incorporating the multiple changes described in Section 3.5 and following many subsequent calibration attempts, a satisfactory model structure was finally found by taking dimerisation into account. Unfortunately, none of the model structures proposed in Section 3.5.4 or combinations thereof, were suitable. However, adding a CLiMP6₃ dimer in the ER is not only biologically plausible [83], but led to a good calibration. The final structure of the model is shown in Figure 30. Dimerisation can occur between two CLiMP6₃ proteins in any palmitoylation state, leading to a dimer with three different palmitoylation states. The rates of dimerisation, dimer dissociation, palmitoylation and depalmitoylation were assumed to be independent of palmitoylation state. Dimerisation and dissociation were modelled according to first order kinetics, while palmitoylation and depalmitoylation were modelled with tQSSA kinetics.

The transport of palmitoylated CLiMP6₃ monomer away from the ER was also removed from the model. Because the parameter sets obtained from

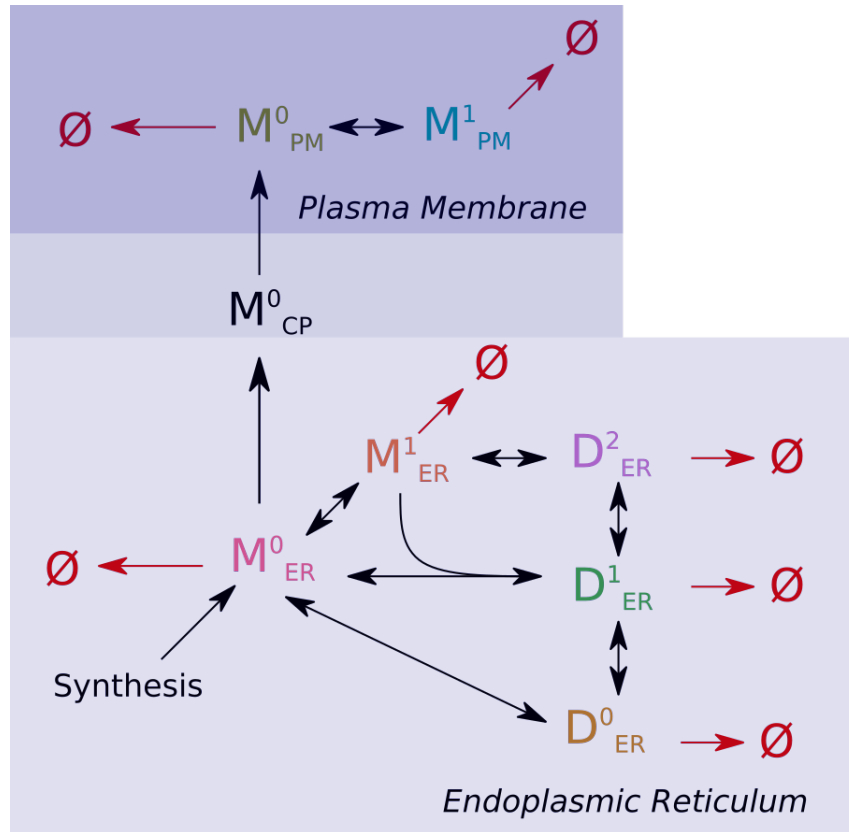


Figure 30: Final CLiMP63 model structure, containing three compartments and eight species (excluding unfolded protein). M denotes the monomeric CLiMP63 protein, the subscript its cellular localisation and the superscript its palmitoylation state, 0 for non-palmitoylated and 1 for palmitoylated. D denotes a CLiMP63 dimer, the subscript its cellular localisation and the superscript its palmitoylation state, 0 for non-palmitoylated, 1 for single palmitoylation and 2 for double palmitoylation. Compared to the initial structure, a dimer was added and transport of the palmitoylated monomer was removed. Reproduced with permission from [74].

the optimisation algorithm didn't use this transport path, it was removed in order to simplify the model representation.

As can be seen in Figure 31, the quality of the model calibration improved substantially compared to the initial efforts (Figure 24). The decay curves which have a pronounced biphasic shape, such as that of ^3H WT, were well matched by the model. The ^{35}S C100A experiment, where palmitoylation is blocked by mutating a cysteine to an alanine (Section 2.2.4), was not as well reproduced as the other experiments. However, the experimental error of this experiment was particularly large, and the model output still within two standard deviations at most. The validation set, shown in Figure 32, was also well reproduced by the model. The experimental curves with a biphasic shape were equally well matched. The last time point ^3H WT in-

	Monomer	Recycling in the PM	zDHHC6 in the CP	Unfolded species in the ER	Dimer
# of pa- rameters	41	45	45	45	48
# of ODEs	27	33	27	31	48

Table 5: Comparison of number of model parameters and ODEs for the different CLiMP63 model structures which were tested.

corporation experiment was not quite reached by the model, but this point had a large error in any case.

Not only were the pulse-chase experiments well matched by the model, but the population distribution was also consistent with the experimental data. As shown in [Figure 33](#), not only is the majority of CLiMP63 at the ER in both WT and zDHHC6 silencing conditions, but there is a significant increase in the surface population when silencing zDHHC6 compared to WT. The palmitoylated fraction of CLiMP63 was very high, also in agreement with the experimental data in [Figure 21](#).

Given the successful calibration which was enabled by the addition of a dimer, it was decided to experimentally validate its presence. [Figure 34](#) shows a Blue native blot using either Digitonin or SDS as a surfactant. Using a Blue Native gel makes it possible to conserve protein complexes such as dimers or multimers, and still separate them according to size. Using Digitonin as a surfactant instead of the more aggressive SDS limits the breakdown of these protein complexes. As shown in [Figure 34](#) it is obvious that a CLiMP63 complex is formed in the cells. However, more investigation is needed to determine the exact composition and stoichiometry of this complex. Nevertheless, the modelling approach here led to the discovery of the central role of the CLiMP63 dimer as an integral part to its regulation, opening up additional avenues of research.

3.7 DISCUSSION

Using the calibrated CLiMP63 model it is possible to get a better understanding of how the system works and make certain predictions that go beyond what is experimentally possible.

One of the first things we can use the model for is to study how the different species of CLiMP63 interconvert during the various experiments. [Figure 35](#) shows WT ^{35}S and ^3H chases.

Let us first look at the ^{35}S chase. At the beginning of the chase we can see that the main species present is the unpalmitoylated monomer. Very

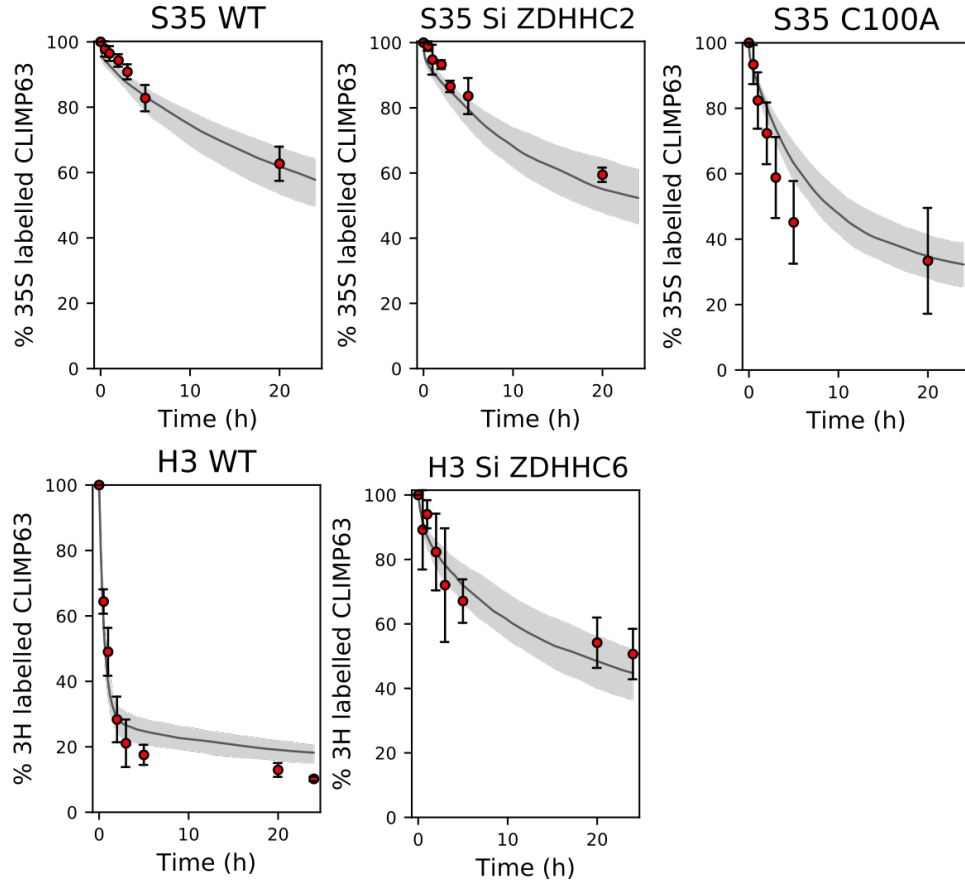


Figure 31: Experimental data set that was used for calibration of the CLiMP6₃ model. Red dots represent the experimental data points (N=5), and their error bars the standard deviation. Solid grey lines show the median of 100 models, with the shaded area representing the first and third quartile. Reproduced with permission from [74].

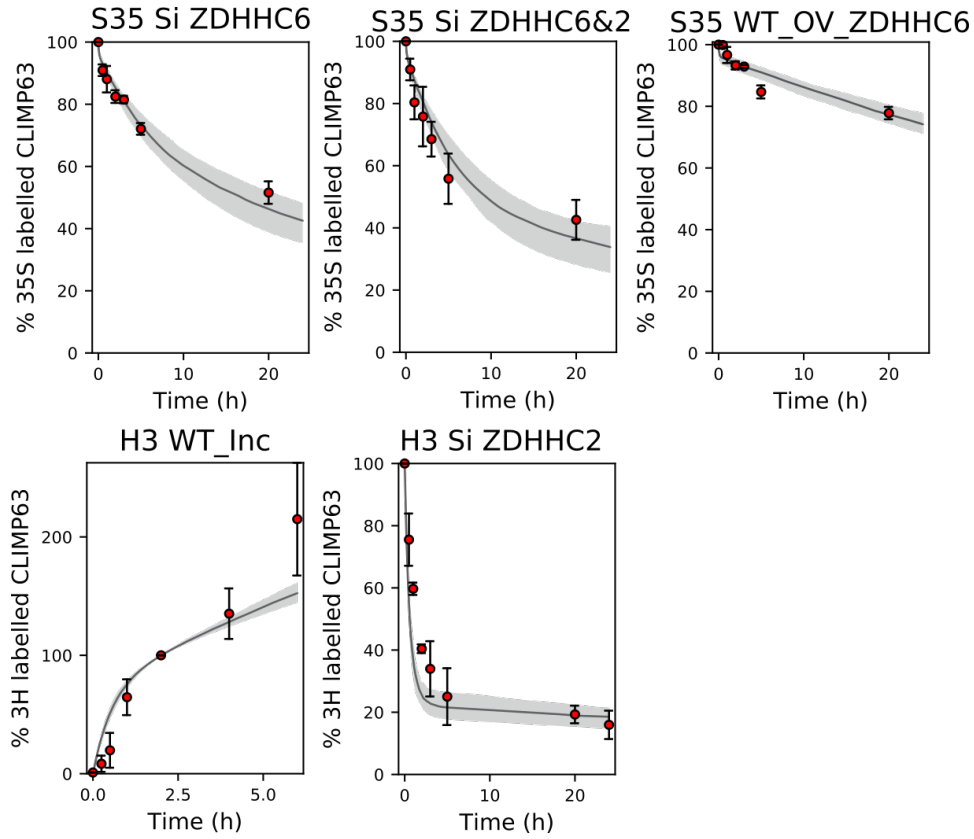


Figure 32: Validation set of the CLiMP₆₃ model. Red dots represent the experimental data points (N=5), and their error bars the standard deviation. Solid grey lines show the median of 100 models, with the shaded area representing the first and third quartile. Reproduced with permission from [74].

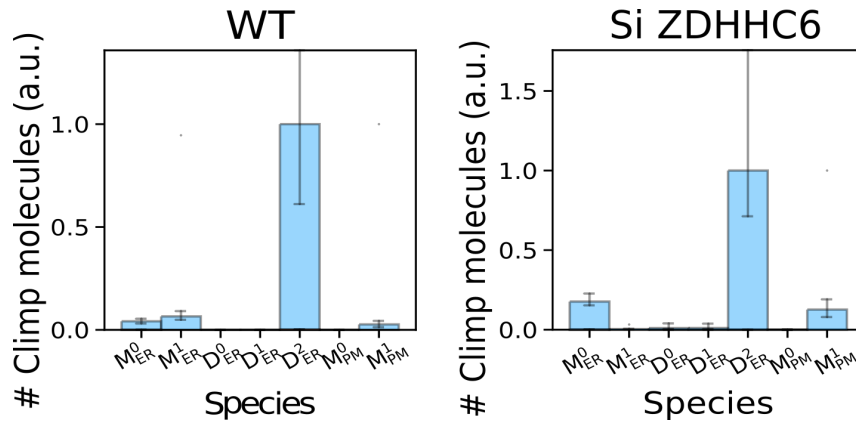


Figure 33: Population distribution of species in the model under normal (WT) conditions and when silencing zDHHC6. The mean of a 100 models is shown. Error bars represent the standard deviation. Reproduced with permission from [74].

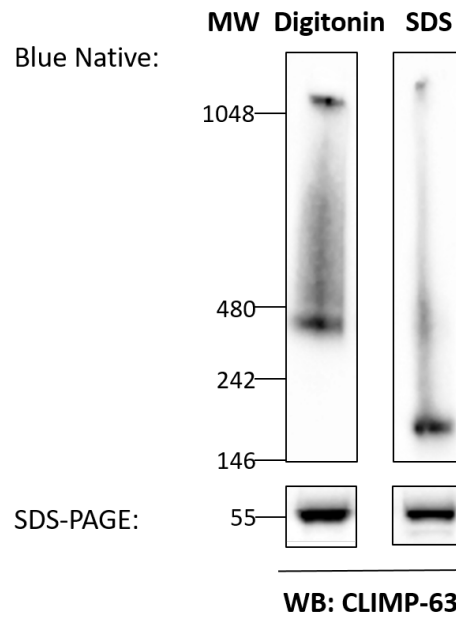


Figure 34: HeLa cells were ran on either Blue Native gel or SDS-PAGE. Two different surfactants, Digitonon and SDS, were used.

rapidly, a large fraction of this monomer is palmitoylated. The majority of the protein is then in the palmitoylated monomeric form. As the chase continues, the majority of the palmitoylated monomer dimerises to form the fully palmitoylated dimer. A small part of the remaining unpalmitoylated monomer is also transported to the plasma membrane where it is immediately palmitoylated. The deconvolution of this experiment into the individual species gives insight into how CLiMP6₃ is formed. And shows the importance of both the palmitoylated and unpalmitoylated monomeric form in the ER. One will dimerise and the other is transported to the PM where it is palmitoylated. The fully palmitoylated dimer and palmitoylated monomer at the PM are the long-lived, stable forms of CLiMP6₃. It is also these forms which carry out the functions of CLiMP6₃. The dimer most probably influences the shape of the ER, and the monomer at the surface acts as a ligand receptor [23].

Now let us focus on the ^3H chase in Figure 35. Because we are labelling with palmitate, we will only see species which have at least one palmitate attached. Similarly to the ^{35}S chase, the dominant species at the beginning is the palmitoylated monomer, while the second most abundant species is the fully palmitoylated dimer. The palmitoylated monomer then quickly decays, leaving mostly dimer and a minority of monomer at the PM. Although it may appear from the ^3H WT chase that the palmitoylated monomer is highly unstable, we must keep in mind that ^3H labelling accounts for interspecies conversion, decay and depalmitoylation. Indeed, when looking

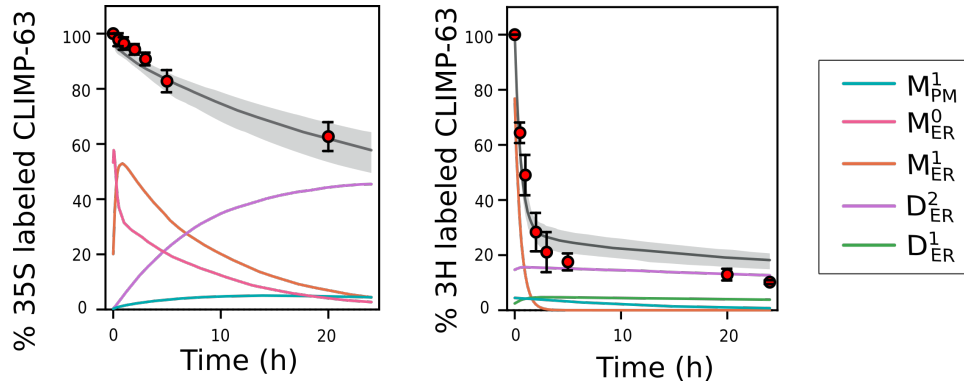


Figure 35: Deconvolution of WT ^{35}S and ^3H chases into the individual labelled species. The model was used to predict the evolution of the individual species over time. Experimental results are shown in red dots ($N=5$). The error bars represent the standard deviation. Solid grey lines represent the calibrated model output, and the shaded areas represent the first and third quartiles. Solid coloured lines show the evolution of individual species over time, their quartiles are omitted for clarity. Reproduced with permission from [74].

at the rates of palmitoylation and depalmitoylation (Figure 36), we can see that the monomer undergoes highly dynamic cycles of palmitoylation and depalmitoylation in the ER, whereas these cycles are much less active at the PM. This is why the monomer decays much faster in the ^3H labelling experiment than in the ^{35}S one. On the other hand, we can see from Figure 36 that although the rate of dimer palmitoylation is relatively small, its depalmitoylation is almost zero. The rate of dimer palmitoylation also probably appears low because there are more palmitoylated monomers than non-palmitoylated (Figure 33). Because of this, dimer formation mostly occurs between already palmitoylated monomers. However, the almost non-existent depalmitoylation of the dimer shows that dimerisation protects CLIMP63 from depalmitoylation.

The model can also be used to predict *bona fide* half lives of the individual species. Because model calibration is carried out in an unbiased way, no prior information as to the general stabilising effect of palmitoylation is given. Any differences in decay constants, and therefore half-lives, are purely a result of calibrating the model parameters with the experimental data. *In silico* half-lives are estimated by labelling a particular species at steady-state, then integrating the model until the labelled fraction drops to half its initial value.

As we can see from Figure 37, the fully palmitoylated dimer is by far the most stable species. Another observation is that palmitoylation does not significantly contribute to stability in the ER. It is the combination of dimerisation and complete palmitoylation which lead to a significant in-

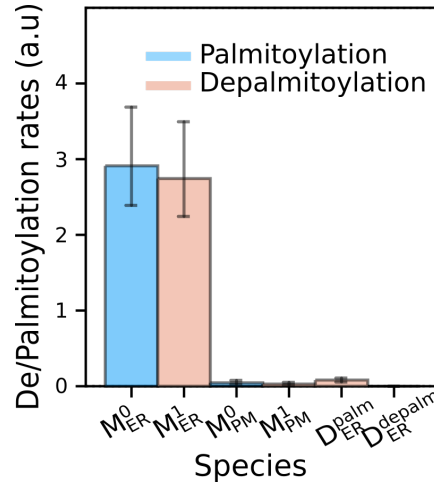


Figure 36: Palmitoylation (blue) and Depalmitoylation (orange) rates at steady-state for the different species in the model. The mean of 100 models is shown. Error bars show first and third quartiles. The palmitoylation rates for the various dimer species are summed. The same summing is done for the depalmitoylation rates of all dimer species. Reproduced with permission from [74].

crease in stability and therefore half-life. The monomeric subpopulation at the PM is more stable than its counterpart in the ER. The effect of palmitoylation of the monomer at the PM also leads to a drastic increase in stability, causing the half live to increase more than two fold.

After having predicted that the fully palmitoyled CLiMP63 dimer had a half life of approximately eighty hours, we tried to validate this experimentally. This was done by expressing a SNAP-tag on mutated CLiMP63 protein. A range of fluorescent reporters will spontaneously form a covalent bond with this tag, enabling *in vivo* labelling of mature proteins. In this case, the fluorescent reporter TMR-star was used. Using this method, any CLiMP63 protein in the cell will be tagged, not just newly synthesized or newly palmitoylated proteins as is the case with radiolabelling. Because the fully palmitoylated dimer was predicted to be the dominant species, labelling with this method will mostly target this species.

The results shown in Figure 38 agree with the predicted half-life of the fully palmitoylated dimer. Over a time period of twenty four hours no decay was observed. Because of experimental limitations, a longer monitoring period was not possible. Assuming a half-life of eighty hours and an exponential decay arising from first order kinetics, the amount of dimer present after twenty four hours should be 87.5% of the original amount. Looking at the experimental data, this is within two standard deviations of the measured amount. Given large uncertainty in the predicted half life shown in Figure 37, the experimental data confirms the validity of this prediction.

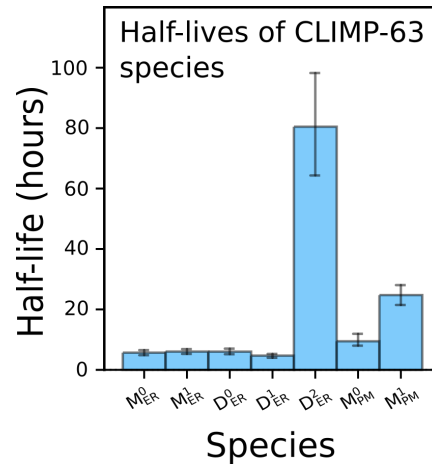


Figure 37: Half-lives of the various CLiMP₆₃ species predicted with the model. The mean of a 100 models is shown. Error bars represent the first and third quartiles. Reproduced with permission from [74].

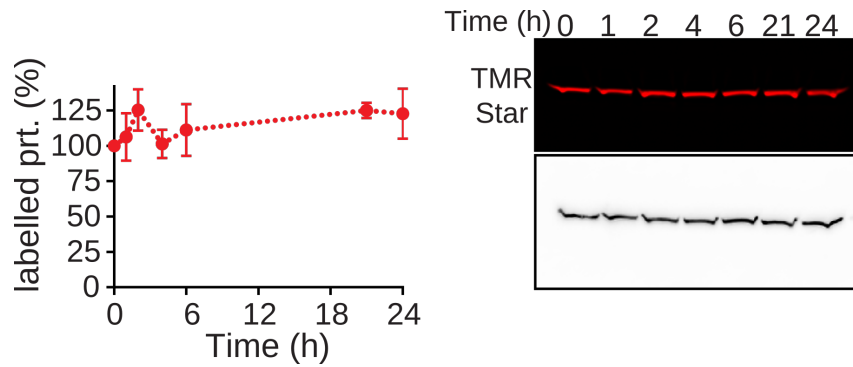


Figure 38: Quantification of CLiMP₆₃ decay labelled with TMR-star (left) (N=3). Fluorescent and Western blot of CLiMP₆₃ labelled with TMR-star (right). The error bars represent the standard deviation. Reproduced with permission from Sandoz *et al.* [74].

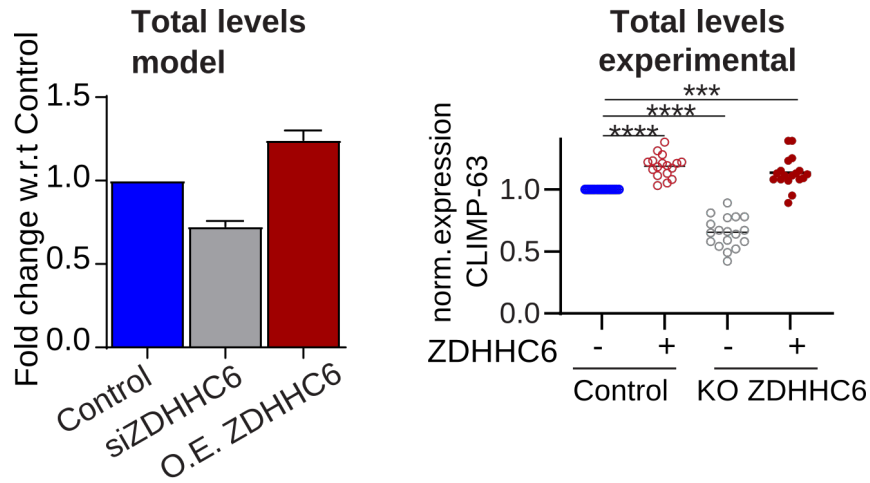


Figure 39: The effect of zDHHC6 silencing. Knockout and overexpression on total CLiMP63 levels is shown. Model predictions are shown on the left plot and experimental validation (N=8, ***p < 0.01, ****p < 0.001) on the right. Error bars represent standard deviation. Reproduced with permission from [74].

Using the model, we made predictions of the effect of zDHHC6 silencing and overexpression on the amount of CLiMP63, shown in Figure 39. Although it is not possible to experimentally measure the amount of each individual species, we can still measure how the total amount changes in relation to zDHHC6. As expected, an increase in zDHHC6 levels increase palmitoylation, therefore increasing stability and the total amount of CLiMP63. Because the radiolabelling experiments contained experiments where zDHHC6 was silenced or knocked out, the silencing and overexpression factors were already estimated during model calibration. Figure 39 clearly shows that the model and experimental measurements are in very good agreement.

Since palmitoylation increases the amount of CLiMP63 dimers and therefore the morphology of the ER, we wanted to simulate what happens when depalmitoylation is reduced. Figure 40 shows a slower palmitate turnover (to be expected), a slightly faster dimer formation, and a slower protein decay. This supports the conclusion that palmitoylation is key to forming mature CLiMP63, and that this can be modulated by either increasing the activity of the palmitoylating enzyme zDHHC6, or reducing depalmitoylation of the corresponding APT. Because the APT of CLiMP63 hasn't yet been identified, these model predictions couldn't be directly experimentally validated. Instead, by inserting a second cysteine next to the first, another palmitoylation site was created, so that instead of one palmitate to be removed by the APT there were now two. In this way, palmitoylation was reduced and the model predictions could be validated, as shown in Figure 40.

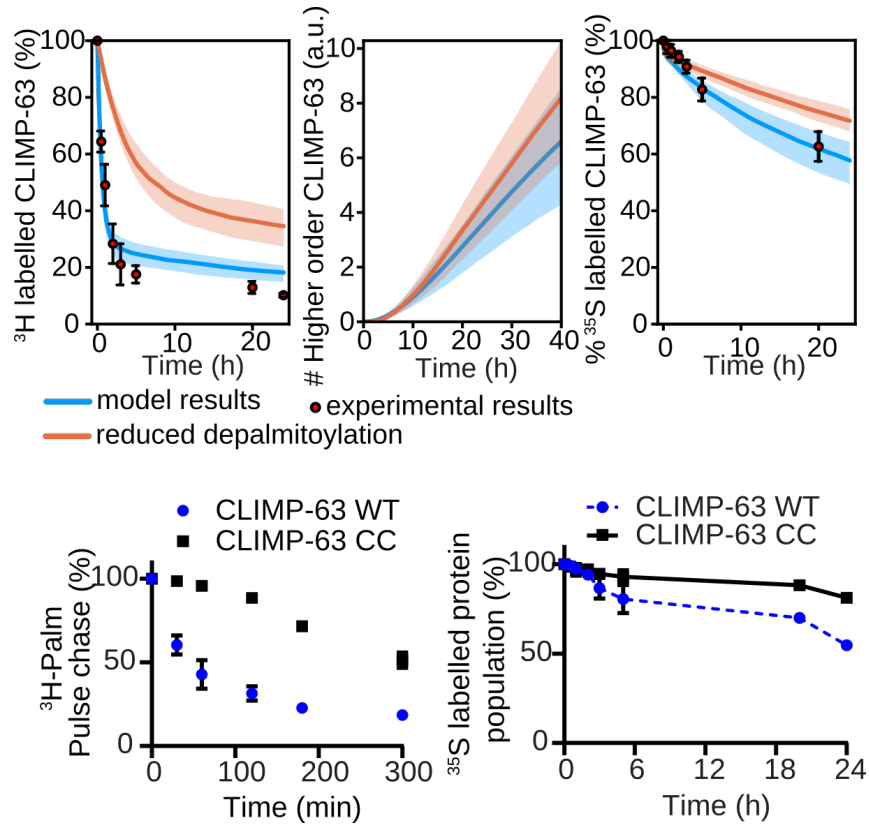


Figure 40: Computational prediction of reduced depalmitoylation. The top three figures show predictions of the effect of reduced depalmitoylation on ^3H labelling decay, dimer formation and ^{35}S labelling decay. Experimental data points ($N=3$) are shown by red dots and the error bars represent standard deviation. The bottom two graphs show the experimental data corresponding to the ^3H and ^{35}S decay ($N=3$). Error bars represent standard deviation. Reduced depalmitoylation was obtained experimentally by introducing a second cysteine adjacent to the endogenous one. Reproduced with permission from Sandoz *et al.* [74].

3.8 CONCLUSION

In this chapter we showed the entire lifecycle of a model, from its development, to its calibration and finally the predictions and additional knowledge it can bring. We highlighted the importance of expert knowledge from the field, and the richness that combining different kinds of experiments can bring. We were able to propose a model which consistently explains all available experimental data and led to the discovery of the importance of the dimeric form of CLiMP63.

Initially using a similar approach to that used for the Calnexin model, we constructed a model of CLiMP63 synthesis, trafficking, palmitoylation, depalmitoylation, dimerisation and decay. We used a combination of first order kinetics and tQSSA for describing enzymatic reactions. Given the background knowledge and available experimental data on CLiMP63 we were able to come up with a model spread across three compartments. Within two of these, enzymatic reactions were present that could palmitoylate and depalmitoylate the single site of CLiMP63. However, calibrating this model was unsuccessful and we would need to rethink the entire pipeline of model construction and calibration.

In order to obtain a model that could accurately reproduce the experimental data we proposed and tested multiple different model structures. With the aim of easily iterating through these different model structures, we used rule-based modelling to facilitate model generation. Because of the increased computational load, we moved to a new programming language called Julia, a high-level, high-performance language targeted at numerical computing. We also changed optimisation algorithms from NSGA-II to CMA-ES and reformulated the error functions to take into account the pulse-chase experiments in an unbiased way. All these efforts made it possible to finally propose a model which included dimerisation of CLiMP63, and which could accurately reproduce the experimental data.

Using this model, we showed that dimerisation is the principal mechanism of retention in the ER. The combination of dimerisation and palmitoylation leads to a particularly stable species of CLiMP63, effectively anchoring it in the ER and regulating its morphology. This combination also prevents its depalmitoylation, possibly by rendering the palmitoylation sites inaccessible to any depalmitoylating enzymes. We showed that membrane targeting is regulated through zDHHC6. Although levels of CLiMP63 are relatively constant when silencing this enzyme, reduced palmitoylation leaves more unpalmitoylated monomer available to be transported to the PM. While palmitoylation by itself has little effect on stability of CLiMP63 at the ER, at the PM it increase half-life more than two fold.

4 CMG₂

In this chapter we will describe the construction and calibration of a kinetic model describing the synthesis, transport, palmitoylation and decay of Capillary Morphogenesis Gene 2 (CMG₂). The modelling approach and experimental methods are an extension of those already mentioned in [Chapter 3](#) and [Chapter 2](#). However, in the case of CMG₂ there is significantly more data available, as well as an increased combinatorial complexity due to the presence of three palmitoylation sites on CMG₂.

We will go through the experimental data pertaining to CMG₂, and show how the palmitoylation enzymes were identified and their localisation deduced. We will also see that the experimental data already indicate that localisation and palmitoylation of CMG₂ are tightly coupled.

Once we have build and calibrated a kinetic model of CMG₂, we will use it to answer several questions concerning its behaviour and how it is modulated by palmitoylation. Firstly, the known roles of CMG₂ are exclusively those of a receptor located at the PM, where CMG₂ is a collagen VI receptor and can also be hijacked by the anthrax toxin. Given its role at the PM, we will evaluate how palmitoylation affects the biosynthetic flux of CMG₂, ultimately influencing the amount of mature, functional CMG₂ at the PM. We will also quantify the influence of different enzymes on this biosynthetic flux, and evaluate any interactions between them. Another question concerns the effect of palmitoylation on subcellular localisation. Specifically, how palmitoylation may modulate the distribution of CMG₂ throughout the different subcellular compartments. We will also estimate the half-lives of the various CMG₂ species, with the aim of evaluating how palmitoylation and localisation affect the stability of CMG₂. The half-life of the CMG₂ subpopulation at the PM is of crucial interest, as this is where its function is carried out.

The experiments on CMG₂ were performed by various members of the Van der Goot lab, namely: Sanja Blaskovic, Laurence Abrami, Fransico Mesquita and Oksana Sergeeva.

4.1 CMG₂ BACKGROUND

Capillary Morphogenesis Gene 2 (CMG₂) is a transmembrane protein that functions as a collagen VI receptor and regulator [106], and is involved in the regulation of the Extracellular Matrix (ECM). Its absence causes the accumulation of collagen VI, which is one of the symptoms of the disease Hyaline Fibromatosis Syndrome (HFS) [106, 107].

There are multiple different mutations in CMG2 that can cause HFS [108–110]. In some cases, single point mutations are sufficient. HFS causes accumulation of extracellular matrix [106], which manifests as nodules. The current treatment of HFS are limited to alleviating the pain and symptoms while being unable to address the root cause. Rather than a precise set of symptoms, HFS manifests itself across a spectrum, where many children with a more severe form die in early childhood, while some of those with a milder form survive until adulthood [111].

Capillary Morphogenesis Gene 2 (CMG2) is also involved in anthrax toxin endocytosis [112, 113], hence why it is sometimes called Anthrax toxin receptor 2 (ANTXR2). Palmitoylation, as well as phosphorylation and ubiquitylation, have all been shown to be required for anthrax toxin endocytosis [114]. Furthermore, it has also been shown that Palmitoylation influences the surface residence time of the anthrax toxin [115].

Similarly to Calnexin and many other signalling proteins, it has a long disordered cytosolic tail [116]. An illustration of this protein's structure is shown in Figure 41. Multiple PTMs target this cytosolic tail. These include phosphorylation, ubiquitylation and palmitoylation [115]. CMG2 has a total of three palmitoylation sites. Two of these are adjacent to the transmembrane domain, as depicted in Figure 41, while the third palmitoylation site is located along the disordered cytosolic tail [117]. As we will show in Section 4.2, CMG2 has two palmitoylating enzymes: zDHHC7 which is located in the ER, and zDHHC3 which is found in the Golgi apparatus.

4.2 EXPERIMENTAL DATA

The experimental methods used to study CMG2 are similar to those used for CLiMP63 (Section 3.2). Namely, ^{35}S and ^3H radiolabelling, siRNA for lowering enzyme levels, mutants for disabling palmitoylation sites, TCE western blots for measuring protein levels and surface biotinylation for measuring surface protein levels. More details on the experimental protocols can be found in the publication by Blaskovic [117].

Like CLiMP63, we will show that CMG2 has two palmitoylating enzymes in two different compartments. However, it has a total of three palmitoylation sites [117], meaning that there are eight different CMG2 species once the occupation of these palmitoylation sites are taken into account. Furthermore, we will also show experiments from two different cell lines, whereas in previous models all data was obtained from a single cell line. The RPE1 cell line was used because it expresses CMG2 endogenously, whereas the HeLa cell line does not and was therefore transfected with CMG2 mutants.

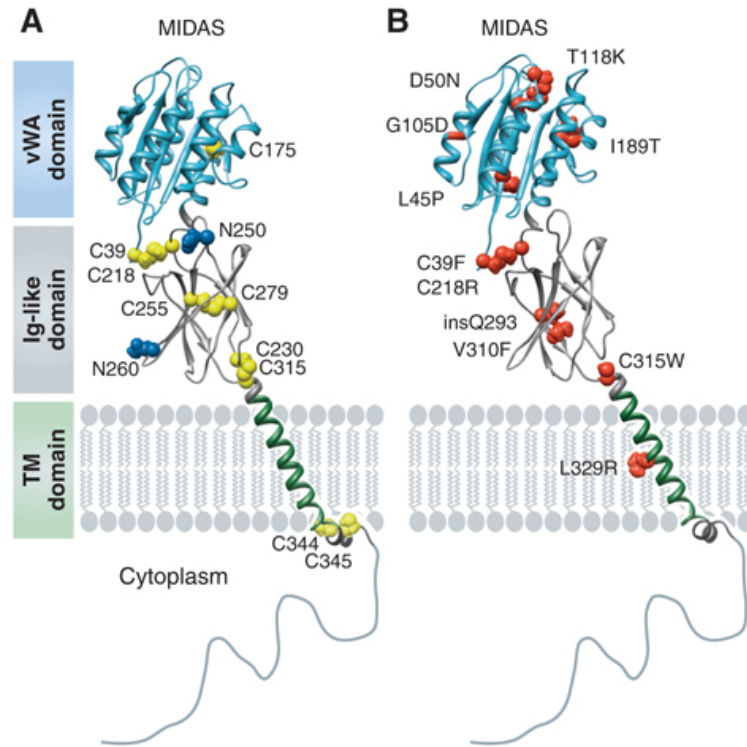


Figure 41: Representation of the structure of CMG2, reproduced with permission from Deuquet *et al.* [118]. On the left panel, cysteines are shown in yellow. The two palmitoylated cysteines C344 and C345 can be seen on the cytosolic side adjacent to the transmembrane domain. On the right panel, mutations linked to HFS are shown in red.

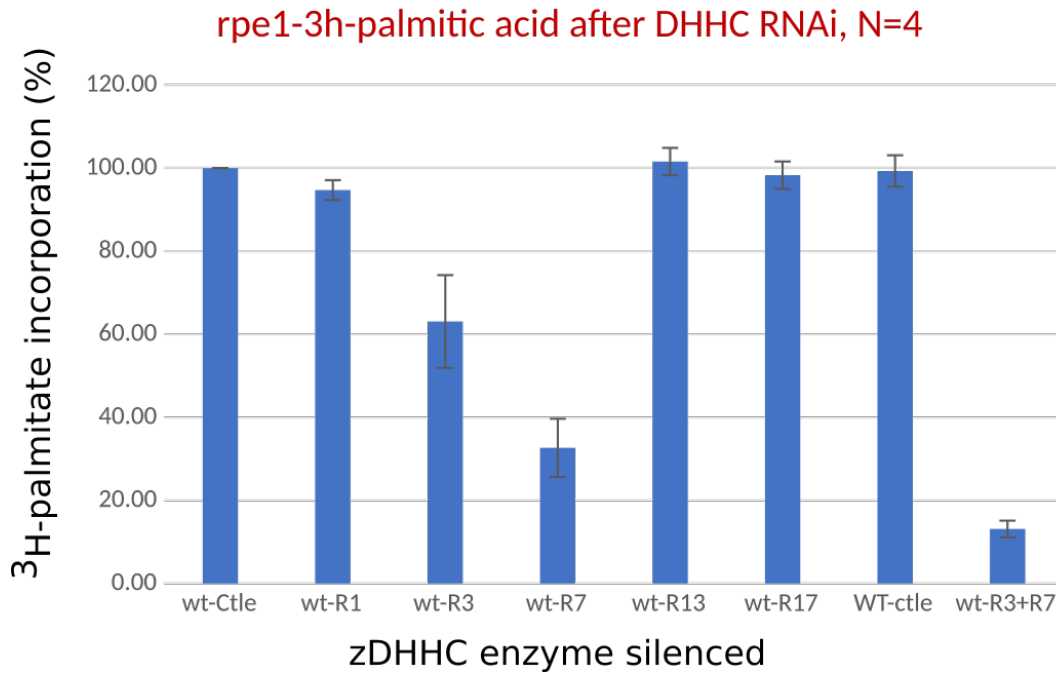


Figure 42: PAT screening experiment for CMG2. ³H palmitate incorporation was performed in WT conditions, and when silencing multiple different PATs in order to determine which ones palmitate CMG2. The notation 'R1' indicates that zDHHC1 was silenced. Error bars show standard deviation calculated from experimental repeats (N=8).

4.2.1 Identifying the palmitoylating enzymes

One of the first questions that need to be answered before building the model, was which PATs palmitoylate CMG2 and where are they located. [Figure 42](#) shows a screening experiment where ³H palmitate incorporation was measured when silencing various PATs. As we can see from the figure, the only two PATs that seem to have an effect on CMG2 are zDHHC3 and zDHHC7.

Although there are several reports of zDHHC3 and ZDHHC7 being localised at both the ER and the Golgi apparatus [[119–123](#)], these tend to contradict each other, either saying they are localised in both compartments or only one. Compounded with the fact that these observations are based on different cell lines, new experiments were performed in RPE1 cells to clarify the localisation of these two PATs. Two drugs were used for this purpose. The drug Brefeldin removes the Golgi apparatus by fusing it to the ER, whereas Cycloheximide prevents protein synthesis in the cell. In short, these drugs act either at the ER or at the Golgi, respectively. As shown in [Figure 43](#), once zDHHC3 is silenced and only zDHHC7 remains, palmitate incorporation is only affected by Cycloheximide and not Brefeldin.

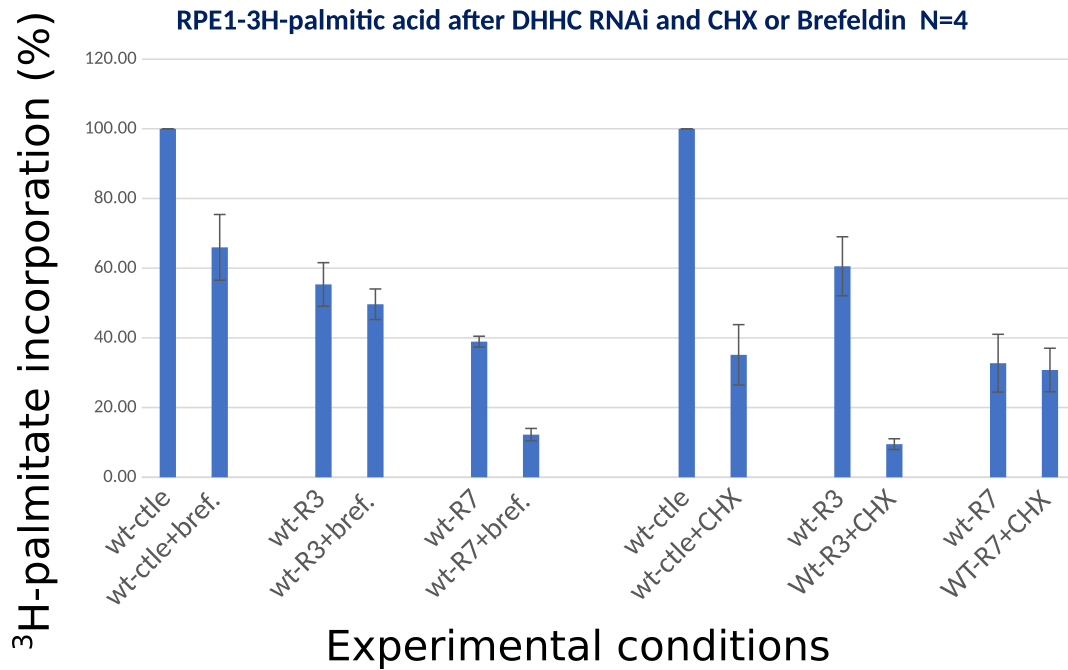


Figure 43: ³H incorporation experiment on CMG₂. Different experimental conditions were used: silencing zDHHC7 (R7) or zDHHC3 (R3), or using the drugs Brefeldin (bref) or Cycloheximide (CHX). Brefeldin fuses the Golgi apparatus to the ER and Cycloheximide prevents protein synthesis. Error bars represent the standard deviation calculated from experimental repeats (N=4).

This means that zDHHC7 is active only in the ER and not the Golgi. Conversely, once zDHHC7 is silenced and only zDHHC3 remains, palmitate is affected by Brefeldin but not Cycloheximide. Showing that zDHHC3 palmitoylates CMG₂ in the Golgi apparatus.

Therefore, zDHHC7 and zDHHC3 were identified as enzymes palmitoylating CMG₂. They are localised at the ER and the Golgi apparatus, respectively.

4.2.2 Characterising the third palmitoylation site

In order to get a better idea of how the individual palmitoylation sites contribute to CMG₂ stability, multiple mutants were prepared. As shown in Figure 44, the single mutation which reduces the amount of CMG₂ the most is the first site. If we look at the double mutants, it seems the second site has the second biggest contribution to stability and that the third site has no effect. Finally, by comparing the AAA mutant, where all palmitoylation sites are disabled, to the AAC mutant, it seems that the third palmitoylation has no discernible effect on protein levels. We can also see this when comparing the ACC and ACA mutants, which have similar protein levels.

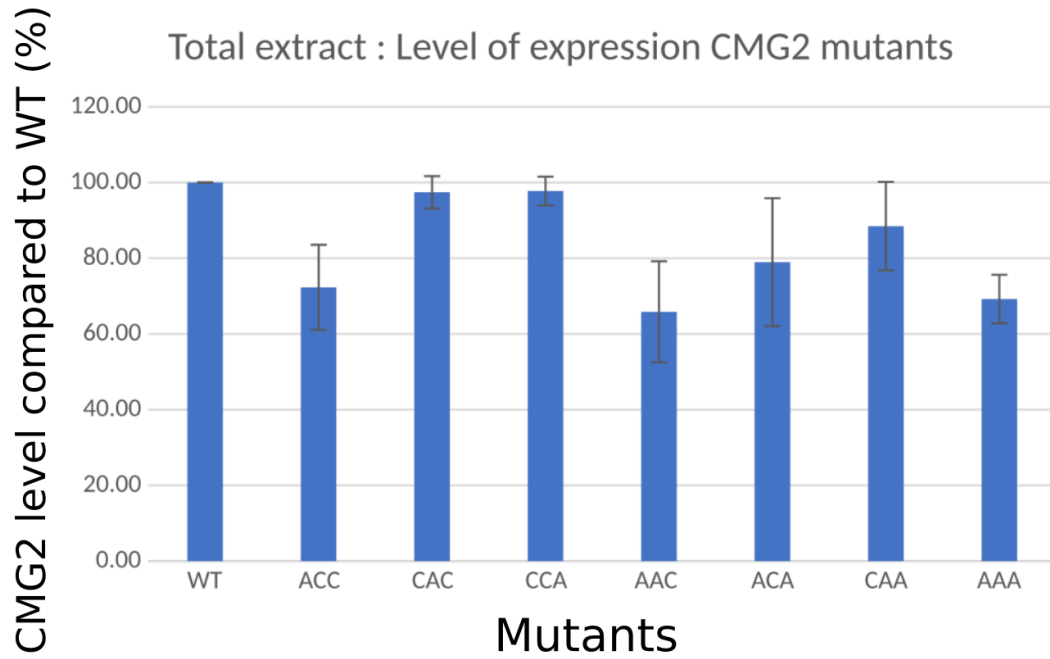


Figure 44: Experiment showing the influence of different mutants on CMG2 levels. Results are normalised to WT conditions. CMG2 has three palmitoylation sites. A 'C' indicates that the Cysteine is still present and the palmitoylation site active. An 'A' indicates the site was mutated to alanine so that palmitoylation is no longer possible. Error bars represent standard deviation calculated from experimental repeats (N=9).

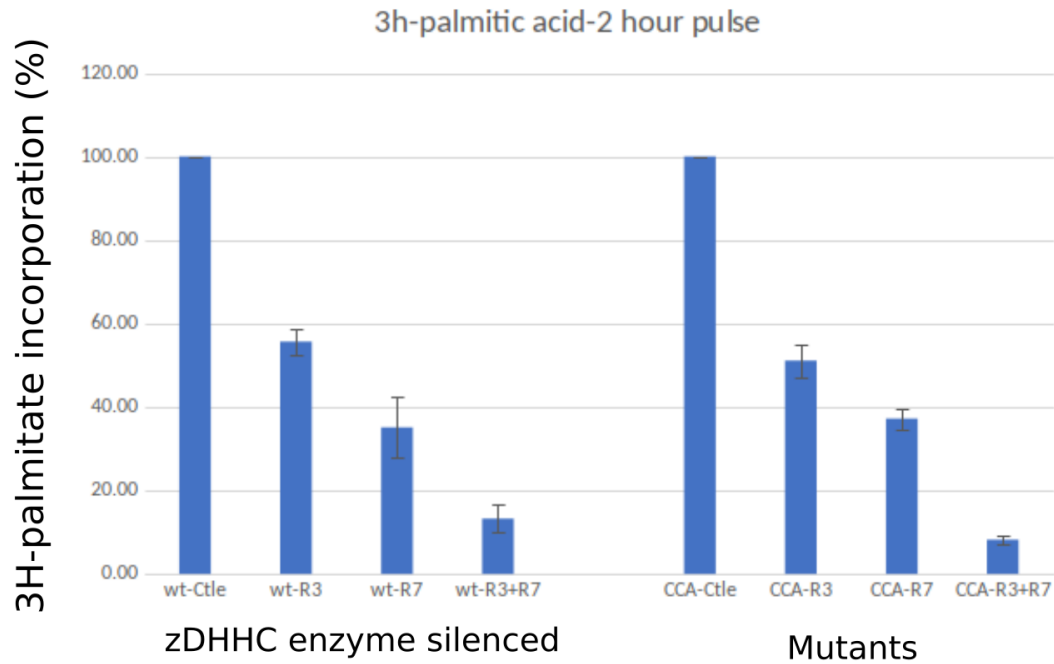


Figure 45: Various experiments showing how ^3H palmitate incorporation changes in response to silencing zDHHC enzymes and mutating the third palmitoylation site of CMG2. Error bars show standard deviation calculated from experimental repeats (N=4).

To better characterise this third palmitoylation site which appeared inactive, experiments were performed silencing both PATs and using the CCA mutant, whose third palmitoylation site is mutated to Alanine and therefore cannot be palmitoylated. As shown in Figure 45, it truly seems that this third site is almost completely inactive, as palmitate incorporation does not change when it is mutated, irrespective of which PAT enzyme is present.

4.2.3 Subcellular distribution

Total cell extract and surface biotinylation were used to study how the subpopulations of CMG2 behave differently. Namely, the two subpopulations studied were the total amount of CMG2 and the subpopulation which is located at the cell surface. The total cell extract quantifies the total amount of CMG2, whereas surface biotinylation quantifies the amount of protein at the PM. These two experimental methods were used in conjunction with PAT and APT silencing to determine how palmitoylation modified the total amount of CMG2 and its distribution in the cell. As shown in Figure 46, the total amount of CMG2 isn't significantly affected when either zDHHC7 or zDHHC3 are silenced. It is, however, slightly reduced when both are silenced at the same time. On the other hand, the surface population is

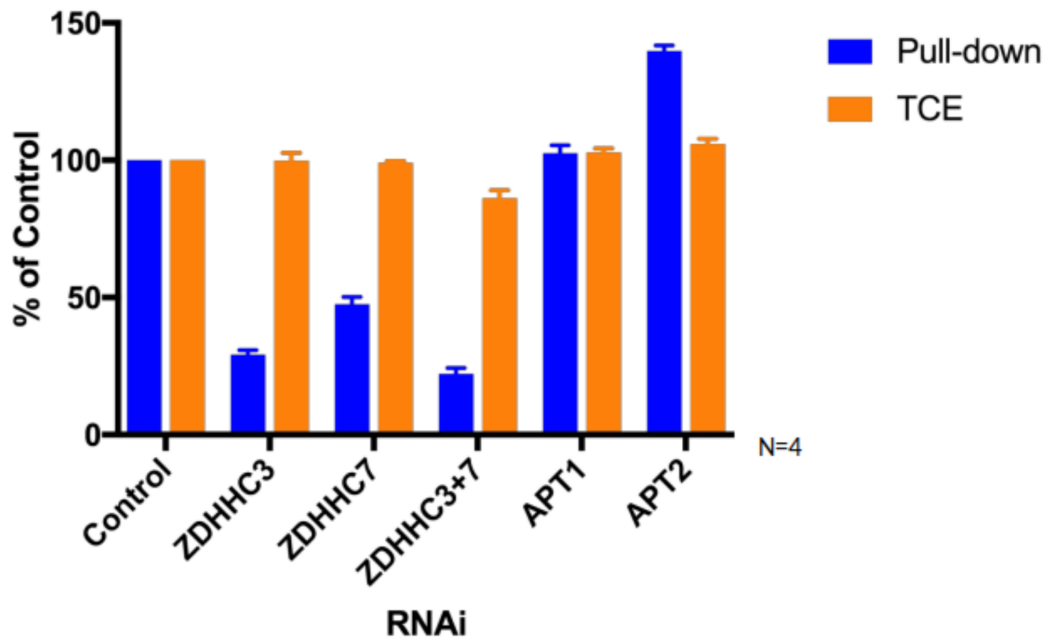


Figure 46: Comparison of TCE and surface biotinylation experiments when silencing various PATs or APTs. Total cell extract values are shown in blue bars, and the values of the surface biotinylation experiment are shown in orange. The top of the bars show the mean, and the error bars show the standard deviation calculated from experimental repeats (N=4).

very sensitive to silencing of either of these enzymes. This shows that although palmitoylation doesn't appear to affect the total amount of CMG₂, its primary function is targeting CMG₂ to the surface of the cell.

Two APTs that were suspected to target CMG₂ were also silenced, with only APT2 having a measurable effect. This may help to identify the APTs of CMG₂ in the future. More importantly it underlines how important palmitoylation is for surface targeting. Showing that palmitoylation generally promotes localisation of CMG₂ to the cell surface.

4.2.4 Radiolabelling

Figure 47 shows multiple ³⁵S pulse-chase experiments. These results suggest that zDDHC3 is downstream from zDHHC7. This is because silencing zDHHC7 affects the beginning of the chase, while silencing zDDHC3 does not. Because this is ³⁵S labelling, the signal we are seeing is that of newly synthesized protein as it moves through the biosynthetic pathway. Given that silencing zDDHC3 only affects the latter part of the chase, this must mean it located further down the biosynthetic pathway than zDHHC7. This also supports the localisation of both these enzymes discussed in [Section 4.2.1](#).

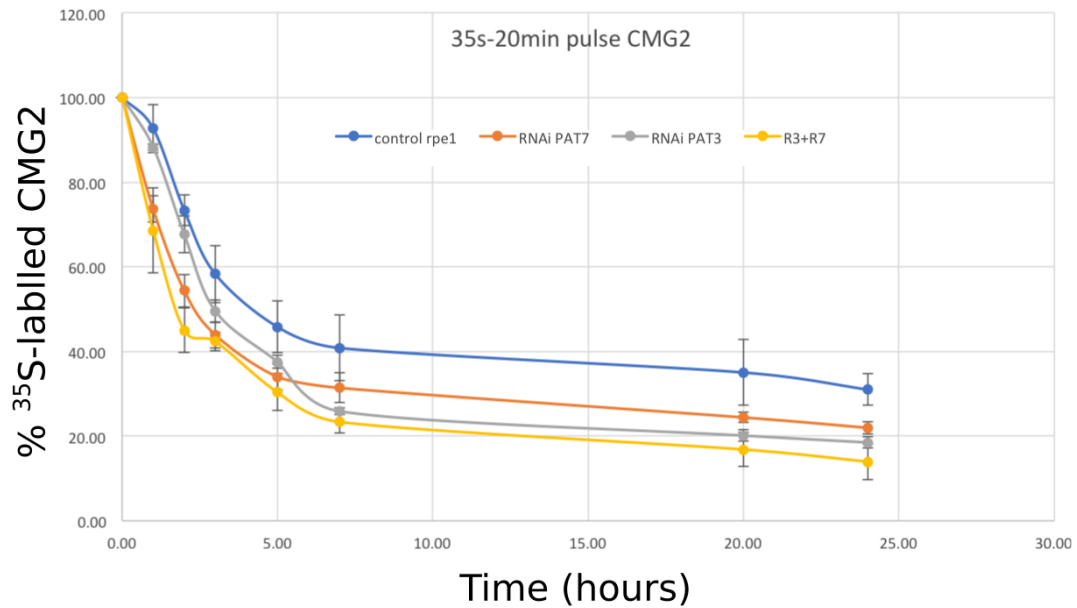


Figure 47: Multiple CMG2 ^{35}S pulse-chase experiments showing the effect of silencing zDHHC7 and/or zDHHC3. Error bars represent standard deviation calculated from experimental repeats (N=3).

The ^3H pulse-chase experiment shown in Figure 48 also supports the proposition of the third cysteine being less active than the other two. This experiment shows that relatively little depalmitoylation occurs on the third cysteine, while the second one is more active.

4.2.5 Comparing RPE1 and HeLa cell lines

Because experiments had been done in both RPE1 and HeLa cell lines, it was important to verify that CMG2 in both these cell lines was behaving similarly. Figure 49 shows a ^3H palmitate pulse-chase experiment where endogenous CMG2 in RPE1 behaves identically to transfected CMG2 in HeLa. This experiment also shows that CMG2 responds identically to the depalmitoylating enzyme APT2 in both these cell lines.

A subset of the TCE and surface biotinylation experiments were repeated in both cell lines to verify that CMG2 was behaving similarly. From the results shown in Figure 50 it is clear that the endogenous and transfected CMG2 is behaving identically.

Based on these experiments, there was no reason to suspect that CMG2 was behaving differently between the cell lines. For this reason, one model was subsequently calibrated for both cell lines.

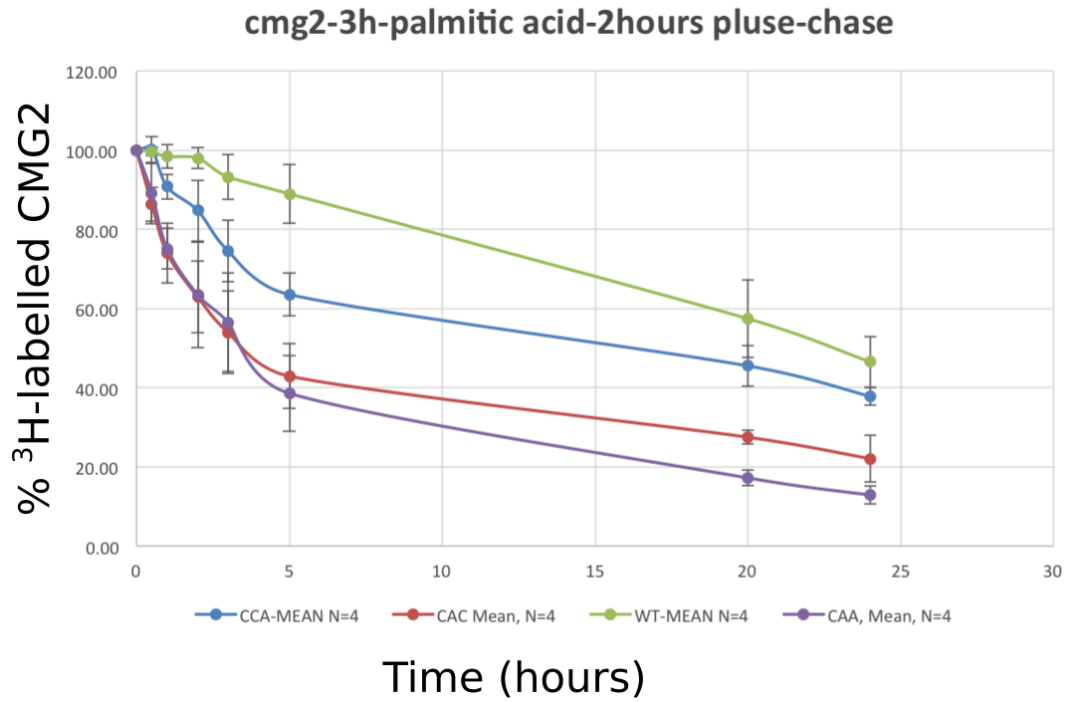


Figure 48: Multiple CMG2 ^3H pulse-chase experiments showing the effect of mutating the cysteines of different palmitoylation sites. Error bars represent standard deviation calculated from experimental repeats (N=4).

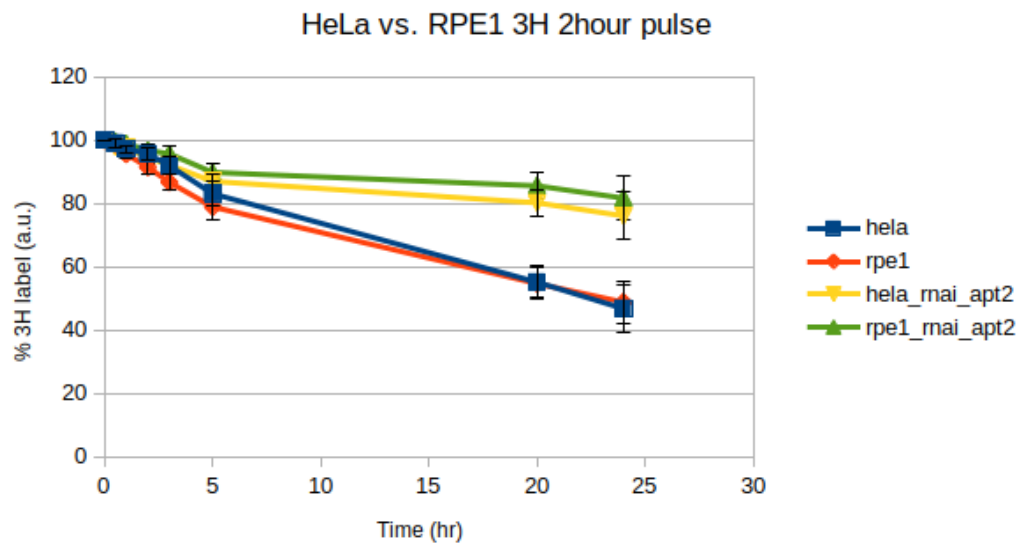


Figure 49: ^3H pulse-chase experiment in WT conditions and silencing APT2. Experiments are shown in both RPE1 and HeLa cell lines. Error bars represent standard deviation calculated from experimental repeats (N=4)

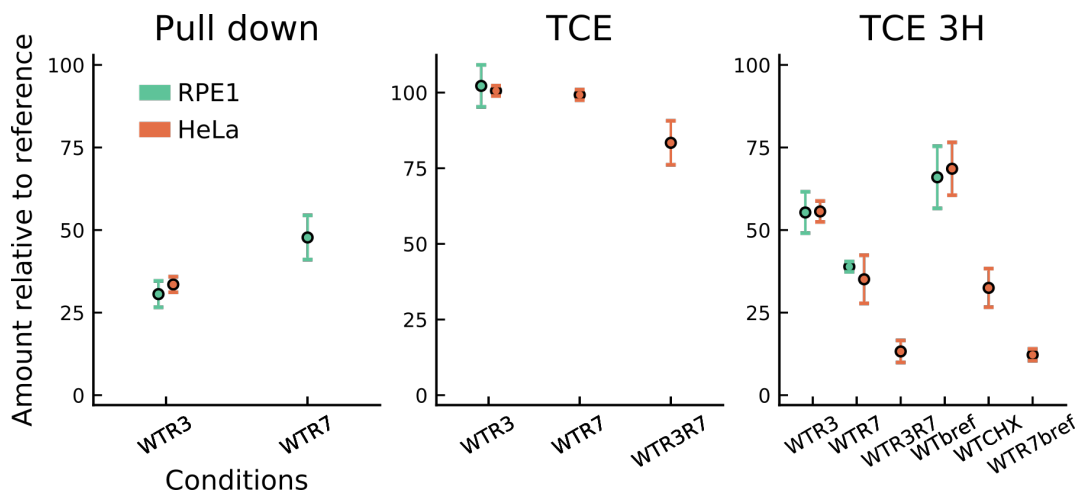


Figure 50: A subset of the TCE and surface biotinylation experiments showing the overlap of experiments performed in both RPE1 and HeLa cell lines. Error bars represent standard deviation calculated from experimental repeats (N=4).

4.2.6 Palmitoylation and anthrax endocytosis

Endocytosis of the anthrax toxin is a multistep process, involving CMG2 and multiple other proteins. The different steps necessary for endocytosis have been well described in the literature [124], an overview of these is shown in Figure 51.

A ^3H palmitate pulse-chase experiment was performed in the presence of Protective Antigen (PA), which is part of the anthrax toxin, to determine if there was any link to palmitoylation. As shown in Figure 52, when in presence of PA, the depalmitoylation of CMG2 is much faster than in control conditions. Not only does this demonstrate that the anthrax toxin somehow interacts with palmitoylation, but also suggests that an APT must be present at the PM in order to depalmitoylate CMG2.

Although this suggests that depalmitoylation is important for CMG2-mediated anthrax endocytosis, it does not clarify why this is.

4.3 MODEL STRUCTURE

Using knowledge from the literature (Section 4.1) and the experimental data (Section 4.2), an initial model structure could be drafted. The part of the model that describes CMG2 in the ER is shown in Figure 53. Given that CMG2 has three palmitoylation sites, there are at least eight different species, creating an intricate network of interconversion. Because single point mutations in CMG2 can cause HFS, it is most likely a poor folding protein. For this reason, the model also allows palmitoylation of the unfolded species.

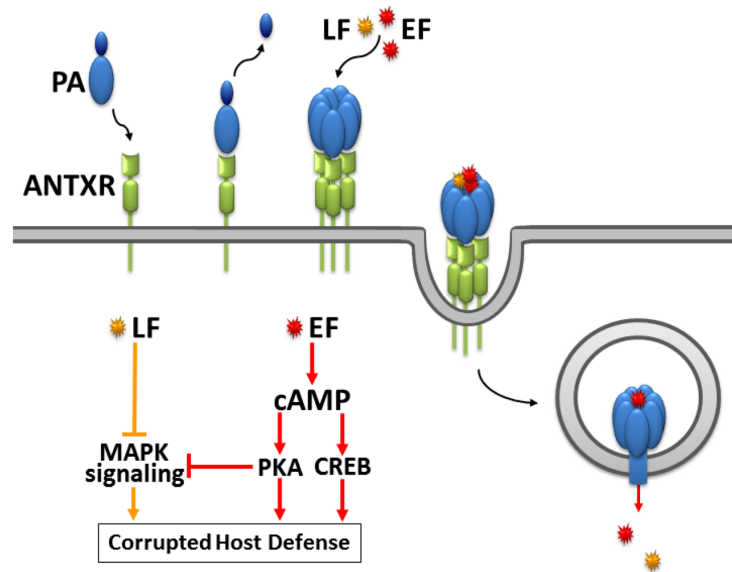


Figure 51: Illustration of the entry mechanism of the anthrax toxin, reproduced with permission from Göttele *et al.* [124]. The Protective Antigen (PA) first binds to an anthrax receptor such as CMG2. Several of these complexes then form a heptamer. Both lethal factor (LF) and edema factor (EF) can then bind to this heptamer. Once in the cell, LF and EF are released and disrupt cell signalling pathways.

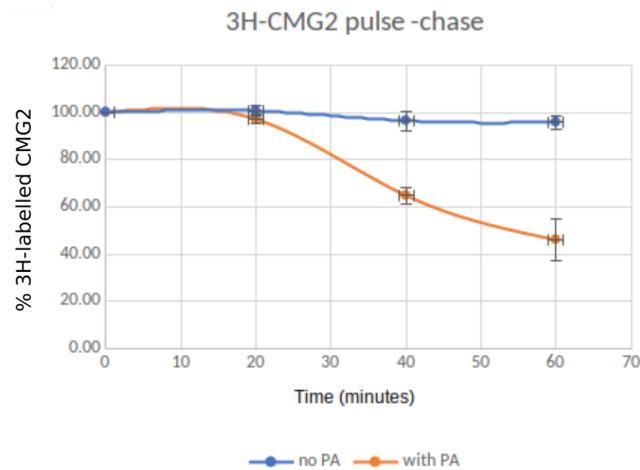


Figure 52: ^3H palmitate pulse-chase experiment following CMG2 in WT conditions and in the presence of the Protective Antigen (PA) that is part of the anthrax toxin. Error bars show standard deviation calculated from experimental repeats (N=3).

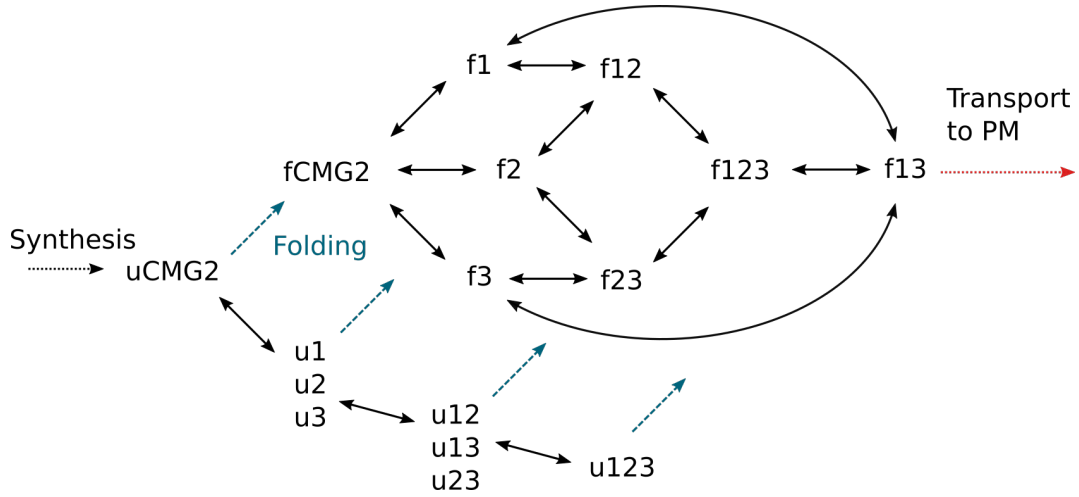


Figure 53: Illustration of the CMG₂ model showing its structure in the ER. The 'u' prefix denotes unfolded CMG₂ and 'f' indicates the folded species. The numbers indicate the palmitoylation sites that are occupied. For example, 'f₁₂' indicates folded CMG₂ which is palmitoylated on both the first and second site.

Because palmitoylation occurs on the cytosolic tail which is intrinsically disordered, the folded and unfolded species have the same palmitoylation and depalmitoylation kinetics. The only difference is that the unfolded species have different decay constants and cannot be transported out of the ER. Again, since palmitoylation occurs on the disordered cytosolic tail, folding was modelled with first order kinetics that are independent from the palmitoylation state.

As was the case with the Calnexin and CLiMP6₃ models, the decay rates depend on palmitoylation, folding and localisation. Palmitoylation and depalmitoylation are modelled with tQSSA and their rates also depend on palmitoylation state and localisation.

Transport from the ER to the Golgi apparatus is not reversible and does not depend on palmitoylation state. It is modelled with first order kinetics. Once in the Golgi, a similar set of interconvertible reactions occur, but catalyzed by zDHHC₃. Unfolded CMG₂ is no longer present, as it cannot be transported from the ER. Figure 54 shows the structure of the model in the Golgi.

Once transported to the PM, CMG₂ can undergo depalmitoylation and degradation but not palmitoylation. As discussed in Section 4.2.6, depalmitoylation of CMG₂ at the PM is linked to anthrax endocytosis, hence an APT was included in this compartment. It also appears that depalmitoylation is central to membrane targeting of CMG₂. For this reason, transport from the Golgi apparatus to the PM was made to be dependent on palmitoylation state. Since no additional PATs have been identified that target CMG₂, and

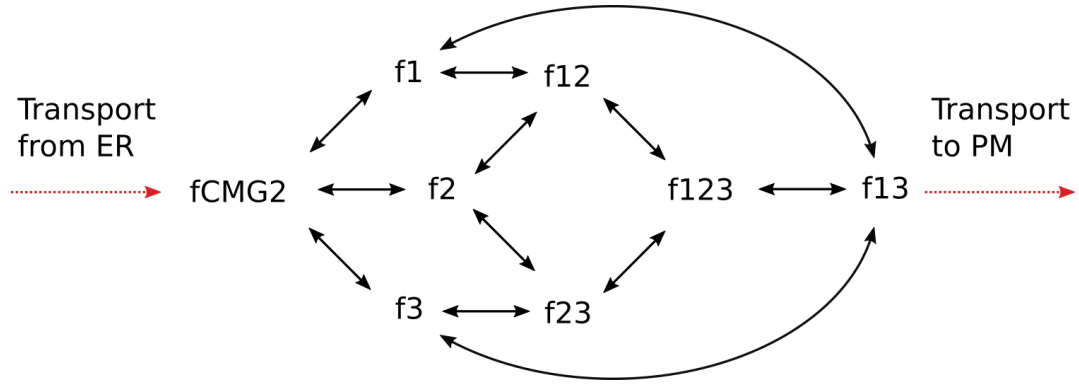


Figure 54: Illustration of the CMG2 model showing its structure in the Gogli apparatus.

because neither zDHHC7 nor zDHHC3 were reported to be present at the PM, no palmitoylation was included in this compartment.

4.4 MODEL CALIBRATION

The same approach was taken to model calibration as for the CLiMP63 model. Namely, rule-based modelling for model creation and CMA-ES for optimisation. The same area-based error function was used for radiolabelling experiments. One difference was that surface biotinylation experiments are available for CMG2. However, due to the nature of the experiment, a different error function was used. Figure 55 shows the two types of error function used for TCE and surface biotinylation experiments. Given the experimental nature of a surface biotinylation experiment, any values inferior to the measured reduction are biologically plausible. Therefore, if the value obtained through model simulation was less than the experimentally measured value, it was assumed to be accurate and the value of the error function was therefore 0, as illustrated in Figure 55. The reason for this behaviour is because the proteins are concentrated during the pull-down step of the surface biotinylation experiment, whereas this is not done when measuring the TCE. It is therefore not possible to accurately and quantitatively compare low values in a surface biotinylation. What can be concluded from this experiment is whether or not the amount of protein at the PM has been significantly affected.

The calibration set was randomly chosen among the experimental data such that it covered approximately a third of the experimental data.

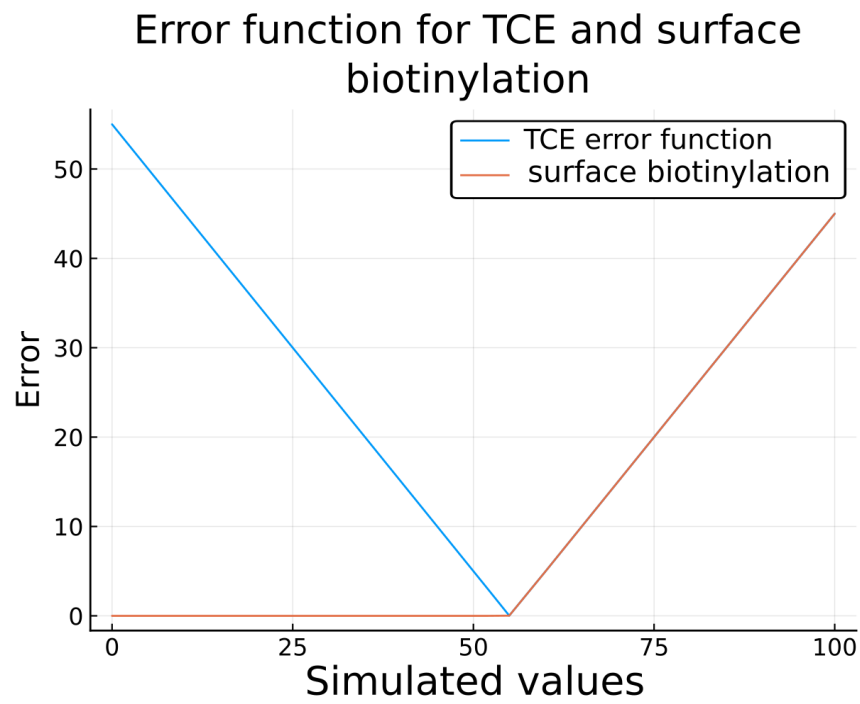


Figure 55: Illustration of the different error functions used for TCE and surface biotinylation experiments. Because the surface biotinylation experiments can not accurately measure small signals, any model output under the measured value is given a zero error.

4.5 RESULTS

4.5.1 Calibration and Validation

Figure 56 shows the calibration set of pulse-chase radiolabelling experiments. Incidentally, this calibration set contains only experiments from the HeLa cell line. A total of 250 models were obtained from the CMA-ES optimisation. Almost all of the experiments in the calibration set are very well reproduced. The only exception is the ^3H pulse-chase with the CAC mutant, which could still be improved.

Figure 58 shows both the calibration and validation set of the TCE and surface biotinylation experiments. Experiments that were used for model calibration are highlighted in red. Only 8 experiments out of a total of 25 were used for calibration. Two variants of the TCE experiment were performed. The first doesn't use any kind of labelling and is called simply TCE, which measures the protein level. By combining ^3H palmitate labelling and TCE, the relative amount of palmitoylated protein can be measured instead.

In the surface biotinylation experiment shown in Figure 58, as discussed in Section 4.4 we can see the effect of the error function used for the surface biotinylation experiments: the model produces values which are much less than those measured. This is to be expected given the error function, as long as the simulated values of the surface biotinylation are less than or equal to the experimental values the biological behaviour is well reproduced by the model. All calibration experiments are well reproduced, with the exception of TCE ^3H palmitate labelling when silencing zDHHHC3. In the validation experiments the trends are generally well reproduced, although the exact values are not perfectly reproduced.

Figure 59 shows the pulse-chase validation experiments performed with the HeLa cell line. All experiments except one are very well reproduced. The good agreement between model and experimental data is also apparent in Figure 61, which shows the pulse-chase validation experiments done in the RPE1 cell line. In this set, the two WT ^3H palmitate labelling experiments out of eight are quite far from the experimental data and could be better reproduced. However, given that about a third of the pulse-chase experiments were put in the calibration set this is still a positive result.

Overall, given the large model size and numerous experiments, these results are promising. They can still be improved in order to reach a better agreement between model and experimental data, but they are already accurate enough to make predictions. Specifically, the dynamic properties of CMG2 captured by the radiolabelling experiments are well reproduced by the model. In particular, improvement could be done in the ^3H -labelling experiments in the RPE1 cell line. Therefore predictions based on these experiments should be experimentally validated. Otherwise, care should be taken when comparing protein levels with different mutants or upon combining

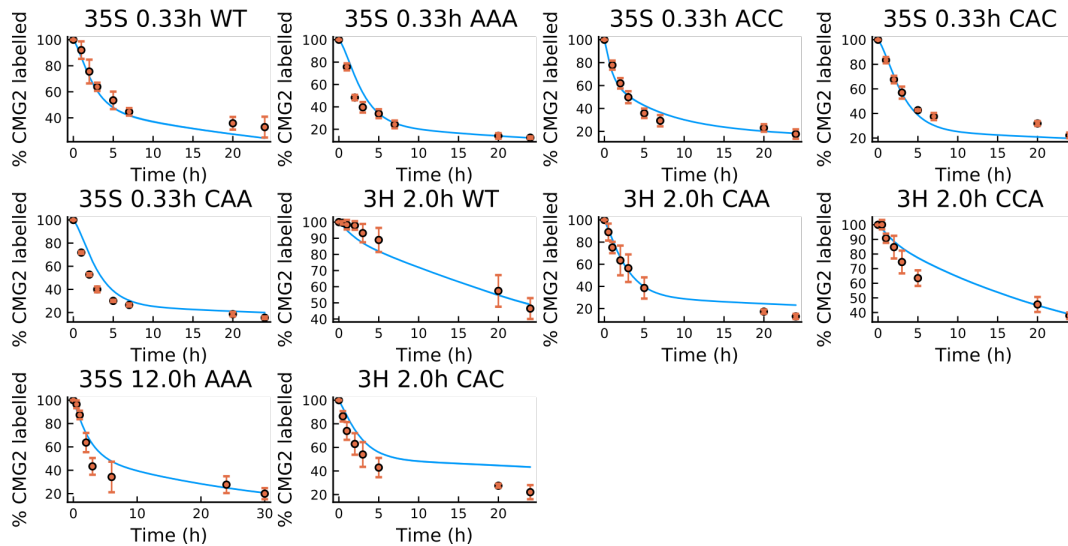


Figure 56: Pulse-chase radiolabelling experiments performed on CMG2. Only the calibration experiments are shown. Experimental results are shown in orange circles, with error bars indicating the standard deviation ($N=3$ for ^{35}S experiments and $N=4$ for ^3H experiments). Solid blue lines show the median of 200 models, with the shaded area representing the first and third quartile. In this case, the models are all very tightly grouped so that the quartiles are not discernible. The type of labelling, the pulse time in hours, as well as any mutants or PAT silencing is indicated in the title of each experiment. The notation 'R3' indicates that zDHHC3 is silenced by RNAi. While the 'R3R7' notation indicates that both zDHHC3 and zDHHC7 are silenced by RNAi.

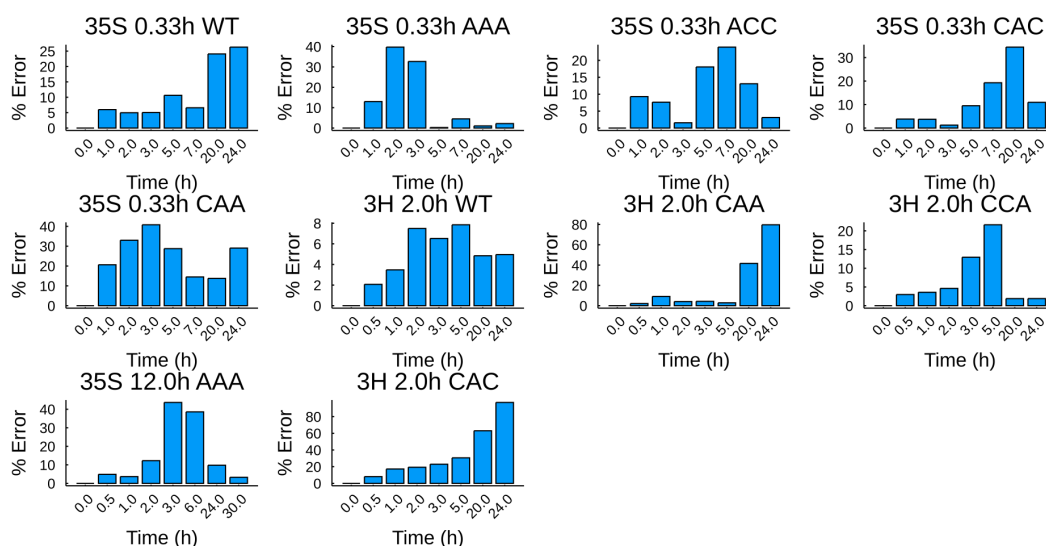


Figure 57: % Error for Pulse-chase radiolabelling experiments performed on CMG2. Only the calibration experiments are shown. The % error is shown between the mean of each experimental data point and the median of the model output. The type of labelling, the pulse time in hours, as well as any mutants or PAT silencing is indicated in the title of each experiment. The notation 'R3' indicates that zDHHC3 is silenced by RNAi. While the 'R3R7' notation indicates that both zDHHC3 and zDHHC7 are silenced by RNAi.

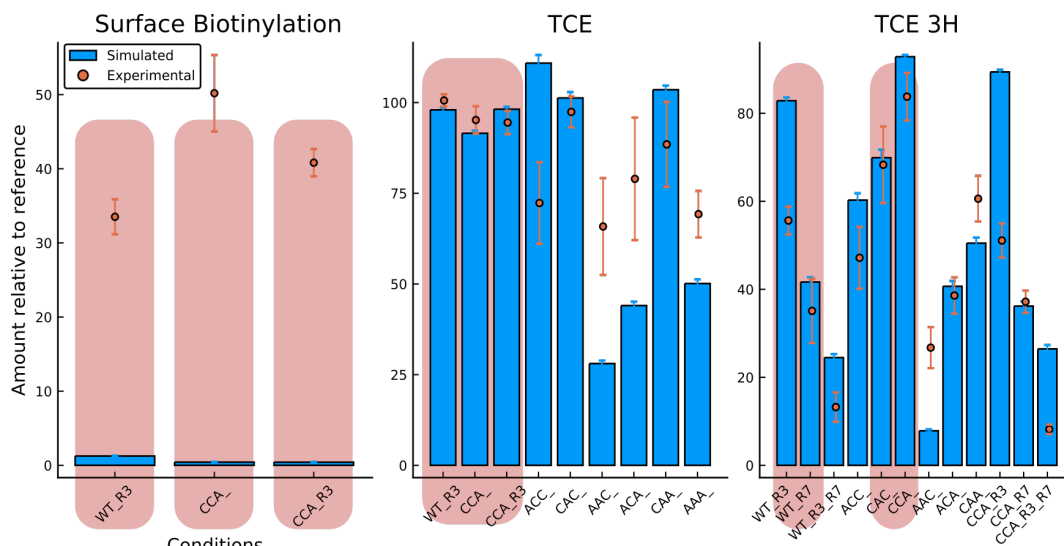


Figure 58: Surface biotinylation and TCE experiments done on CMG2. Experimental results are shown with orange circles, the error bars represent standard deviation ($N=4$ for surface biotinylation, $N=9$ for TCE and $N=4$ for TCE ^3H experiments). Simulated results from the model are shown with blue bars, the error bars correspond to the interquartile range. Calibration experiments are highlighted in red, all other experiments belong to the validation set. Both normal TCE and ^3H -labelling TCE were performed.

enzyme silencing and CMG2 mutants. In all the TCE and surface biotinylation validation experiments, the trends are well reproduced, but in some cases the exact values don't agree with experimental data.

4.5.2 Biosynthetic Flux

One of the key questions regarding CMG2 is how does palmitoylation affect the biosynthetic flux, ultimately leading to mature CMG2 at the PM. In order to better understand this, we used the model to simulate these fluxes while silencing zDHHC7 and zDHHC3. Several Sankey diagrams of the flux between compartments is shown in [Figure 63](#). We can see that in WT conditions, only about half of the synthesized CMG2 arrives at the PM. However, when silencing zDHHC7 this falls to 28%, due to an increased degradation in the ER. When silencing zDHHC3, only 35% reaches the PM, the reduction being caused by an increased degradation in the Golgi. However, when silencing both enzymes this falls to 11%, showing an additive effect of zDHHC7 and zDHHC3 on membrane targeting.

It should also be noted that the predictions in this section rely on the model correctly reproducing the change in CMG2 levels upon enzyme silencing. As discussed in [Section 4.5.1](#), the experimental data which involved silencing zDHHC3 or zDHHC7 was generally well reproduced.

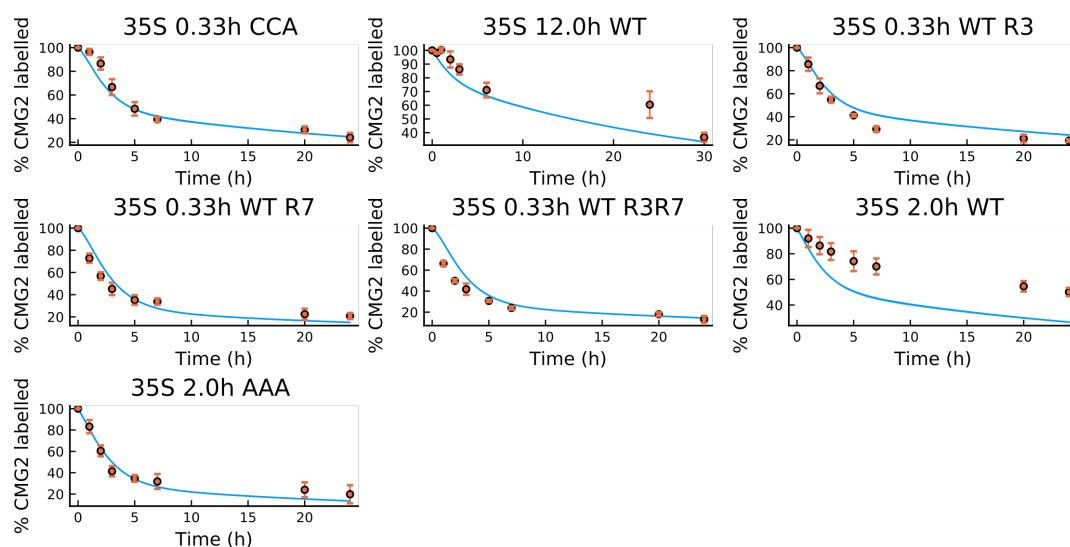


Figure 59: Pulse-chase radiolabelling experiments performed in the HeLa cell line.

Only validation experiments are shown. Experimental results are shown in orange circles, with error bars indicating the standard deviation (N=3). Solid blue lines show the median of 200 models, with the shaded area representing the first and third quartile. In this case, the models are all very tightly grouped so that the quartiles are not discernible. The type of labelling, the pulse time in hours, as well as any mutants or PAT silencing is indicated in the title of each experiment. The notation 'R3' indicates that zDHHC3 is silenced by RNAi. While the 'R3R7' notation indicates that both zDHHC3 and zDHHC7 are silenced by RNAi.

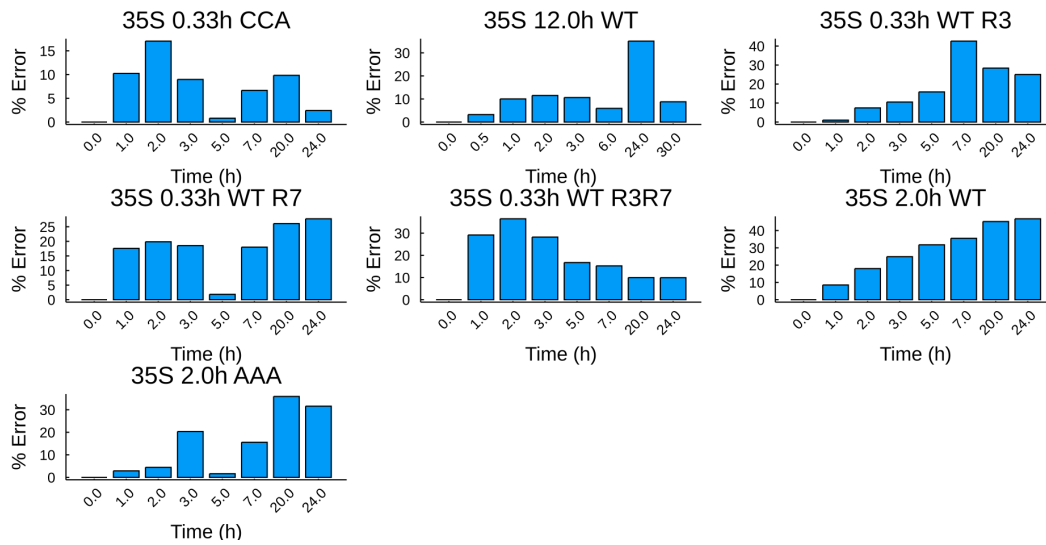


Figure 60: % Error for Pulse-chase radiolabelling experiments performed in the HeLa cell line. Only the validation experiments are shown. The % error is shown between the mean of each experimental data point and the median of the model output. The type of labelling, the pulse time in hours, as well as any mutants or PAT silencing is indicated in the title of each experiment. The notation 'R3' indicates that zDHHC₃ is silenced by RNAi. While the 'R3R7' notation indicates that both zDHHC₃ and zDHHC₇ are silenced by RNAi.

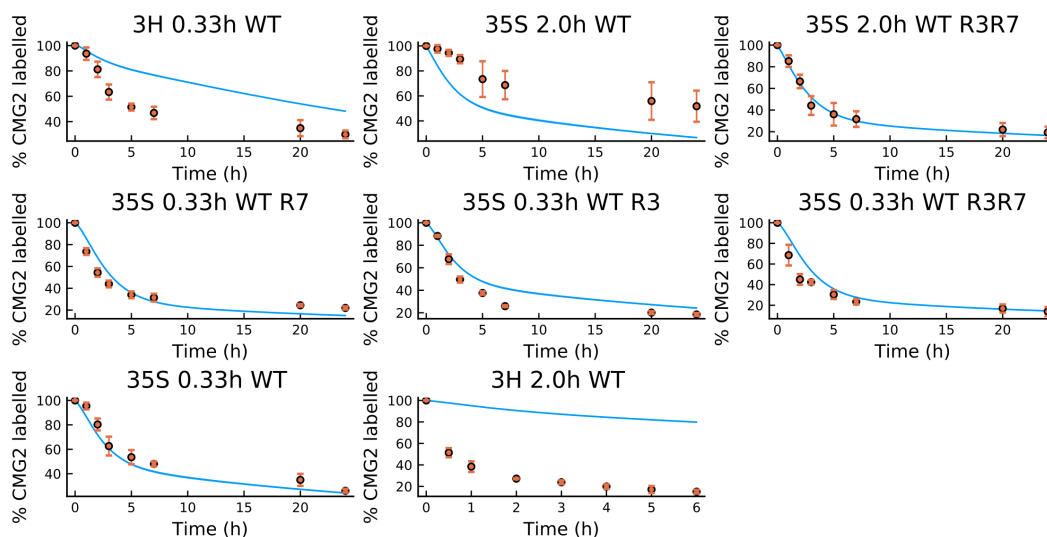


Figure 61: Pulse-chase radiolabelling experiments performed in the RPE1 cell line. Only validation experiments are shown. Experimental results are shown in orange circles, with error bars indicating the standard deviation (N=3 for ^{35}S experiments and N=4 for ^3H experiments). Solid blue lines show the median of 200 models, with the shaded area representing the first and third quartile. In this case, the models are all very tightly grouped so that the quartiles are not discernible. The type of labelling, the pulse time in hours, as well as any mutants or PAT silencing is indicated in the title of each experiment. The notation 'R3' indicates that zDHHC3 is silenced by RNAi. While the 'R3R7' notation indicates that both zDHHC3 and zDHHC7 are silenced by RNAi.

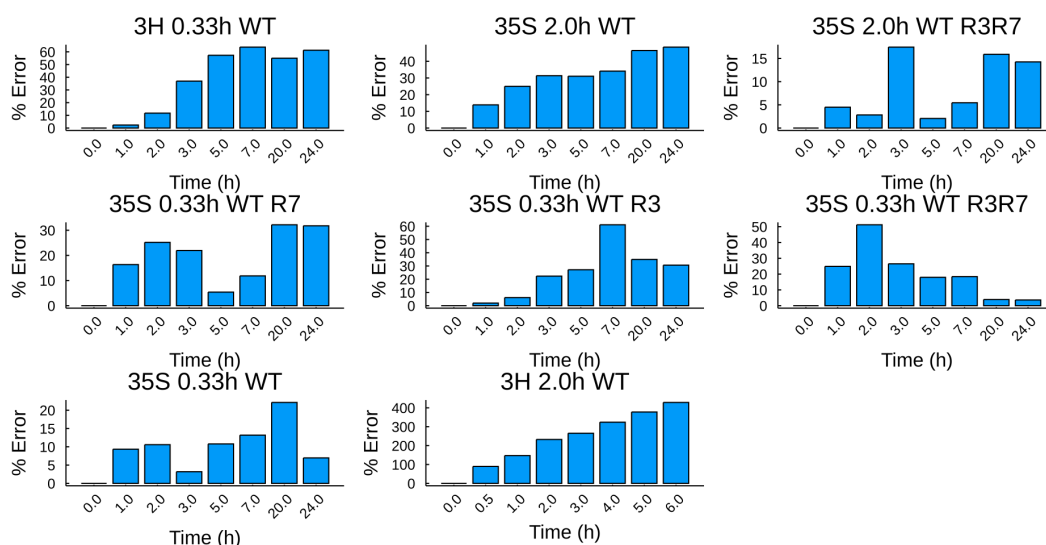


Figure 62: % Error for Pulse-chase radiolabelling experiments performed in the RPE1 cell line. Only the validation experiments are shown. The % error is shown between the mean of each experimental data point and the median of the model output. The type of labelling, the pulse time in hours, as well as any mutants or PAT silencing is indicated in the title of each experiment. The notation 'R3' indicates that zDHHC3 is silenced by RNAi. While the 'R3R7' notation indicates that both zDHHC3 and zDHHC7 are silenced by RNAi.

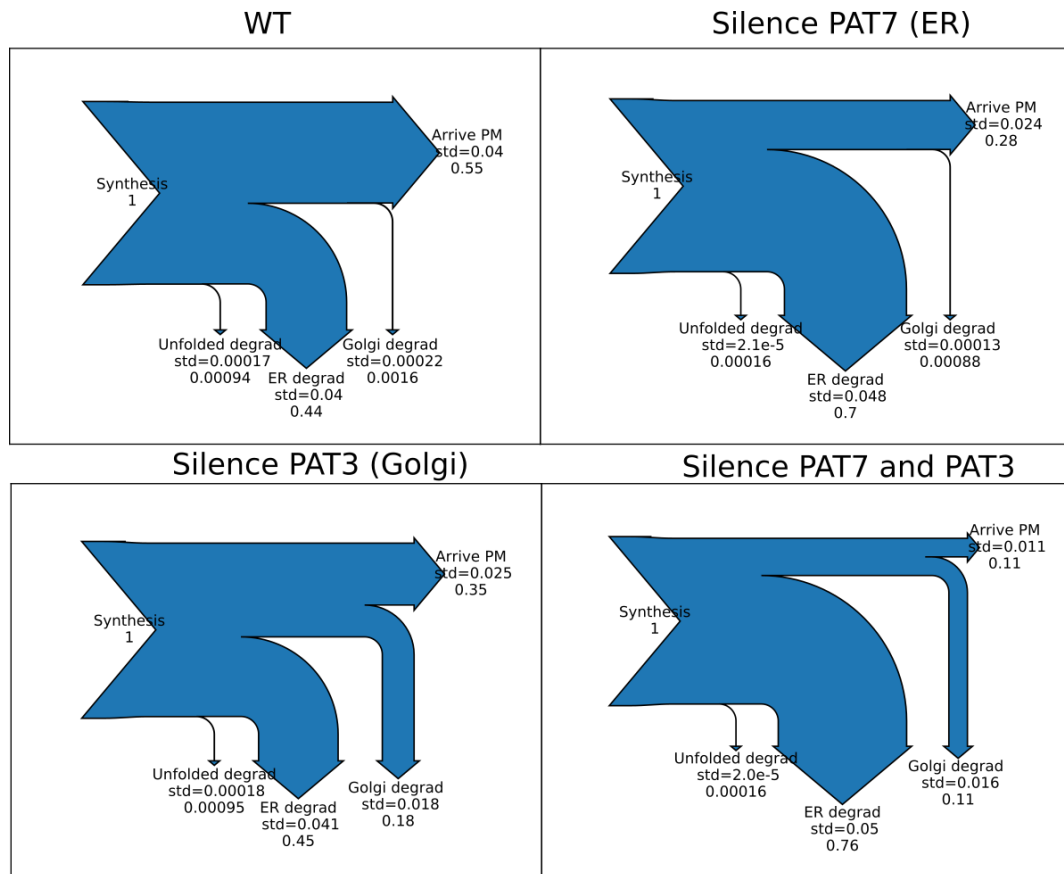


Figure 63: Sankey diagrams showing model predictions of how the synthesis flux of CMG2 is distributed throughout the cell. WT conditions as well as zDHHC7 silencing and zDHHC3 silencing are shown. The median and standard deviation of 200 models are shown on the graph.

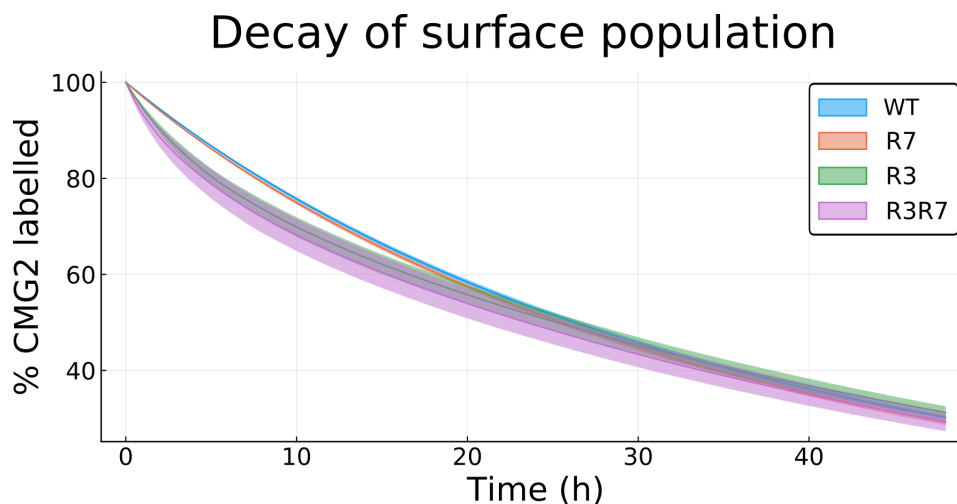


Figure 64: *In silico* ^{35}S labelling of surface-resident CMG2. Both zDDHC7 and zD-HHC3 were silenced and the experiment reproduced in order to estimate the apparent half-life of surface-resident CMG2. For each of the four conditions shown, the values were normalised with respect to the initial value at $t=0$ hours. The solid lines represent the median of 200 models. The shaded area the interquartile range.

4.5.3 Half-life of surface subpopulation

A similar question is how does palmitoylation affect the half life of CMG2 once it arrives at the PM. Using the model, we performed an experiment analogous to ^{35}S labelling but where only the surface resident species were labelled. This was then repeated while silencing zDHHHC7 and zDHHHC3, the results are shown in Figure 64. We can see that the decay of the surface population is only minimally affected when silencing any of the palmitoylation enzymes. This is quite striking considering that silencing either zD-HHC has a strong impact on the amount of CMG2 transported to the PM. However, if we look at the absolute values of these radiolabelling experiments (Figure 65) instead of the normalised ones, we can see that there is still significantly less CMG2 when silencing either zDHHHC, as we previously showed. The reason for the half-life not changing is that the distribution of palmitoylation states does not change. This is shown in Figure 66, where we can see that the total amount is greatly reduced, but that the majority of CMG2 is still in the fully palmitoylated form.

4.5.4 Effect of palmitoylation on surface subpopulation

Given how important depalmitoylation is for anthrax endocytosis, we should also study how this affects CMG2. In Figure 67 we perform the same ^{35}S labelling of surface CMG2 as previously, except that this time we overexpress

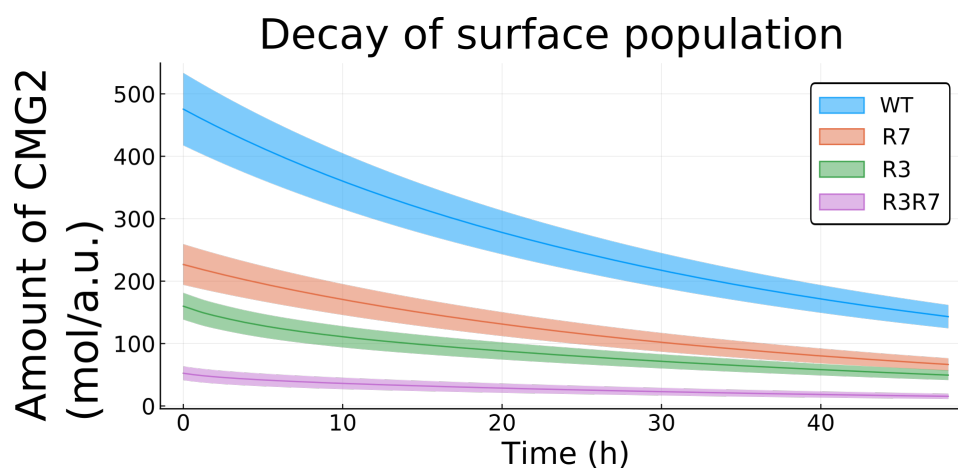


Figure 65: *In silico* ^{35}S labelling of surface-resident CMG2. Total amounts of CMG2 are shown. The solid lines represent the median of 200 models. The shaded area the interquartile range.

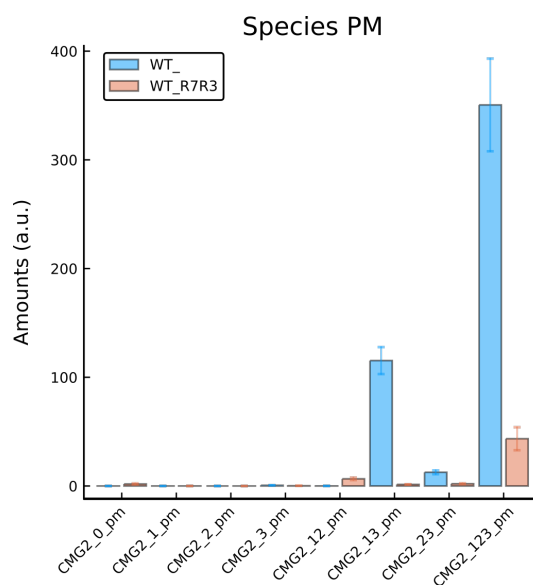


Figure 66: Distribution of palmitoylation states of CMG2 at the PM. WT conditions and silencing both zDHHC7 and zDHHC3 are shown. The mean and standard deviation of 200 models are shown.

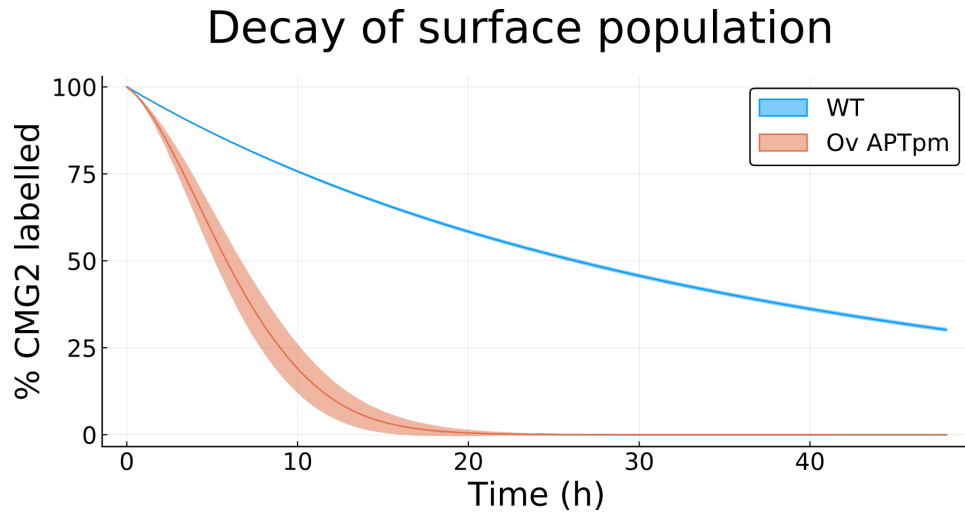


Figure 67: *In silico* ^{35}S labelling of surface-resident CMG2. WT conditions and APT overexpression are shown. The solid lines represent the median of 200 models. The shaded area the interquartile range.

the APT at the PM, to simulate the depalmitoylation that occurs in the presence of PA. We picked a factor of 100 for overexpression, which is comparable to the silencing factor of both zDHHC enzymes. We can see that the half life is now drastically reduced. However, the change in amount of CMG2 is less drastic, as shown in Figure 68. What has changed is the distribution of palmitoylation states. When overexpressing the APT, depalmitoylation is increased so there is much less of the fully palmitoylated form of CMG2.

Half-life is therefore significantly reduced by depalmitoylation of CMG2 at the surface. While the total amount of CMG2 at the surface is only slightly reduced. Furthermore, it isn't possible to achieve the same effect by acting upstream of the PM, either in the Golgi or the ER. This is because the transport between the Golgi and PM is dependent on palmitoylation, leading to a surface population which is mostly fully palmitoylated.

4.6 DISCUSSION

As we have seen in Section 4.5.1, calibration and validation of the CMG2 model was successfully carried out. The radiolabelling experiments describing the dynamic behaviour of CMG2 were very well reproduced, however some of the TCE experiments describing the change in total amounts of CMG2 were not as satisfactory. Although the general trend for these experiments was accurately reproduced by the model, the exact values in some experiments were off.

Using this model, we then quantified how palmitoylation by zDHHC7 and zDHHC3 affects the biosynthetic flux of CMG2. That is, the amount of CMG2 that is synthesized and trafficked to the PM. We showed that both

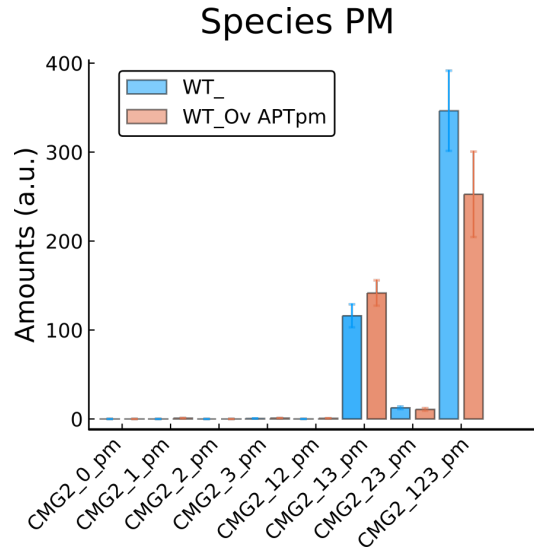


Figure 68: Distribution of palmitoylation states of CMG2 at the PM. WT conditions and APT overexpression are shown. The mean and standard deviation of 200 models are shown.

enzymes have an additive positive effect on the biosynthetic flux, although zDHHC7 has the bigger effect of the two.

We then went on to show that neither zDHHC7 nor zDHHC3 affect the half-life of CMG2 at the PM, but rather it is the total protein level which is modulated. Given the difficulty in calibrating the model, the exact quantitative effect of these enzymes on CMG2 should be considered an estimate. Improvements in the calibration would yield more accurate predictions of the change in protein levels when silencing various enzymes.

Contrary to reducing palmitoylation by silencing zDHHC7 or zDHHC3, we showed that increasing depalmitoylation at the cell surface has a large effect on CMG2 half-life. Higher depalmitoylation drastically reduces the half-life of surface-resident CMG2. However, the total levels of CMG2 are only mildly affected when increasing depalmitoylation, the change in half-life being instead caused by a change in palmitoylation state. Again, due to difficulties in calibration the exact values of the distribution of palmitoylation states under increased depalmitoylation should be interpreted with caution. Although the general trends can be considered trustworthy.

4.7 CONCLUSION

We started the process of modelling CMG2 by assembling all the experimental data from multiple sources and organising it in a coherent way. Not only was this a significant task given the amount of data, but this already gave some insight into the considerations to be taken later during model construction. One of the key questions at this point was whether endoge-

nous CMG2 in RPE1 and CMG2 transfected in HeLa had similar behaviour. A subset of pulse-chase experiments, TCE and surface biotinylation experiments suggested this was the case, and we proceeded with building a single model for both cell lines.

Given the size of the model and the amount of data, we managed a very promising calibration. Using less than a third of the experimental data for calibration, we reproduced almost all the pulse-chase experiments and captured the most important trends in the TCE and surface biotinylation experiments. Although promising, this calibration could still be improved in the future.

We then used the model to study PM targeting of CMG2 and its interplay with palmitoylation. We showed how zDHHC7 and zDHHC3 have an additive effect of increasing the amount of CMG2 at the PM by stabilising it through palmitoylation. Although these enzymes have a large influence on the protein levels at the PM, they have no discernible influence on the half-life of the surface population. This is because the final transportation step from the Golgi to the PM is dependent on palmitoylation, resulting in a majority of fully palmitoylated CMG2 making it to the PM.

In order to simulate the presence of anthrax toxin, we increased depalmitoylation at the PM by overexpressing the relevant APT. We showed that this had a significant effect on the half-life of the surface population by altering the distribution of palmitoylation states. These predictions also make it clear why promoting depalmitoylation at the PM would increase anthrax endocytosis. By making CMG2 less stable without reducing its level significantly there is plenty of CMG2 to bind to, but it is now internalised at a faster rate.

4.7.1 *Future Outlook*

Although a promising calibration was obtained using relatively few experiments, some experiments were not adequately reproduced. This should be improved in the future, by more careful optimisation attempts and by analysing precisely why certain experiments are not well reproduced.

Given the increasing size of the model and amount of experimental data, it could also be worth looking into other modelling paradigms and problem formulations. This may also help make the model more tractable and easier to calibrate.

Another aspect that could be investigated in more depth is the selection of calibration and validation sets. Although the calibration set was randomly chosen to cover approximately a third of the experimental data, this selection could potentially impact calibration quality. It would be of interest try and calibrate the model based on other calibration sets, with the goal of improving the calibration quality. However, some of the chal-

lenges in doing this are the manual interventions needed to obtain a good calibration, and the computational cost due to the size of the model.

There are clearly more questions to be answered by using the CMG₂ model. The issue of specificity between the two zDHHCs and the different palmitoylation sites hasn't been addressed. Indeed, discerning the function of each palmitoylation site would help better understand this protein. This is especially relevant when considering the position of the first two palmitoylation sites next to the transmembrane domain, while the third site is much further along on the disordered cytosolic tail.

The CMG₂ model developed can also serve as a tool in developing an approach to treating HFS. Namely, it can help to better understand dysregulation of CMG₂ in HFS patients. The quantitative nature of kinetic models can help determining which cellular conditions synthesize sufficient functional CMG₂. For example, the different enzyme levels which lead to a sufficient biosynthetic flux which therefore lead to a sufficient amount of CMG₂ at the cell surface so that the ECM may be properly regulated.

This work also raises questions concerning mitigating the toxicity of anthrax. As has been described, depalmitoylation of CMG₂ appears to be essential for toxin endocytosis. A natural mitigation may be reducing the amount of depalmitoylating APT at the PM. However, one of the problems with this approach may be the substrate overlap of the targeted enzymes. In the case of DHHC enzymes, which add a palmitate chain, it has been shown that multiple enzymes of this family may palmitoylate the same substrate, and the specificity is not clearly defined nor understood [61, 125]. The family of APTs, which carry out depalmitoylation, are not as well studied as the DHHC family. However, if they also possess substrate overlap, a therapy targeting a specific APT may target other proteins and have a very broad range of side effects. Although the work here has shown how depalmitoylation can clearly favour toxin endocytosis through increased turnover but similar protein levels, more research in the APTs is needed to find a potential mitigation of anthrax toxicity.

This chapter is a manuscript in preparation.

In it, we will discuss the concept of Global Sensitivity Analysis (GSA) (first introduced in [Section 2.4](#)) in the context of large-scale kinetic models of metabolism. We will discuss some of the challenges associated with large-scale metabolic models, and how GSA can be of use in refining these models and guiding experimental efforts. We will also show some of the difficulties that can be encountered when trying to use GSA with large-scale models and how we have devised a workflow to address these issues.

5.1 ABSTRACT

Kinetic models of metabolism can elucidate dynamic behaviour and essential cellular regulatory schemes, thus providing useful information for metabolic engineering and fundamental biology. The construction of these models is challenging due to the numerous parameters and partially known kinetic mechanisms. Various computational frameworks surmount this by sampling unknown parameters, which can create large uncertainties in model properties. Although local sensitivity analysis may identify sources of uncertainty in smaller models, a holistic approach is needed when dealing with high dimensions and wide parameter ranges. One such approach is Global Sensitivity Analysis (GSA). However, it is rarely applied to large-scale models due to the extensive sampling needed.

We herein present a workflow for efficiently applying GSA to large-scale kinetic models in order to identify predominant sources of uncertainty. A published kinetic model of aerobically grown *E.coli* was used to demonstrate how the proposed workflow can be used to analyze the sensitivity of Metabolic Control Analysis (MCA) control coefficients to kinetic parameters. The sensitivity of the control coefficients can be studied at different levels of resolution depending on the study requirements.

When applying this workflow to an *E. coli* kinetic model, we were able to identify the largest sources of uncertainty. These were then validated *in silico*, leading to a reduction in variance of 75%. This workflow can be applied to any other kinetic model and could even benefit other fields, especially when performing sensitivity analysis in high-dimensional spaces. We also found that sources of uncertainty are spread out across the model, making the identification of uncertainty sources harder as model size grows.

5.2 AUTHOR SUMMARY

Kinetic models of metabolism can provide highly useful information for the fields of metabolic engineering and fundamental biology. However, only partial information is currently available for creating large-scale kinetic models. This can be surmounted by sampling the missing information, although this leads to uncertainty in the behaviour of the resulting model. We present a workflow that uses GSA in order to target the largest sources of uncertainty, providing a ranking of which parameters to measure in order to achieve the highest reduction in uncertainty. We applied this framework to a kinetic model of the bacterium *E. coli*, showing that it was possible to achieve reductions in model uncertainty of 75% if certain parameters could be accurately estimated. Due to the flexible nature of GSA and our workflow, it is in principle possible to apply to any kinetic model. We also show that metabolism is a highly integrated system, where sources of uncertainty from different parts of metabolism interact to contribute to uncertainty in model behaviours.

5.3 INTRODUCTION

New computational frameworks are enabling the construction of genome-scale kinetic models that are consistent with stoichiometric, thermodynamic and physiological constraints [36]. Despite the increasing availability of experimental data concerning kinetic parameters [126–128] and methods to estimate these [129], significant uncertainty in their nominal values remains [130]. Adding to this uncertainty, the number of kinetic parameters increases with the size of the models. To overcome the problem of assigning unique kinetic parameter values, one solution is to sample the kinetic parameter space, generating multiple alternative models [38, 131–133]. However, this uncertainty in kinetic parameters can result in kinetic models with contradicting properties and conclusions [134]. Gutenkunst *et al.* [135] suggest that models in systems biology are ‘sloppy models’ and that usually only few parameters affect model outputs, suggesting that identifying these parameters is a worthwhile pursuit. Hence, classifying kinetic parameters by their impact on uncertainty in model outputs is essential for improving model predictions.

Different sensitivity analysis approaches can be used to trace a perturbation in model inputs to its effect on model outputs [136–138]. Variance-based GSA is one of the most established techniques for performing sensitivity analysis on nonlinear systems [139]. GSA can be used to estimate first order sensitivity indices (S_i), which quantify the average reduction in variance of a model output when fixing an input parameter. GSA also makes it possible to estimate total effect sensitivity indices (S_t), which describe how the perturbation of an input parameter is amplified or attenuated by the

values of one or several other input parameters. The word ‘Global’ refers to the whole parameter space being considered, in contrast to local methods which only focus on a single point. By using variance as an estimator for uncertainty, these sources of uncertainty can be efficiently identified and ranked. The model itself is treated as a black box, making GSA applicable to many types of models. Kiparissides and Hatzimanikatis developed a GSA-based procedure for analyzing genome-scale stoichiometric models that have thermodynamic constraints [140]. However, to our knowledge GSA has not been applied to genome-scale nonlinear kinetic models.

On a smaller scale, both local and global sensitivity analysis methods have been applied to models up to the size of pathways and subsystems [139]. However, due to the high dimension and wide range of the parameter space found in genome-scale kinetic models, a global approach that considers the entire parameter space becomes necessary. Performing GSA on genome-scale kinetic models of metabolism is challenging and computationally expensive. We utilize the Optimization and Risk Analysis of Complex Living Entities (ORACLE) framework to generate populations of kinetic models [38, 133]. ORACLE implicitly samples the kinetic parameter space by explicitly sampling enzyme saturations, this allows a better understanding of the enzyme state, which is a physiological property. The kinetic parameters can then be used to calculate Control Coefficient (CC)s using MCA [141–143]. For the scope of this work, the inputs of MCA are thus enzyme saturation levels and the outputs CCs. Hence, we developed a variance-based GSA approach for assessing the sensitivity of CCs to enzyme saturations.

We perform GSA on an *E.coli* model that was reduced [144, 145] from the iJO1366 genome-scale model [146]. The model consists of 271 enzymatic reactions, 247 lumped reactions and 160 metabolites, resulting in a total of 3083 enzyme saturation levels to be sampled. In order to study the total and first order effects of enzyme saturations on Flux Control Coefficient (FCC), Variance-based sensitivity indices were computed based on the Sobol method [147]. Exhaustively calculating sensitivity indices for each enzyme saturation would require considerable computational efforts. Instead, we first developed a workflow for identifying parts of the network that contribute the most to the variance of model outputs using a coarse-grain sampling approach. Once we have identified these parts of the network, we can perform a fine-grain sampling of the input parameters to identify the ones contributing the most to the variance of model outputs. The workflow was used to rank input parameters based on their contribution to the variance of MCA outputs and can be applied to any large-scale nonlinear kinetic model.

5.4 RESULTS AND DISCUSSION

We have developed a workflow combining variance-based global sensitivity analysis and metabolic control analysis, named Global Sensitivity of Metabolic Control Analysis (GMCA). This workflow requires a kinetic model for a given physiology and allows a flexible definition of study scope. The study scope describes which parts of the metabolic network to study and at which resolution. The resolution can be adjusted from groups of reactions, to individual reactions or even individual enzyme kinetic parameters. The first step involves sampling kinetic parameters in order to characterize the solution space. After this, new populations of kinetic parameters are re-sampled according to the study scope. These multiple populations are then used to compute sensitivity indices, as shown in [Section 5.6.2](#). Based on these sensitivity indices, parameters can be ranked by their effect on the variance of kinetic model outputs, ie. the Flux Control Coefficient (FCC)s computed with MCA. In this way, the propagation of uncertainty from kinetic parameters to FCCs is mapped out. *In silico* validation can then be performed by reducing the variance of the kinetic parameters with high sensitivity indices and verifying that the variance in FCCs is also reduced. These findings can be used to guide experimental efforts to measure enzyme kinetic parameters as well as metabolic engineering decisions.

5.4.1 Illustrative example

Using the illustrative model in [Figure 70A](#) we would like to study the flux through reaction v_4 , this flux can be modeled using irreversible Michaelis-Menten enzyme kinetics [[148–150](#)]:

$$v_4 = V_{\max,4} \frac{m_4/K_{m,4}}{1 + m_4/K_{m,4}} \quad (45)$$

Where m_4 is the concentration of the metabolite, and $K_{m,4}$ the Michaelis-Menten coefficient of m_4 with respect to the enzyme catalyzing the reaction v_4 . Using techniques such as FBA [[151, 152](#)] and TFA [[153, 154](#)] we can estimate the values of both v_4 and m_4 . However, this still leaves both $V_{\max,4}$ and $K_{m,4}$ as unknowns. If one of them could be estimated, the previous Michaelis-Menten equation could be used to calculate the remaining unknown parameter. The ORACLE workflow [[38, 133](#)] solves this by implicitly sampling $K_{m,4}$. This is done by introducing the saturation term σ

$$\sigma = \frac{m_4/K_{m,4}}{1 + m_4/K_{m,4}} \quad (46)$$

Unlike the concentration or the Michaelis-Menten coefficient, the saturation term is well bounded $\in [0, 1]$. We will therefore sample this saturation

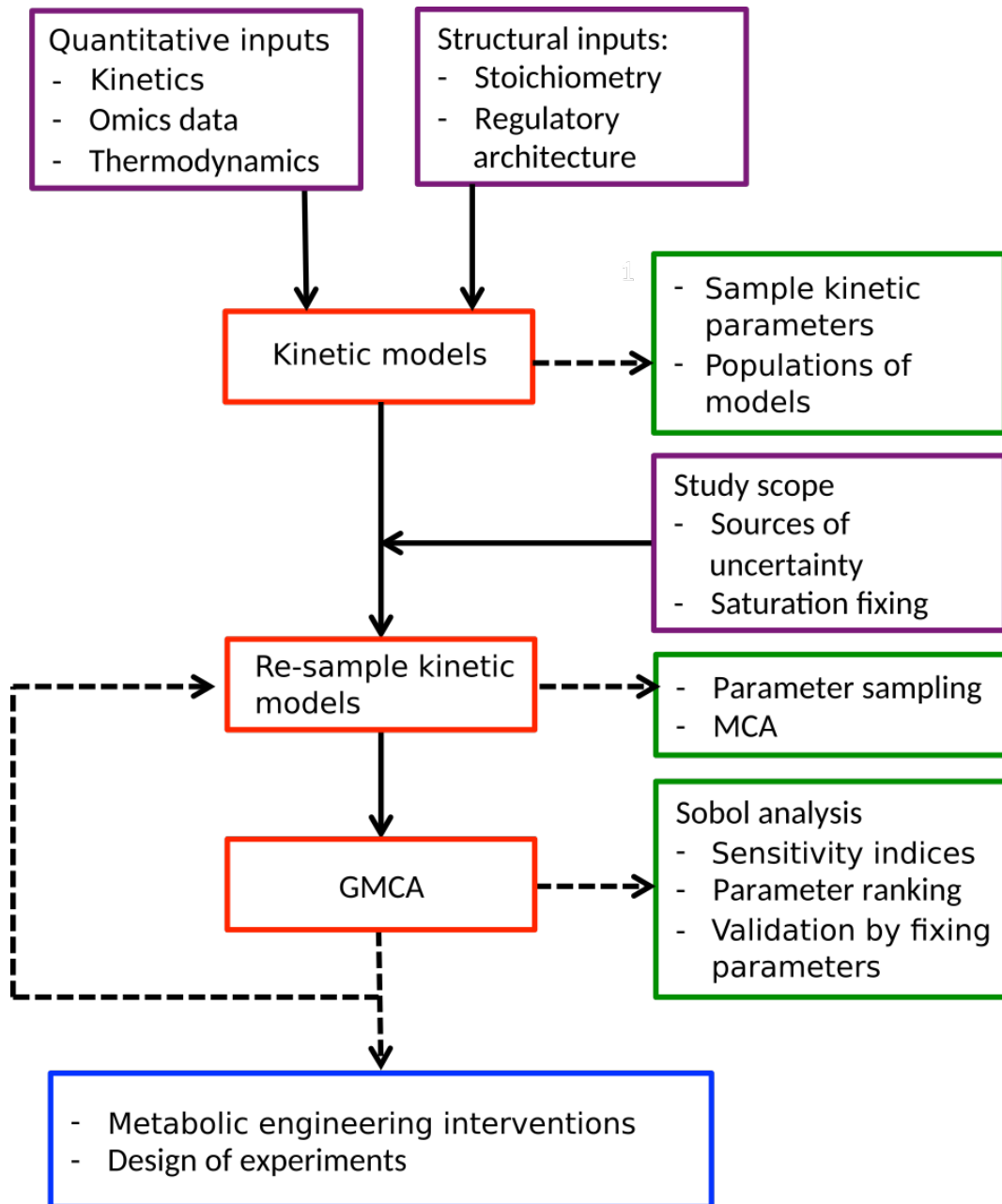


Figure 69: Global Sensitivity of Metabolic Control Analysis (GMCA) workflow for characterizing sources of uncertainty in large-scale kinetic models. Diagram providing details of the various steps required for the characterization of parameters responsible for variance in kinetic models and their outputs.

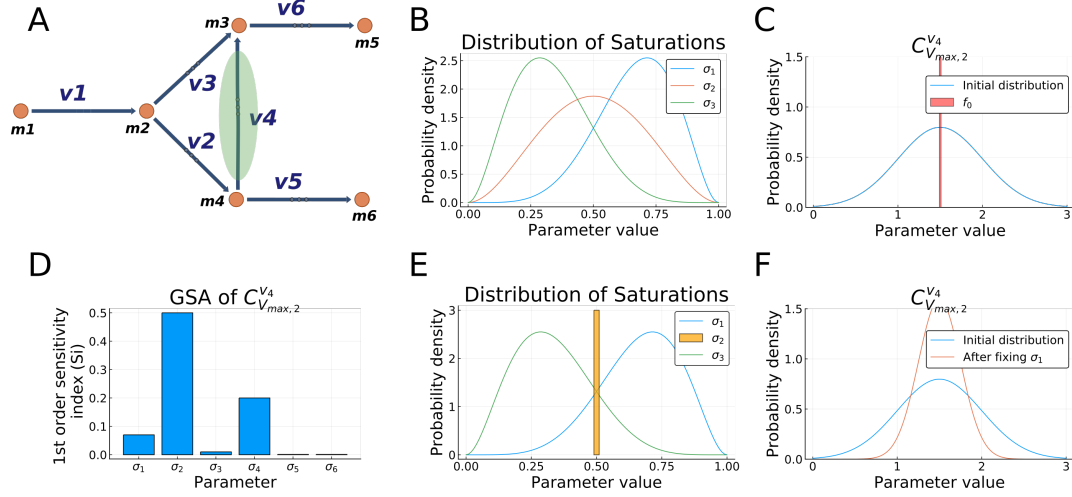


Figure 70: Illustration of applying the GMCA workflow to a branched pathway model. (A) Overview of the branched pathway used for illustrative purposes. The parameters controlling the reaction v_4 will be further studied. (B) Distribution of 3 of the total 6 saturations (σ) present in the model, the other 3 saturations were omitted for clarity. Beta distributions were chosen for illustration purposes. Their values are obtained by sampling according to the ORACLE workflow. (C) Distribution of the control coefficient: $C_{V_{max,2}}^{v_4}$, the control of the flux of reaction v_4 with respect to V_{max} of reaction v_2 . A normal distribution with mean 1.5 and standard deviation of 0.5 was chosen for illustration purposes. The mean (f_0) is shown in red. (D) Global Sensitivity Analysis (GSA), showing the first order sensitivity index (Si) of $C_{V_{max,2}}^{v_4}$ with respect to the 6 saturations present in the model. (E) After having determined that σ_2 had the highest Si, we assume to have determined its value exactly. (F) Resulting distribution of $C_{V_{max,2}}^{v_4}$ once the value of σ_2 is determined.

term in order to obtain a population of possible saturations, as shown in [Figure 70B](#). Once we have obtained samples of the saturation term, we can use them to determine the value of $K_{m,4}$:

$$\frac{m_4}{K_{m,4}} = \frac{\sigma}{1 + \sigma} \quad (47)$$

Where the value of m_4 can be determined using methods such as TFA. We can finally use [Equation 45](#) in order to calculate the remaining unknown $V_{max,4}$.

We then want to study the effect that the parameter $V_{max,2}$ has on the flux through v_4 . To do this, we will use the formalism of MCA [[141–143](#)]. Using this framework, we can use the previous information we have obtained with the ORACLE workflow to calculate control coefficients. These come in two kinds, either a Concentration Control Coefficient (CCC), denoted by C_p^x , or a Flux Control Coefficient (FCC), denoted by C_p^v . They describe the fractional change in a metabolite concentration or reaction flux in response to a fractional change in a kinetic parameter:

$$C_p^x = \frac{d \ln x_i}{d \ln p} \quad (48)$$

$$C_p^v = \frac{d \ln v}{d \ln p} \quad (49)$$

$$(50)$$

Further discussion of both concentration and flux control coefficients can be found in [Section 5.6.1](#).

Since we have a distribution of possible saturations, the control coefficients have an associated distribution rather than fixed values, shown in [Figure 70C](#). We will perform Global Sensitivity Analysis (GSA) in order to identify the saturations that are responsible for the variance of the control coefficients. To do this we perform an initial sampling of the saturations, and two further resamplings, where only a subset of the saturations are re-sampled. This yields three matrices whose columns refer to each saturation parameter, and whose rows refer to each sample. In this example, we would like to study the contribution of σ_2 to the uncertainty in $C_{V_{max,2}}^{v_4}$, hence the following resamplings are performed:

$$\Sigma_A = (\sigma_{a,1} \cdots \sigma_{a,6}) \quad (51)$$

$$\Sigma_{B2} = (\sigma_{b,1}, \sigma_{a,2}, \sigma_{b,3} \cdots) \quad (52)$$

$$\Sigma_{C2} = (\sigma_{a,1}, \sigma_{c,2}, \sigma_{a,3} \cdots) \quad (53)$$

$$(54)$$

To construct the matrix Σ_{B2} , the column $\sigma_{a,2}$ is taken from the matrix Σ_A and all other columns are resampled. To construct the matrix Σ_{C2} , all columns except $\sigma_{a,2}$ are taken from Σ_A and only the column is $\sigma_{c,2}$ resampled.

We then can then use these matrices of saturations to calculate the corresponding vectors of flux control coefficients:

$$\mathbf{c}_A = f(\Sigma_A) \quad (55)$$

$$\mathbf{c}_{B2} = f(\Sigma_{B2}) \quad (56)$$

$$\mathbf{c}_{C2} = f(\Sigma_{C2}) \quad (57)$$

$$(58)$$

The sensitivity indices for a particular saturation σ_j are calculated as follows:

$$f_0 = \sum_i^n \mathbf{c}_{A,i}/n \quad (59)$$

$$S_i = \frac{V_{\sigma_j}(E_{\sigma_j}(c|\sigma_j))}{V(c)} = \frac{\mathbf{c}_A \mathbf{c}_{B_j}^T / n - f_0^2}{\mathbf{c}_A \mathbf{c}_A^T / n - f_0^2} \quad (60)$$

$$St = 1 - \frac{V_{\sigma_j}(E_{\sigma_j}(c|\sigma_j))}{V(c)} = 1 - \frac{\mathbf{c}_A \mathbf{c}_{C_j}^T / n - f_0^2}{\mathbf{c}_A \mathbf{c}_A^T / n - f_0^2} \quad (61)$$

In this example, the vectors \mathbf{c}_A , \mathbf{c}_{B2} , \mathbf{c}_{C2} are sufficient to calculate the sensitivity of $C_{V_{\max,2}}^{v_4}$ with respect to σ_2 . For every parameter other than σ_2 , the resamplings need to be performed and the control coefficients calculated again as shown.

The first order sensitivity index S_i describes the average reduction in the variance of the chosen control coefficient when fixing a specific saturation parameter σ_j . In [Figure 70D](#) the first order sensitivity indices S_i are shown for all 6 saturation parameters in the model. Out of these, σ_2 has the highest S_i of 0.5. This means that fixing σ_2 to any exact value will reduce the variance in $C_{V_{\max,2}}^{v_4}$ by 50% on average. In [Figure 70E](#), the fixing of σ_2 is shown, and the accompanying reduction in variance of $C_{V_{\max,2}}^{v_4}$ is shown in [Figure 70F](#).

In addition to this, the total effect sensitivity index St includes the interaction effects with other saturation parameters in the model. For example, the contribution of σ_2 to the uncertainty in $C_{V_{\max,2}}^{v_4}$, will depend on the value of σ_4 . If σ_4 is at maximum saturation, it won't be possible to increase the flux v_4 , therefore any change in σ_2 will have less effect on $C_{V_{\max,2}}^{v_4}$ than otherwise. In this case, this is a second order interaction, the effect of one parameter depends on the value of another. However, the effect of one

parameter may depend on the values of multiple other parameters, as in multi-step biosynthesis pathways for example. The total effect sensitivity index St includes not only second order but all higher order interactions.

Si should by definition always be lower or equal to St . Similar values for Si and St indicate that interaction effects are negligible. Whereas a value of St much greater than Si indicates that interaction effects strongly influence the uncertainty in model output.

By calculating sensitivity indices, it is possible to efficiently propose target parameters for experimental studies that will best elucidate control of fluxes and concentrations within the model. A high first order sensitivity index will identify parameters that will, on average, cause the largest reduction in uncertainty when their value is precisely determined.

5.4.2 *Kinetic model*

We used a published [155] kinetic model, describing the physiology of aerobically grown *E. coli* for the purpose of this study. The kinetic model had 3083 parameters to be sampled. To characterize the kinetic parameter space, we used the ORACLE workflow for constructing populations of kinetic models [36, 38, 133, 143, 156–158]. The same kinetic mechanisms and steady states for the metabolite concentrations and fluxes were used for this kinetic model as the ones provided in the publication [155], which drew on previous models and data from [146, 159].

5.4.3 *Uncertainty in control of the pentose phosphate pathway*

We decided to focus on control of the Pentose Phosphate Pathway (PPP), containing 12 reactions for a total of 144 FCCs. In order to better quantify the control within this subsystem, we wanted to determine which saturations are responsible for the variance in the control coefficients. To do this, we first grouped kinetic parameters together at three different levels of resolution. The first level groups together all parameters belonging to a subsystem, for example Glycolysis or PPP. The second level groups parameters that belong to the same reaction; the number of parameters varying according to the mechanism of each reaction Table 6. The third level looks at each kinetic parameter individually. In this way, we avoid needing to exhaustively perform re-sampling for every kinetic parameter. Instead, we first identify and rank the subsystems from which most of the uncertainty originates, then we rank the reactions belonging to the top subsystem, and finally the individual kinetic parameters from the top reactions. This workflow also has the benefit of first providing an overview at the level of the subsystems, while increased resolution can be achieved through further numerical simulations. We performed GMCA according to this suggested workflow in order

Reaction	# of Km	Km Parameters
EDA	3	$K_{\text{EDA},2\text{ddg6p}}, K_{\text{EDA},\text{g3p}}, K_{\text{EDA},\text{pyr}}$
EDD	2	$K_{\text{EDD},6\text{pgc}}, K_{\text{EDD},2\text{ddg6p}}$
FBA ₃	3	$K_{\text{FBA}_3,\text{s17bp}}, K_{\text{FBA}_3,\text{dhap}}, K_{\text{FBA}_3,\text{e4p}}$
G6PDH _{2r}	4	$K_{\text{G6PDH}_{2r},\text{g6p}}, K_{\text{G6PDH}_{2r},\text{nadp}}, K_{\text{G6PDH}_{2r},6\text{pgl}}, K_{\text{G6PDH}_{2r},\text{nadph}}$
GND	4	$K_{\text{GND},\text{nadp}}, K_{\text{GND},6\text{pgc}}, K_{\text{GND},\text{nadph}}, K_{\text{GND},\text{ru5p-D}}$
PFK ₃	4	$K_{\text{PFK}_3,\text{atp}}, K_{\text{PFK}_3,\text{s7p}}, K_{\text{PFK}_3,\text{adp}}, K_{\text{PFK}_3,\text{s17bp}}$
PGL	2	$K_{\text{PGL},6\text{pgl}}, K_{\text{PGL},6\text{pgc}}$
RPE	2	$K_{\text{RPE},\text{ru5p-D}}, K_{\text{RPE},\text{xu5p-D}}$
RPI	2	$K_{\text{RPI},\text{ru5p-D}}, K_{\text{RPI},\text{r5p}}$
TALA	4	$K_{\text{TALA},\text{g3p}}, K_{\text{TALA},\text{s7p}}, K_{\text{TALA},\text{e4p}}, K_{\text{TALA},\text{f6p}}$
TKT ₁	4	$K_{\text{TKT}_1,\text{r5p}}, K_{\text{TKT}_1,\text{xu5p-D}}, K_{\text{TKT}_1,\text{g3p}}, K_{\text{TKT}_1,\text{s7p}}$
TKT ₂	4	$K_{\text{TKT}_2,\text{e4p}}, K_{\text{TKT}_2,\text{xu5p-D}}, K_{\text{TKT}_2,\text{f6p}}, K_{\text{TKT}_2,\text{g3p}}$

Table 6: Reactions in the Pentose Phosphate Pathway (PPP) and their corresponding Km parameters. The Km parameters are grouped together when studying the contribution of an entire reaction to the uncertainty of a control coefficient.

to rank all kinetic parameters based on their contributions to uncertainty in the FCCs of the PPP.

In order to efficiently sample the kinetic parameter space, we first use TFA [153, 154] to sample the flux and metabolite concentration space in order to obtain a representative sample. The representative metabolite concentrations can then be used to calculate the reaction’s thermodynamic displacement from equilibrium. We sample enzyme saturations due to their well-bounded nature. These sampled enzyme saturations are then used together with a representative flux and concentration profile in order to calculate the values of the Michaelis constants (K_m) [36]. Finally, the value of V_{max} can be calculated to fit with the representative fluxes (Figure 71). The reader is referred to Section 5.6.1 for more details.

5.4.3.1 Computation of sensitivity indices

The bounds of the enzyme saturations for all reactions were initially left unconstrained such that we sample them uniformly between 0 and 1, allowing us to consider the full range of kinetic parameter values. We sampled a population of 200’000 kinetic models and then computed the first order (S_i) and total effect (S_t) sensitivity indices of the chosen FCCs using the Sobol method, as shown in Section 5.6.2. All these 200’000 models are biologically

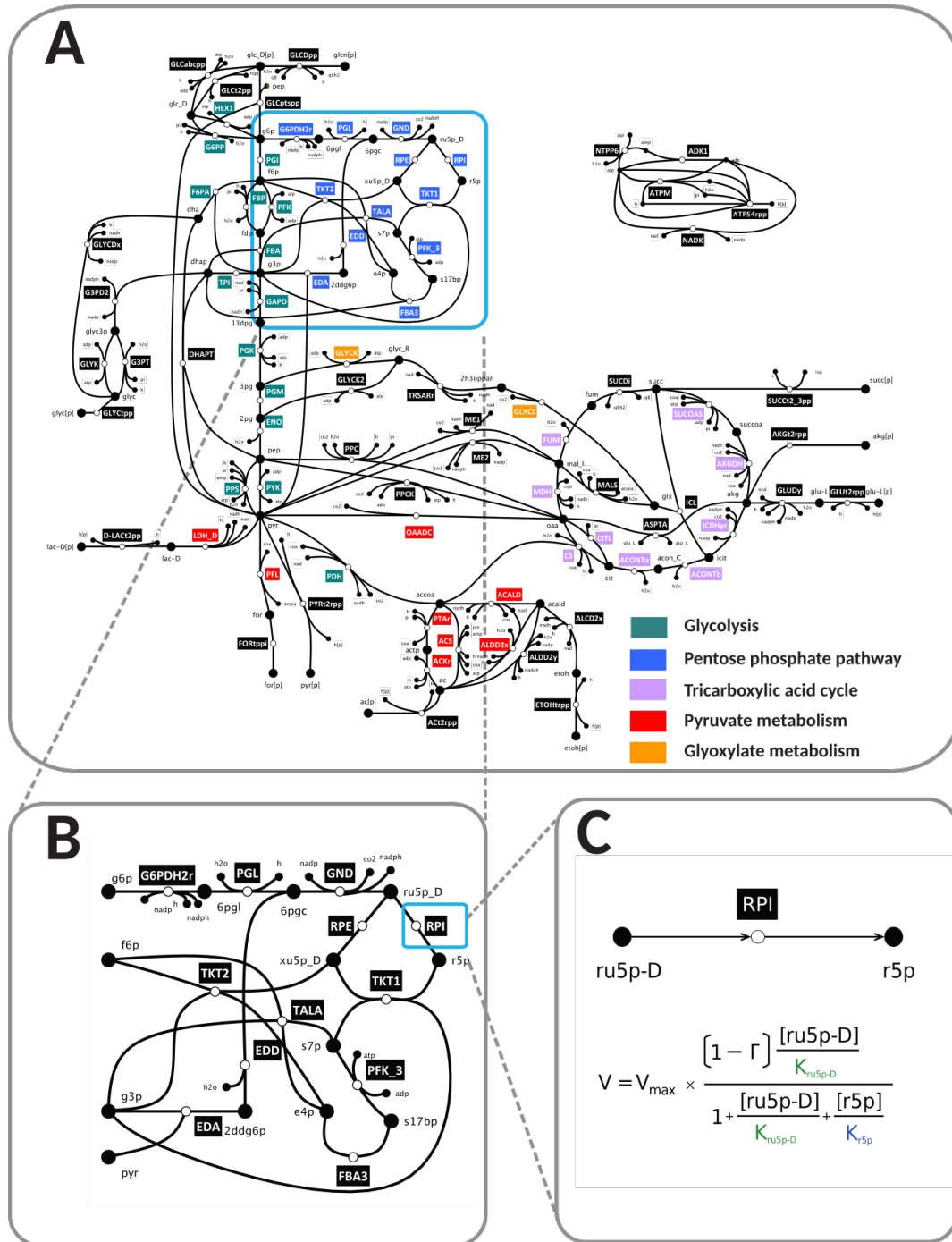


Figure 71: E.coli network diagram illustrating the reactions of the kinetic model. Diagram does not include all the reactions of the model. (A) Overview of the reaction network, color-coding indicates to which subsystem a reaction belongs. Abbreviations are as follows: G/G - Glycolysis and Gluconeogenesis, PPP - Pentose Phosphate Pathway, TCA - Tricarboxylic Acid Cycle, Pyr - Pyruvate Metabolism, Glyox - Glyoxylate Metabolism. The reactions indicated in blue correspond to the Pentose phosphate pathway (PPP) subsystem, which was the focus of this study. A larger version is available in the Appendix (Figure 75). (B) Focus on the PPP subsystem. (C) Single reaction belonging to the PPP, the kinetic mechanism of the reaction is shown, the two kinetic parameters to be sampled using the ORACLE workflow are highlighted in blue and green.

plausible and agree with the results from TFA, namely metabolite concentrations and fluxes. This number was chosen so as to be high enough to give a small enough error on the calculated sensitivity indices.

We started at the first level of resolution by grouping parameters together by subsystem. Sobol sensitivity indices were computed for all FCCs in the PPP. We chose the control coefficient $C_{\text{GND}}^{\text{EDA}}$ for illustration purposes, which describes the control of the EDA reaction flux by $V_{\text{max,GND}}$ (Figure 71). For this control coefficient we found that S_i was negligible for all 5 subsystems except PPP (Figure 72), while S_t was high for all subsystems. This suggests that the interactions between subsystems are significantly more important than their 1st order contributions. In other words, fixing all saturations in the PPP would cause only a small reduction, about 10%, in the variance of $C_{\text{GND}}^{\text{EDA}}$. Hence, determining the kinetic parameters of the PPP alone would only slightly reduce the uncertainty in the chosen FCC, due to uncertainty in the other subsystems.

As the S_i values were all close to zero (Figure 72), we hypothesized that one or several important parameters outside of the study scope were causing this effect. In order to limit the study scope to the 5 subsystems (Glycolysis, PPP, Tricarboxylic acid cycle, Pyruvate metabolism and Glyoxylate metabolism), any other kinetic parameters were therefore constrained. The previous results were used to compute mean values of these parameters and their sampling ranges were constrained to $\pm 10\%$ around this mean. We then repeated the previous experiment. Interestingly, the only S_i which increased was that of the PPP subsystem (Figure 72), all others remained negligible. Constraining parameters outside of the study scope worked as expected, and the results also made physiological sense, as both EDA and GND belong to the PPP subsystem. It also serves to highlight that uncertainty in the kinetic parameters outside of the chosen subsystems can greatly influence the results. This method of constraining parameters was therefore used for the rest of the computations, so that noise coming from parameters outside the study scope was reduced.

5.4.3.2 *Increasing the resolution to individual parameters*

After having devised a way to reduce noise originating from outside the study scope we proceeded to pick another FCC within the PPP, and perform GSA first at the resolution of subsystems, then reactions and finally individual kinetic parameters. We picked the FCC of EDA with respect to $V_{\text{max,EDD}}$, both within the PPP.

At the level of the subsystems, we had similar results as previously (Figure 73), with PPP having the highest first order sensitivity index (S_i) of about 0.4. This means that determining the exact values of all kinetic parameters within the PPP would lead to a reduction in the variance of the chosen FCC of 40% on average. Again, this makes physiological sense, as we are exam-

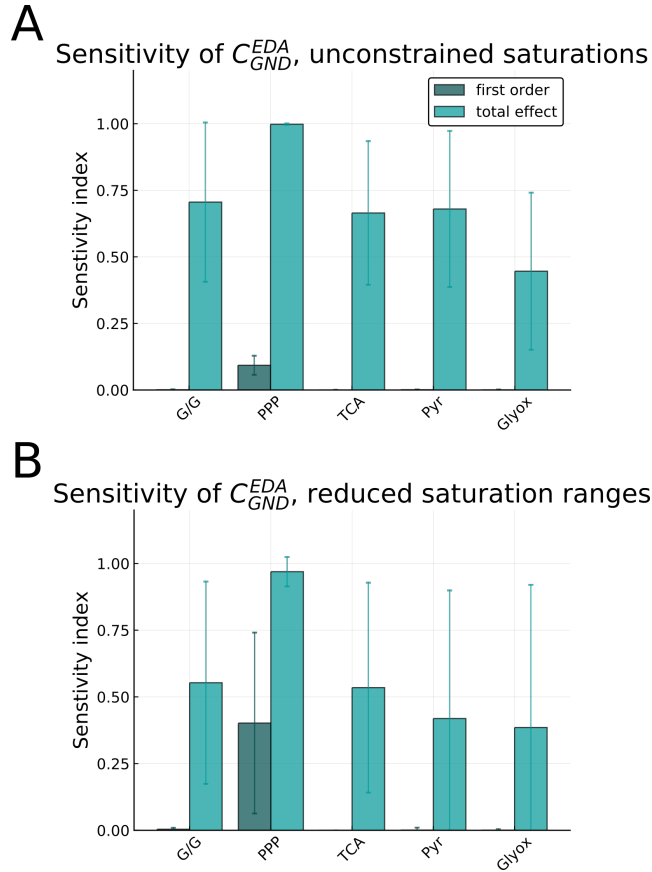


Figure 72: Sobol sensitivity indices for flux control coefficient with respect to subsystem enzyme saturation levels. First order (dark green) and total effect (light green) Sobol sensitivity indices of control coefficient C_{GND}^{EDA} , corresponding to the sensitivity of the flux of reaction EDA with respect to $V_{max,GND}$. All enzyme saturations within a subsystem were grouped when performing sensitivity analysis. The 200'000 samples were split into three groups, from which the mean and standard deviation were calculated. The whiskers indicate the standard deviation and the bars report the mean. (A) Saturations sampled over their entire feasible range. (B) Saturations outside of the 5 studied subsystems, ie. the study scope, are sampled between their mean $\pm 10\%$.

ining the control of a flux within the PPP by a kinetic parameter within the same subsystem. It is also important to note that this gives a quantitative estimate of how much uncertainty originates from outside the PPP. However, there isn't another subsystem with a high Si, meaning that interactions between subsystems plays an important role if further reduction in the FCC uncertainty is desired.

Increasing the resolution of the analysis to the reaction level can make it possible to identify exactly which reactions should be measured within the PPP to most efficiently reduce uncertainty in the model output. This avoids the need of laboriously measuring all kinetic parameters within a subsystem. Once we look at the reactions individually (Figure 73), we can see that the only two reactions with a significant Si are EDA and EDD, the same reactions involved in the FCC we are studying. However, the interactions between reactions are much less important when restricting the study scope to the PPP. The combined Si of EDA and EDD mean that the variance in C_{EDD}^{EDA} could be reduced by 85% on average when determining the values of $V_{max,EDA}$ and $V_{max,EDD}$.

Further increasing the resolution to look at individual kinetic parameters can show us whether it is the affinity of substrate or product that contributes most to uncertainty. If we take the EDD reaction (Figure 73), we can see that out of the two metabolites it is 2-Dehydro-3-deoxy-D-gluconate 6-phosphate (2ddg6p) that has a much higher Si. In the given physiology, 2ddg6p is a product of the EDD reaction. Although we are comparing the uncertainty originating from the affinity of two metabolites to the EDD enzyme, and not EDA, the metabolite closer to the EDA reaction (2ddg6p) has a three-fold higher Si. This means that the saturation of the EDD enzyme by its product, and not its substrate, has a larger effect on how $V_{max,EDD}$ will control the flux through the EDA reaction.

In this demonstration, we applied the GMCA workflow to the C_{EDD}^{EDA} control coefficient. We showed that the uncertainty in C_{EDD}^{EDA} is mostly dependent on the upstream reaction EDD. Although this may seem expected, by performing GMCA we quantified the sources of uncertainty in the system. We showed that the uncertainty contribution of EDD is almost three times more than that of EDA. Within the EDD reaction, the saturation of the EDD enzyme by its product is also roughly three times important than the saturation by its substrate. By applying a methodical approach such as GMCA, we may quantify the different sources of uncertainty, and focus experimental efforts only on the parameters which will improve our understanding of the metabolic system.

5.4.3.3 Application and validation

The results from the previous section suggested that precisely determining the kinetic parameter values of EDA and EDD would reduce the variance

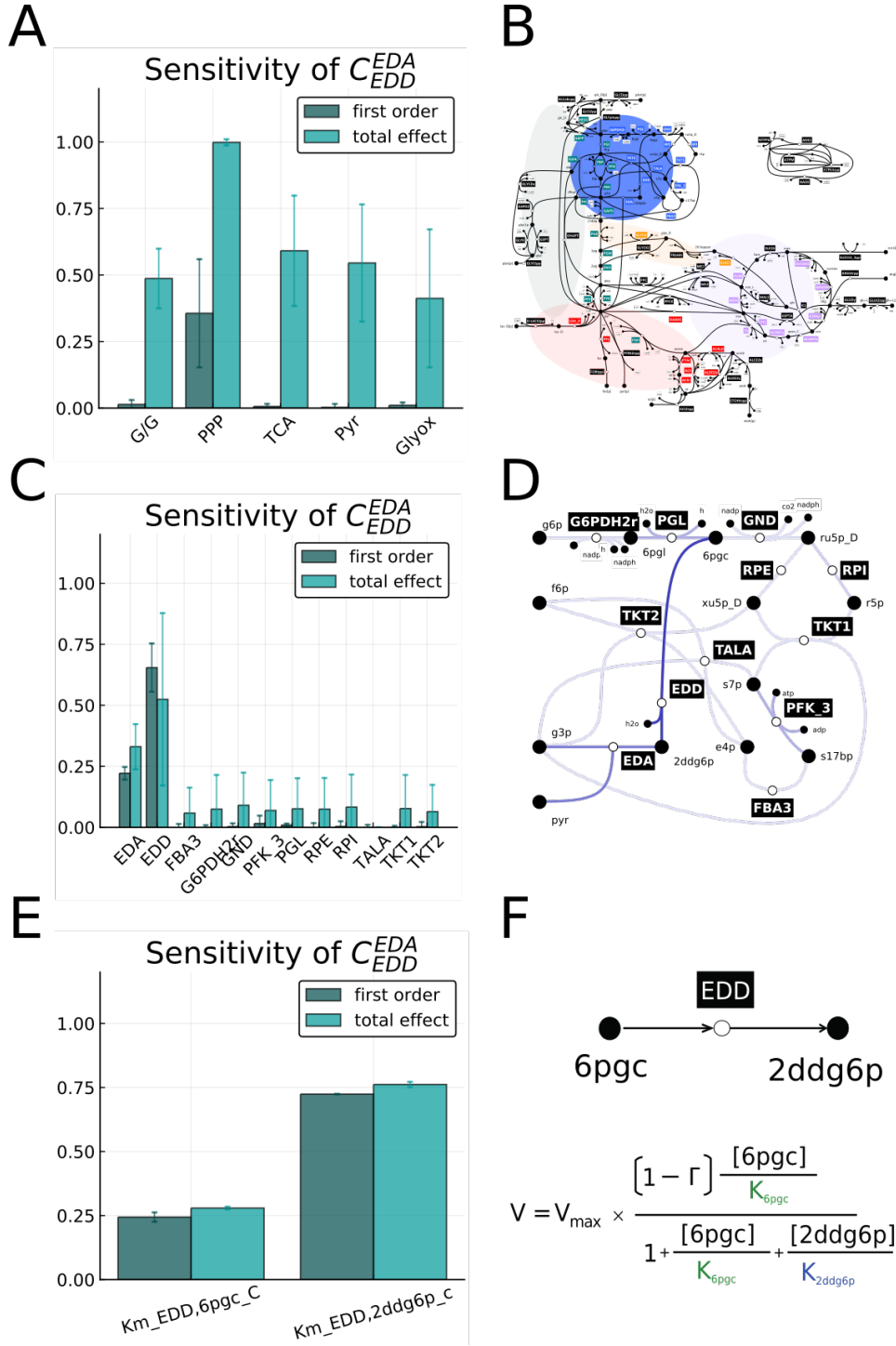


Figure 73: Example of the proposed GMCA workflow, performing GSA at different levels: first subsystems, then reactions and finally individual saturations. First order (dark green) and total effect (light green) Sobol sensitivity indices of control coefficient C_{EDD}^{EDA} , corresponding to the effect of $V_{\max,EDD}$ on the flux of reaction EDA. (A) GSA was first performed by grouping enzyme saturations into subsystems. (B) An illustration of the first order sensitivity indices is superimposed on the network diagram, darker shades indicate higher first order sensitivity indices. (C) GSA was then performed by considering only reactions from the PPP and grouping enzyme saturations according to reaction. (D) First order sensitivity indices are superimposed on an illustration of the PPP. Darker shades indicate higher values. (E) Finally, the contribution of individual kinetic parameters in the EDD reaction were analysed. (F) The EDD reaction is shown, including substrates and its kinetic expression.

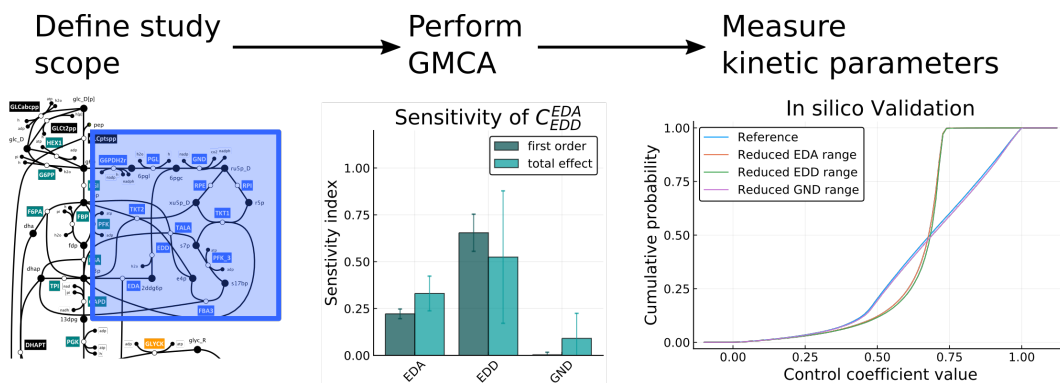


Figure 74: Overview of GMCA in a metabolic engineering context and *in silico* validation of results. An overview of how GMCA fits into a continuous improvement cycle. A study scope is first defined, selecting the control coefficients and sources of uncertainty. These can correspond to a subsystem, a reaction or any desired set of parameters. GMCA is then performed, identifying which sources of uncertainty contribute the most to the uncertainty in the control coefficients. Once identified, these sources of uncertainty can be better estimated, for example by experimental kinetic measurements, literature mining or data mining. This ultimately leads to better kinetic model predictive power, as the uncertainty in the control coefficients has been reduced.

of C_{EDD}^{EDA} by around 85% (Figure 73). The specific Michaelis-Menten parameters for both of these reactions are shown in Table 6. The initial variance of this FCC was 0.27. When we reduced the ranges of the sampled kinetic parameters for EDA and EDD to 10% around their median saturation levels, we reduced the variance of this FCC to 0.07 (Figure 74). This is a reduction in variance of 74%, which is expected based on the Si EDD (Figure 73). However, the reduction in variance of the FCC when reducing the range of kinetic parameters in the EDA reaction was more than expected: 74% versus the expected 25%. This may be due to the fact that Si describes an average reduction in variance, based on fixing the kinetic parameter across multiple points in its distribution. Whereas we fixed the parameter to a range of 10% around a single point, their median. We also attempted reducing the range of the GND kinetic parameters, which had no effect on the variance of the FCC, as predicted (Figure 74). This shows that the Sobol method can effectively identify kinetic parameters that contribute to uncertainty in the model outputs, namely the FCCs.

The FCCs have distributions that are generally slow to converge and have relatively heavy tails. Consequently, future efforts should focus on better estimating appropriate sample sizes and convergence, such as in [160], as well as better sampling methods. Nevertheless, we obtained relatively small errors in the Si coefficients, indicating their correctness (Figure 73, Figure 74).

However, as expected [160], the St coefficients require more samples than Si in order to converge and their error bars are usually larger than for Si. Different metrics for quantifying the convergence of Sobol indices should be considered [160, 161].

5.4.4 *Future opportunities and limitations*

A limitation of GSA is the computational cost, which becomes prohibitive in high dimensions such as those of kinetic models of GEMs. In order to produce the results shown here, approximately two months of computation time was needed on a Dell Precision 7820 with an Intel Xeon Gold 6136 CPU (48 cores, at 3.7GHz). Furthermore, the convergence of statistical moments for the FCCs can take large numbers of samples. Consequently, it takes a significant amount of samples and computational resources to reach convergence of the Sobol indices. Sensitivity analysis and the convergence of underlying sensitivity indices has been studied in pharmacokinetic [162] and environmental [160] modeling leading to similar observations. In certain cases, an environmental model with fifty parameters can require around half a million samples for convergence of Sobol indices [160]. However, with the increasing availability of computational power, variance-based sensitivity analysis approaches are becoming more accessible for large-scale kinetic models. Another area that may alleviate the computational cost is the method of calculating of sensitivity indices. Although Sobol indices are well-established and have their advantages, other non-parameteric sensitivity indices are a promising avenue, such as those based on the Kolmogorov-Smirnov statistic [163]. Alternatively, Kiparissides and coworkers have suggested that Derivative-based Global Sensitivity Measures (DGSM) [137] provide a more efficient method for performing sensitivity analysis. Although these methods may provide advantages compared to GMCA, they have not been compared in depth nor implemented in practice.

A further direction of investigation would be to study the populations of kinetic parameters to understand how they result in the FCC distributions and their convergence. It could be expected from previous research that a subset of kinetic parameters explain this slow convergence [135] [164]. One approach could be to use machine learning algorithms inspired by the iSHRUNK workflow [164], that is based on classification and regression trees (CART), to study which kinetic parameters contribute to the slow convergence of the FCC distributions. Ultimately, understanding what parameters cause the slow convergence of the FCC statistical moments could further help characterize sources of uncertainty. It may be advisable in future studies to perform such analysis prior to carrying out variance-based sensitivity analysis.

5.5 CONCLUSIONS

We hereby introduced a workflow that performs variance-based sensitivity analysis on large-scale kinetic models of metabolism. To our knowledge, this type of study has not been carried out on such high-dimensional kinetic models, as most previous studies were performed on systems with several dozens of parameters. We demonstrated how we can complete a sensitivity analysis focused on control coefficients of the 12 reactions in the PPP with respect to the activities of their enzymes, with the goal of ranking all 3083 V_{\max} parameters present in the *E. coli* model based on their contribution to uncertainty in the FCCs.

Although GSA can be computationally expensive, we developed a workflow that involves progressively increasing the resolution of GSA from subsystems to enzyme saturations, delivering interpretable results while reducing computation time. While doing this, we also highlighted how uncertainty from other subsystems propagates to uncertainty in the control of the PPP. Furthermore, we found that care should be taken when analyzing outputs of kinetic models, in regard to convergence and error estimates. It may be advisable to first perform an *ab initio* study of the kinetic model and its control coefficients, prior to engaging into sensitivity analysis. We suggest that machine learning approaches, such as iSHRUNK [164], could open up new avenues for unraveling information about parameters that contribute to uncertainty. As an alternative, we suggest exploring the suitability of other GSA methods to kinetic models of metabolism [137, 163].

5.6 MATERIALS AND METHODS

5.6.1 MCA, sampling saturations

In order to describe a micro-organism growing in batch culture, mass balances can be used to form a system of ordinary differential equations [151]. These assume a well-mixed system [165], where the time scales of metabolism are much faster than those of the process conditions [151]:

$$\frac{dx}{dt} = \mathbf{N}\mathbf{v}(\mathbf{x}, \mathbf{p}) \quad (62)$$

where \mathbf{N} is the stoichiometric matrix in which each row corresponds to a metabolite, each column to a reaction and the entries are the stoichiometric coefficients, \mathbf{v} is the vector of fluxes, \mathbf{x} is the vector of metabolite concentrations, \mathbf{p} is the vector of parameters describing both enzyme kinetics and system conditions, such as temperature and pH.

Metabolic systems often include conserved moieties [166], such as the NADH - NADH⁺ pair. These manifest in linear dependence between the

relevant rows of the stoichiometric matrix \mathbf{N} . In order to obtain a full-rank matrix, denoted by \mathbf{N}_R , we can split the metabolites into independent, \mathbf{x}_i , and dependent, \mathbf{x}_d , as shown in [141]. A set of parameters representing the total concentration of these conserved moieties, \mathbf{p}_m , is also needed. These moiety parameters relate the concentration of dependent and independent metabolites.

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{N}_R \mathbf{v}(\mathbf{x}_i, \mathbf{x}_d(\mathbf{x}_i, \mathbf{p}_m), \mathbf{p}) \quad (63)$$

In the framework of MCA [167], the concentration control coefficients (\mathbf{C}_p^x) and the flux control coefficients (\mathbf{C}_p^v) are the fractional change of the metabolite concentrations, \mathbf{x} , and metabolite fluxes, \mathbf{v} , respectively, in response to a fractional change in system parameters \mathbf{p} . From the log(linear) formalism shown in [141, 168], if the system is linearized around the steady state, these control coefficients can be expressed like so:

$$\mathbf{C}_p^x = \frac{d \ln \mathbf{x}_i}{d \ln \mathbf{p}} = -(\mathbf{N}_R \mathbf{V} \mathbf{E})^{-1} \mathbf{N} \mathbf{V} \mathbf{\Pi} \quad (64)$$

$$\mathbf{C}_p^v = \frac{d \ln \mathbf{v}}{d \ln \mathbf{p}} = \mathbf{E} \mathbf{C}_p^x + \mathbf{\Pi} \quad (65)$$

$$(66)$$

where, \mathbf{N}_R is the reduced stoichiometric matrix, \mathbf{V} is the diagonal matrix containing the steady-state fluxes, \mathbf{E} is the elasticity matrix with respect to metabolites and $\mathbf{\Pi}$ is the matrix of elasticities with respect to kinetic parameters.

To illustrate how to calculate these elasticity matrices \mathbf{E} and $\mathbf{\Pi}$, let us consider a uni-uni reversible Michaelis-Menten enzymatic reaction $S \leftrightarrow P$, its thermodynamic displacement, Γ_i [169–171], is defined as:

$$\Gamma_i = \frac{1}{K_{eq,i}} \frac{P}{S} \quad (67)$$

where, K_{eq} is the thermodynamic equilibrium constant of the reaction. The rate expression for this reaction would hence be given by:

$$v_i = v_{max,i} \frac{(1 - \Gamma_i) \frac{S}{K_{mS,i}}}{1 + \frac{S}{K_{mS,i}} + \frac{P}{K_{mP,i}}} = v_{max,i} \frac{(1 - \Gamma_i) \tilde{S}_i}{1 + \tilde{S}_i + \tilde{P}_i} \quad (68)$$

where, Γ_i is the thermodynamic displacement of the reaction i and $v_{max,i}$ is its maximum flux. $K_{mS,i}$ and $K_{mP,i}$ correspond to the Michaelis-Menten constants of metabolites S and P , respectively. \tilde{S}_i and \tilde{P}_i are the metabolite

concentrations S and P scaled by their corresponding Michaelis-Menten constants.

The kinetic parameter space is characterized by uniformly sampling the saturation terms of each reaction, in this case there are two such terms \tilde{S}_i and \tilde{P}_i . The saturation, σ , is the fraction of a binding site that is occupied by a substrate and is by definition well bounded $\in [0, 1]$. We define the saturation of the enzyme of reaction i with respect to S as:

$$\sigma = \frac{\frac{S}{K_{mS,i}}}{1 + \frac{S}{K_{mS,i}}} = \frac{\tilde{S}_i}{1 + \tilde{S}_i} \quad (69)$$

We can hence define the scaled concentrations in terms of the sampled saturations as:

$$\tilde{S}_i = \frac{\sigma}{1 - \sigma} \quad (70)$$

As illustrated in the literature [36, 172], the elasticities of the reaction with respect to its metabolites directly depend on the scaled concentrations and are given as:

$$E_{i,S} = \frac{\partial \ln v_i}{\partial \ln S} = \frac{1}{1 - \Gamma_i} - \frac{\tilde{S}_i}{1 + \tilde{S}_i + \tilde{P}_i} \quad (71)$$

$$E_{i,P} = \frac{\partial \ln v_i}{\partial \ln P} = -\frac{\Gamma_i}{1 - \Gamma_i} - \frac{\tilde{P}_i}{1 + \tilde{S}_i + \tilde{P}_i} \quad (72)$$

Sampling saturations facilitates the computation of scaled concentrations that can directly be used to populate the elasticity matrix E required for computing the control coefficients. Although only the case for a uni-uni reversible Michaelis-Menten reaction is shown here, the derivation of the analytical expressions for metabolite elasticity E_i and parameter elasticity Π_i needs to be done for every kinetic mechanism type.

5.6.2 GSA, calculating sensitivity indices

First we construct matrix Σ_A , consisting of n samples and k input parameters. The input parameters in this case are the sampled saturations (σ). We construct matrix Σ_{Bj} by taking column j of matrix Σ_A and we resample all saturations other than σ_j . Inversely, we construct Σ_{Cj} by taking all columns

from Σ_A except the column j , meaning that we only resample column j , corresponding to the saturation σ_j .

$$\Sigma_A = \begin{pmatrix} \sigma a_{1,1} & \cdots & \sigma a_{1,k} \\ \vdots & \ddots & \vdots \\ \sigma a_{n,1} & \cdots & \sigma a_{n,k} \end{pmatrix} \quad (73)$$

$$\Sigma_{Bj} = \begin{pmatrix} \sigma b_{1,1} & \cdots & \sigma a_{1,j} & \cdots & \sigma b_{1,k} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \sigma b_{n,1} & \cdots & \sigma a_{n,j} & \cdots & \sigma b_{n,k} \end{pmatrix} \quad (74)$$

$$\Sigma_{Cj} = \begin{pmatrix} \sigma a_{1,1} & \cdots & \sigma c_{1,j} & \cdots & \sigma a_{1,k} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \sigma a_{n,1} & \cdots & \sigma c_{n,j} & \cdots & \sigma a_{n,k} \end{pmatrix} \quad (75)$$

Sometimes a stable model can not be obtained for a set of parameters (row) from Σ_A . We attempted up to 1000 trials for each sample of Σ_A in order to generate Σ_{Bj} and Σ_{Cj} . If, for one row of Σ_A , 1000 samples didn't produce a stable row in Σ_{Bj} or Σ_{Cj} , then this row was skipped.

We evaluate the models to generate output vectors c_A , c_{Bj} and c_{Cj} , these contain n values of a single control coefficient calculated with different sampled saturations σ . The control coefficient corresponds to one entry of C_p^x or C_p^y , calculated as shown in [Section 5.6.1](#). Vector c_A is of length n but, vectors c_{Bj} and c_{Cj} may be shorter than n if some realizations of Σ_{Bj} or Σ_{Cj} were not stable.

Let c denote the model output we are interested in. In this case c is a particular control coefficient being studied, ie. it is one entry of the C_p^x or C_p^y matrices. Note that c is a random variable in the context of statistics, whereas c_A , c_{Bj} and c_{Cj} are samples of this variable. Sensitivity indices are calculated as follows for a model output c :

$$Si = \frac{V_{\sigma_j}(E_{\sigma_{\sim j}}(c|\sigma_j))}{V(c)} = \frac{c_A c_{Bj}^T / n - f_0^2}{c_A c_A^T / n - f_0^2} \quad (76)$$

$$St = 1 - \frac{V_{\sigma_{\sim j}}(E_{\sigma_j}(c|\sigma_{\sim j}))}{V(c)} = 1 - \frac{c_A c_{Cj}^T / n - f_0^2}{c_A c_A^T / n - f_0^2} \quad (77)$$

where, Si and St are the first order and total effects, respectively, f_0 corresponds to the mean value of c . Si corresponds to the independent contribution of a saturation to the variance of the control coefficient c . St is

the total contributions of a saturation (including its interaction with other saturations) to the variance of the model output c . Hence, by definition:

$$0 \leq S_i \leq S_t \leq 1 \quad (78)$$

When S_i is equal to S_t , we can say that the saturation in question is uniquely responsible for the variance of a control coefficient. This means that the effect of a saturation on the control coefficient is independent of the values of other saturations in the model. When $S_i < S_t$, this means that the effect of a saturation on the control coefficient will depend on the value of one or several other saturations, these are called interactions. For full derivations of the sensitivity indices the reader is referred to Sobol's publication [147].

6 CONCLUSION

Building mechanistic models that accurately capture the kinetics of biological systems and reproduce experimental data isn't a trivial task. There are a multitude of approaches and problem formulations to choose from, all of them potentially leading to the desired goal. The difficulty is only increased by the indirect nature of biological experiments, and the resulting uncertainty in model parameters. However, if the endeavour is successful, one can reap a very high payoff in the form of a detailed mechanistic understanding of the system studied.

In this thesis we have shown the applicability of modelling to protein regulation through palmitoylation. We have shown how to use modelling to facilitate the understanding of varied and complex experimental data. We also showed that it is possible to make predictions, how these are relevant to biological understand, and validated some of these experimentally. As the models built were successively larger, We have shown that this approach works with increasing model size and complexity.

In *Chapter 1* we introduced the mathematical background common to all models treated in this thesis. We discussed Global Sensitivity Analysis (GSA) and how it could be applied to the Calnexin model. We described the internal workings of GSA and how to interpret the resulting Sobol sensitivity indices. We proceeded to apply GSA to the existing Calnexin model in order to demonstrate how it can help better understand complex systems and the relationship between multiple parameters and emergent model behaviour. We showed how the process of model calibration can be facilitated with GSA. In particular, how conflicting parameters can be identified in order to escape local minima and successfully calibrate a model. Incidentally, we also showed that the calibrated parameters of the Calnexin model can be recombined with each other without reducing the quality of calibration. This demonstrates that the parameters actually describe a space within which the model reproduces the experimental data.

In *Chapter 2* we built a model of Cytoskeleton-linking Membrane Protein 63 (CLiMP63) using papers from the literature, expert knowledge and experimental data. This model was successfully calibrated to accurately reproduce radiolabelling and immunofluorescence microscopy data using Covariance Matrix Adaptation Evolution Strategy (CMA-ES). We suggested multiple different model structures that could potentially reproduce the experimental data and attempted to calibrate each one. We found that a higher-order oligomeric structure was necessary in order to faithfully reproduce experimental data. In the model, this was implemented in the form

of a dimer. The existence of this oligomer was subsequently validated experimentally. We then described the biosynthesis and trafficking routes of CLiMP63 and how these are modulated by palmitoylation.

In *Chapter 3* we built a model of CMG₂ and calibrated it using experimental data. We described how the zDHHC enzymes affect the biosynthetic pathway of CMG₂. In particular, how the silencing of both zDHHC₃ and zDHHC₇ cause an additive reduction in the biosynthesis of CMG₂. We showed that zDHHC₇ has a higher contribution to CMG₂ biosynthetic flux, and that silencing either of these enzymes does not have an affect on CMG₂ half-life at the PM. We then simulated the effect of the Anthrax toxin through increased depalmitoylation at the Plasma Membrane (PM). This was done by overexpressing the corresponding APT *in silico*, leading to a large increase in turnover, while the total levels of CMG₂ are maintained roughly constant. In other words, the presence of the anthrax toxin does not greatly reduce the amount of CMG₂ at the surface, but likely due to endocytosis of the toxin-CMG₂ complex, the turnover of CMG₂ is greatly increased.

In *Chapter 4* we proposed a Global Sensitivity Analysis (GSA) approach that aims to refine the results of Metabolic Control Analysis (MCA) on large scale metabolic models. We pointed out the pitfalls and computational cost of such an analysis. Our approach uses parameter grouping to alleviate the computational cost while still producing useful and interpretable results. We used a model of the bacteria *E. coli* to study uncertainty in the Pentose Phosphate Pathway (PPP) and showed how uncertainty quantification has the potential to improve the predictions of MCA and kinetic models.

6.1 FUTURE OUTLOOK

6.1.1 Palmitoylation models

Although the approach we have taken to modelling palmitoylation has worked through increasing model size and complexity, the increase in computational cost has been noticeable. Developing an alternative to this approach would be beneficial to build models of higher complexity. There are two aspects in particular to focus on. The first is software tooling. At a small scale, writing some of the code by hand is not an issue. However, as models get larger this is not only tedious, but highly error-prone. Existing tools in the domain of rule-based modelling [96, 97] are already used, but should be more tightly integrated in order to avoid manual curation steps. The second aspect to focus on is the mathematical formulation. The current approach uses a system of ODEs to represent the protein, from its synthesis, through trafficking and palmitoylation, to its degradation. This is then integrated in order to obtain dynamic responses. However, under certain conditions it is possible to reduce this to a first order a system and therefore solve it analytically instead of performing a costly numerical integration.

As detailed in [53], the enzyme reactions are modelled according to tQSSA kinetics:

$$v = \frac{V_{\max} S_1}{K_{m,1}(1 + E/K_{m,1} + \sum S_i/K_{m,i})} \quad (79)$$

The term $\sum S_i/K_{m,i}$ is also called the competition term, as it takes into account all substrates that can bind to the catalyzing enzyme and their affinity to this enzyme. In all three models of palmitoylation, these competing species include proteins with different labelling states. Because we have made the reasonable assumption that labelling does not affect the reaction kinetics, the binding affinity of these different proteins is the same. In other words, $K_{m,i}$ does not change when a protein is labelled. During the pulse chase, labelled species gradually replace the non-labelled ones, and vice-versa during the chase phase. This means that $\sum S_i$ is constant, and therefore the competition term is actually constant. We can therefore simplify the reaction equation to the following 1st order reaction rate equation:

$$v = \frac{V_{\max}}{K_{m,1}\Lambda} S_1$$

This opens up many new exciting possibilities. The first is the potential computational speedup of not needing to numerically integrate the system of ODEs. These first order equations may be solved analytically given the starting conditions. The second is the possibility of describing the system as a set of eigenvalues and eigenvectors. It may therefore be interesting to extract the dominant eigenvalues from the experimental data first and calibrate the model with these rather than performing a numerical or analytical integration.

In any case, the important observation is that the sum of labelled and unlabelled species is constant during the various labelling experiments. Exploiting this will surely lead to many more interesting avenues. One of the first problems to deal with will be determining the steady state. Previously, both the steady state and dynamic response were determined by the kinetic parameters. However, the dynamic responses measured in the experimental data do not provide any direct information on the steady state. There is also a possibility of there being multiple states consistent with experimental data. Devising a good formulation to tackle this problem will be key to taking these models further.

Some fields which may be of help with this challenges include Optimal Design and Scientific Machine Learning. Optimal design could potentially be used in order to design experiments that extract the maximum information out of the system [173–176]. In this way, it is possible to stimulate the system in such a way that a maximum number of parameters and characteristics may be estimated. Scientific machine learning is a promising new

field that combines the ability of describing non-linear systems while still retaining interpretability of the results [177, 178]. For parts of the palm-toylation which are non-linear as a function of system parameters, like the steady-state, this could be a promising avenue to investigate.

6.1.2 *Uncertainty in Metabolic models*

High dimensional models are a challenge for GSA, as the extensive sampling means the computational cost is high. Although the method of grouping parameters together is one way alleviate this problem, investigating other uncertainty quantification methods could reveal ones better suited to large scale metabolic models. Polynomial chaos expansion [179–185], for example, may be interesting to investigate.

Different sensitivity indices are another aspect that would be worth investigating. Approaches using the Kolmogorov-Smirnov test exist [186] and do not rely on the assumption of normally distributed parameters as Sobol indices do. Whether the choice of sensitivity index may ease the computational cost isn't clear, but a comparison of methods available would be useful to determine the most suitable.

Kinetic models of metabolism are still relatively young, and a lot remains to be explored. The kinetic parameter space is large and not well understood nor characterised. Classification approaches such as Support Vector Machines or Classification Trees would be a good first step in trying to discretise this space into several subspaces which characteristic behaviours.

Part I

APPENDIX

A ENZYME KINETICS

In this chapter we will present some more detail related to the derivation and validity of the various enzyme kinetics discussed in 2.1.3.

A.1 EQUILIBRIUM APPROXIMATION

As previously mentioned in 2.1.3, the equilibrium approximation is made by assuming the enzyme and substrate form an immediate equilibrium. In mathematical terms this means:

$$k_b[E][S] = k_{ub}[ES] \quad (80)$$

From the conservation law of the enzyme, the free enzyme is the total enzyme minus the enzyme-substrate complex:

$$[E] = [E]_T - [ES] \quad (81)$$

We can use this conservation law to replace the free enzyme $[E]$ in the equilibrium approximation:

$$\begin{aligned} k_b([E]_T - [ES])[S] &= k_{ub}[ES] \\ k_b[E]_T[S] - k_b[ES][S] &= k_{ub}[ES] \\ k_{ub}[ES] - k_b[ES][S] &= k_b[E]_T[S] \\ [ES](k_{ub} - k_b[S]) &= k_b[E]_T[S] \\ [ES] &= \frac{k_b[E]_T[S]}{k_{ub} - k_b[S]} \\ [ES] &= \frac{k_b[E]_T[S]}{k_b(\frac{k_{ub}}{k_b} - [S])} \end{aligned} \quad (82)$$

By introducing the dissociation constant $K_d = \frac{k_{ub}}{k_b}$, we can simplify the equation to:

$$[ES] = \frac{[E]_T[S]}{K_d - [S]} \quad (83)$$

Given that the rate of change of $[P]$ is $k_{cat}[ES]$, we can now write:

$$v = \frac{d[P]}{dt} = \frac{V_{max}[S]}{K_d - [S]} \quad (84)$$

Where $V_{\max} = k_{\text{cat}}[E]_T$.

This is the same approximation and derivation originally made by Michaelis and Menten in 1913. The core approximation rests on the enzyme and substrate being in constant equilibrium during the reaction. This only works if $k_{\text{cat}} \ll k_{\text{ub}}$, that is the rate constant of product formation is significantly less than that of enzyme-substrate dissociation. In other words, the substrate and enzyme must reach equilibrium on a much faster time scale than that of product formation.

A.2 QUASI STEADY-STATE APPROXIMATION

The Quasi Steady-State Approximation (QSSA) takes a slightly different approach than the equilibrium approximation, assuming that the concentration of the enzyme-substrate complex $[ES]$ does not change on the time scale of product formation. We can then take the differential equation for ES and set it to zero to obtain the following:

$$\begin{aligned} \frac{d[ES]}{dt} &= k_b[E][S] - k_{\text{ub}}[ES] - k_{\text{cat}}[ES] = 0 \\ k_b[E][S] &= k_{\text{ub}}[ES] + k_{\text{cat}}[ES] \end{aligned} \quad (85)$$

We can notice that this is the same as the equilibrium approximation, but with the additional term $k_{\text{cat}}[ES]$. We can therefore perform the same derivation as before, but with the Michaelis constant $K_M = \frac{k_{\text{ub}} + k_{\text{cat}}}{k_b}$. This leads to the equation:

$$v = \frac{d[P]}{dt} = \frac{V_{\max}[S]}{K_M + [S]} \quad (86)$$

Compared to the equilibrium approximation, QSSA does not require the enzyme and substrate to be in immediate equilibrium. The criteria for validity is that the enzyme concentration $[E]$ be much less than the substrate concentration $[S]$ or the Michaelis constant K_M .

Although both Calnexin and CLiMP63 are abundant proteins [23, 94], no information concerning the relative abundance of their associated enzymes could be found. Furthermore, the relative abundance may change across cell lines or between cellular compartments. No information as to the copies per cell of CMG2 could be found.

A.3 TOTAL QUASI STEADY-STATE APPROXIMATION

The full derivation of the Total Quasi Steady-State Approximation (tQSSA) is relatively involved and beyond the scope of this thesis [54]. However, the

starting assumptions can be given for perspective compared to the previous two approximations.

The tQSSA starts by defining a total substrate and enzyme quantities:

$$\begin{aligned} S_T &= S + ES \\ E_T &= E + ES \end{aligned} \tag{87}$$

Contrary to the QSSA, it is actually shown that a quasi steady-state of the enzyme-substrate complex is reached. For more detail, the reader is referred to the original work [54].

This approximation is valid when

$$\frac{K}{2S_t} \left(\frac{E_t + K_m + S_t}{\sqrt{(E_t + K_m + S_t)^2 - 4E_t S_t}} - 1 \right) \ll 1 \tag{88}$$

In general, this is valid for almost all in vivo conditions [53], both when enzyme concentrations are high or low relative to the substrate concentration.

B CLIMP63 MODEL

B.1 MODEL STRUCTURE

Below is a copy of the rule-based modelling file used to generate the ODEs for the CLIMP63 model. All parameter units related to time are in hours. Other parameters related to concentrations, volumes, surfaces *etc* are in arbitrary units. This is because no experimental data is available in absolute units.

```

1 begin model
  begin parameters

        S35switch      1                                #35S Labelling
        switch (0=off 1=on)
        H3Switch        0                                #3H labelling
        switch (0=off 1=on)
6      ks                13
      kfnp                30
      kfp                 30
      kdim                10
      kbdim               10
11     kcat6              7931
      KM6                  455
      KM6U                 455
      KM6dim               455
      kcatB6               18.5
16     KMb6               39.2
      KMb6U                39.2
      KMb6dim              39.2
      kcat2                59.7
      KM2PM                99.6
21     KM2CP              99.6
      kcatB2               102.9
      KMb2PM               163.7
      KMb2CP               163.7

26     kdu0               1.19
      kdu1                 1.19
      kdC0ER               4.04
      kdC1ER               .03
      kdC0PM               .007
31     kdC1PM             .02

```

```

kdDim2      1
kdDim1      1
kdDim0      1

36  vol_EC      47      # effective compartment volumes
    vol_PM      4.3
    vol_CP      23.9
    vol_ER      1.5
    vol_EX      4.0

41  knpER_CP    1
    knpCP_ER    1
    knpCP_PM    1
    knpPM_CP    1
46  kpER_CP     1
    kpCP_ER     1
    kpCP_PM     1
    kpPM_CP     1

51  kptoEX      1
    kpfromEX    1
    knptoEX     1
    knpfromEX   1

56  kpat2CP     0
    kpat2PM     0
    kapt2PM     0
    kapt2CP     0

61  kdumB       0
    kdumUB      0

end parameters

66  begin compartments
    EC  3  vol_EC      #External environment
    PM  2  vol_PM  EC  #Plasma membrane
    CP  3  vol_CP  PM  #Cytoplasm
    EX  2  vol_EX  CP  #Exosome
71  EN  2  vol_ER  CP  #Endoplasmic reticulum

end compartments

begin molecule types
76  DHHC6(DHHC)
    DHHC2(DHHC)
    Climp63(L~off~on,F~y~n,C100~U~P~PL)

```

```

      APTpm(B,S1~U~P)
      APTer(B,S1~U~P)
81      Dim(L~0~1~2,C~0~1~2,CL~0~1~2)
      DHHC6Dum()
      Synthesis()          # structureless molecule for species
        synthesis
      Degradation()        # structureless molecule for species
        degradation
    end molecule types
86
    begin species
      @EN:DHHC6(DHHC)                0.1526
      @PM:DHHC2(DHHC)                0.1522
91      @CP:DHHC2(DHHC)                0.1522
      @EN:Climp63(L~off,F~n,C100~U)  0
      @EN:Climp63(L~on,F~n,C100~U)   0
      @PM:APTpm(B,S1~P)              0.1523
      @CP:APTpm(B,S1~P)              0.1523
96      @EN:APTer(B,S1~P)              0.1523
      @EN:Synthesis()                1
      @EN:Degradation()              0
      @PM:Degradation()              0
      @EN:Dim(L~0,C~0,CL~0)          0
101      @EN:DHHC6Dum()                0
    end species

    begin observables
      # Enzymes
106      Molecules DHHC6free DHHC6(DHHC)@EN
      Molecules DHHC2PM DHHC2(DHHC)@PM
      Molecules DHHC2CP DHHC2(DHHC)@CP
      Molecules APTerfree APTer(B,S1~P)@EN
      Molecules APTPM APTpm(B,S1~P)@PM
111      Molecules APTCP APTpm(B,S1~P)@CP
      # Climp and dimer
      Molecules ClimpU0ER Climp63(F~n,C100~U)@EN
      Molecules ClimpU1ER Climp63(F~n,C100~P)@EN, Climp63(F~n,C100~PL
        )@EN
      Molecules ClimpC0ER Climp63(F~y,C100~U)@EN
116      Molecules ClimpC1ER Climp63(F~y,C100~P)@EN, Climp63(F~y,C100~PL
        )@EN
      Molecules Dim00 Dim(C~0,CL~0),Dim(C~0,CL~0)
      Molecules Dim01 Dim(C~1,CL~0),Dim(C~1,CL~0), Dim(C~0,CL~1),Dim
        (C~0,CL~1)
      Molecules Dim11 Dim(C~2,CL~0),Dim(C~2,CL~0), Dim(C~0,CL~2),Dim
        (C~0,CL~2), Dim(C~1,CL~1),Dim(C~1,CL~1)

```

```

Molecules ClimpC0CP Climp63(F~y,C100~U)@CP
121 Molecules ClimpC1CP Climp63(F~y,C100~P)@CP, Climp63(F~y,C100~PL
      )@CP
Molecules ClimpC0PM Climp63(F~y,C100~U)@PM
Molecules ClimpC1PM Climp63(F~y,C100~P)@PM, Climp63(F~y,C100~PL
      )@PM
Molecules ClimpC0EX Climp63(F~y,C100~U)@EX
Molecules ClimpC1EX Climp63(F~y,C100~P)@EX, Climp63(F~y,C100~PL
      )@EX
126 #35S labelled species
Molecules ClimpU035S Climp63(L~on,F~n)@EN
Molecules ClimpC1ER35s Climp63(L~on,F~y,C100~P)@EN, Climp63(L~
      on,F~y,C100~PL)@EN
Molecules ClimpDim01_35S Dim(L~1,C~1,CL~0), Dim(L~1,C~0,CL~1),
      Dim(L~2,C~1,CL~0),Dim(L~2,C~1,CL~0), Dim(L~2,C~0,CL~1),
      Dim(L~2,C~0,CL~1)
Molecules ClimpDim11_35S Dim(L~1,C~2,CL~0), Dim(L~1,C~0,CL~2),
      Dim(L~1,C~1,CL~1), Dim(L~2,C~2,CL~0),Dim(L~2,C~2,CL~0),
      Dim(L~2,C~0,CL~2),Dim(L~2,C~0,CL~2), Dim(L~2,C~1,CL~1),Dim(
      L~2,C~1,CL~1)
131 Molecules ClimpC1CP35S Climp63(L~on,F~y,C100~P)@CP, Climp63(L~
      on,F~y,C100~PL)@CP
Molecules ClimpC1PM35s Climp63(L~on,F~y,C100~P)@PM, Climp63(L~
      on,F~y,C100~PL)@PM
Molecules ClimpC1EX35s Climp63(L~on,F~y,C100~P)@EX, Climp63(L~
      on,F~y,C100~PL)@EX
Molecules ClimpC0ER35s Climp63(L~on,F~y,C100~U)@EN
Molecules ClimpDim00_35S Dim(L~1,C~0,CL~0), Dim(L~2,C~0,CL~0),
      Dim(L~2,C~0,CL~0)
136 Molecules ClimpC0CP35S Climp63(L~on,F~y,C100~U)@CP
Molecules ClimpC0PM35s Climp63(L~on,F~y,C100~U)@PM
Molecules ClimpC0EX35s Climp63(L~on,F~y,C100~U)@EX

# 3H labelled species
141 Molecules ClimpC1ER3h Climp63(F~y,C100~PL)@EN
Molecules Dim01_3H Dim(C~0,CL~1), Dim(C~1,CL~1)
Molecules Dim11_3H Dim(C~0,CL~2), Dim(C~0,CL~2)
Molecules ClimpC1CP3h Climp63(F~y,C100~PL)@CP
Molecules ClimpC1PM3h Climp63(F~y,C100~PL)@PM
146 Molecules ClimpC1EX3h Climp63(F~y,C100~PL)@EX
#Totals
Molecules Climp_35Slabelled Climp63(L~on), Dim(L~1), Dim(L~2),
      Dim(L~2)
Molecules Climp_3Hlabelled Climp63(C100~PL), Dim(CL~1), Dim(CL
      ~2), Dim(CL~2) # Dimer counted twice, because two climp
      proteins

```

```

Molecules ClimpDim0 Dim(C~0,CL~0), Dim(C~1,CL~0), Dim(C~0,CL~1)
    # All dimer that can be palmit, used for rxn rates
151 Molecules ClimpDim1 Dim(C~2,CL~0), Dim(C~1,CL~0), Dim(C~0,CL~1)
    , Dim(C~1,CL~1), Dim(C~0,CL~2) # All dimer that can be
    depalmit, used for rxn rates
Molecules ClimpER Climp63()@EN, Dim(), Dim()
Molecules ClimpERFolded Climp63(F~y)@EN, Dim(), Dim()
Molecules Dimtot Dim()
Molecules CLimpCP Climp63()@CP
156 Molecules ClimpPM Climp63()@PM
Molecules ClimpEX Climp63()@EX
Molecules Climptot Climp63(), Dim(), Dim()
Molecules ClimptotFolded Climp63(F~y), Dim(), Dim()

161 end observables

begin reaction rules
####SYNTHESIS AND FOLDING (climp63):
Synthesis() -> Synthesis()+Climp63(L~off,F~n,C100~U)@EN (1-S35switch)*
    ks
166 Climp63(F~n,C100~U)@EN -> Climp63(F~y,C100~U)@EN kfnp
    Climp63(F~n)@EN -> Climp63(F~y)@EN
    kfp exclude_reactants(1,Climp63(C100~U))
####S35 LABELLING REACTIONS
Synthesis() -> Synthesis()+Climp63(L~on,F~n,C100~U)@EN S35switch*ks

171 ####Dimerisation
Climp63(L~off,F~y,C100~U)@EN + Climp63(L~off,F~y,C100~U)@EN <-> Dim(L
    ~0,C~0,CL~0)@EN kdim, kbdim
Climp63(L~off,F~y,C100~P)@EN + Climp63(L~off,F~y,C100~U)@EN <-> Dim(L
    ~0,C~1,CL~0)@EN kdim, kbdim
Climp63(L~off,F~y,C100~P)@EN + Climp63(L~off,F~y,C100~P)@EN <-> Dim(L
    ~0,C~2,CL~0)@EN kdim, kbdim
Climp63(L~off,F~y,C100~P)@EN + Climp63(L~off,F~y,C100~PL)@EN <-> Dim(L
    ~0,C~1,CL~1)@EN kdim, kbdim
176 Climp63(L~off,F~y,C100~PL)@EN + Climp63(L~off,F~y,C100~U)@EN <-> Dim(L
    ~0,C~0,CL~1)@EN kdim, kbdim
Climp63(L~off,F~y,C100~PL)@EN + Climp63(L~off,F~y,C100~PL)@EN <-> Dim(L
    ~0,C~0,CL~2)@EN kdim, kbdim

Climp63(L~on,F~y,C100~U)@EN + Climp63(L~off,F~y,C100~U)@EN <-> Dim(L~1,
    C~0,CL~0)@EN kdim, kbdim
Climp63(L~on,F~y,C100~P)@EN + Climp63(L~off,F~y,C100~U)@EN <-> Dim(L~1,
    C~1,CL~0)@EN kdim, kbdim
181 Climp63(L~on,F~y,C100~P)@EN + Climp63(L~off,F~y,C100~P)@EN <-> Dim(L~1,
    C~2,CL~0)@EN kdim, kbdim

```



```

Climp63(L~on,F~y,C100~P)@EN + Climp63(L~off,F~y,C100~PL)@EN <-> Dim(L
~1,C~1,CL~1)@EN      kdim, kbdim
Climp63(L~on,F~y,C100~PL)@EN + Climp63(L~off,F~y,C100~U)@EN <-> Dim(L
~1,C~0,CL~1)@EN      kdim, kbdim
Climp63(L~on,F~y,C100~PL)@EN + Climp63(L~off,F~y,C100~PL)@EN <-> Dim(L
~1,C~0,CL~2)@EN      kdim, kbdim

186 Climp63(L~on,F~y,C100~U)@EN + Climp63(L~on,F~y,C100~U)@EN <-> Dim(L~2,C
~0,CL~0)@EN      kdim, kbdim
Climp63(L~on,F~y,C100~P)@EN + Climp63(L~on,F~y,C100~U)@EN <-> Dim(L~2,C
~1,CL~0)@EN      kdim, kbdim
Climp63(L~on,F~y,C100~P)@EN + Climp63(L~on,F~y,C100~P)@EN <-> Dim(L~2,C
~2,CL~0)@EN      kdim, kbdim
Climp63(L~on,F~y,C100~P)@EN + Climp63(L~on,F~y,C100~PL)@EN <-> Dim(L~2,
C~1,CL~1)@EN      kdim, kbdim
Climp63(L~on,F~y,C100~PL)@EN + Climp63(L~on,F~y,C100~U)@EN <-> Dim(L~2,
C~0,CL~1)@EN      kdim, kbdim
191 Climp63(L~on,F~y,C100~PL)@EN + Climp63(L~on,F~y,C100~PL)@EN <-> Dim(L
~2,C~0,CL~2)@EN      kdim, kbdim

####TRANSPORT REACTIONS:
Climp63(F~y,C100~U)@EN -> Climp63(F~y,C100~U)@CP
knpER_CP
Climp63(F~y,C100~U)@CP -> Climp63(F~y,C100~U)@EN
knpCP_ER
196 Climp63(F~y,C100~U)@CP -> Climp63(F~y,C100~U)@PM
knpCP_PM
Climp63(F~y,C100~U)@PM -> Climp63(F~y,C100~U)@CP
knpPM_CP
Climp63(F~y,C100)@EN -> Climp63(F~y,C100)@CP      kpER_CP
exclude_reactants(1,Climp63(C100~U)@EN)
Climp63(F~y,C100)@CP -> Climp63(F~y,C100)@EN      kpCP_ER
exclude_reactants(1,Climp63(C100~U)@CP)
Climp63(F~y,C100)@CP -> Climp63(F~y,C100)@PM      kpCP_PM
exclude_reactants(1,Climp63(C100~U)@CP)
201 Climp63(F~y,C100)@PM -> Climp63(F~y,C100)@CP      kpPM_CP
exclude_reactants(1,Climp63(C100~U)@PM)

Climp63(F~y,C100~U)@PM -> Climp63(F~y,C100~U)@EX      knptoEX
Climp63(F~y,C100)@PM -> Climp63(F~y,C100)@EX      kptoEX
exclude_reactants(1,Climp63(C100~U)@PM)
Climp63(F~y,C100~U)@EX -> Climp63(F~y,C100~U)@PM
knpfromEX
206 Climp63(F~y,C100)@EX -> Climp63(F~y,C100)@PM
kpfromEX      exclude_reactants(1,Climp63(C100~U)@EX)

```

```

DHC2()@PM -> DHC2()@CP
                                kpat2CP
DHC2()@CP -> DHC2()@PM
                                kpat2PM
APTpm()@PM -> APTpm()@CP
                                kapt2CP
211 APTpm()@CP -> APTpm()@PM
                                kapt2PM

####PALMITOYLATION:
Climp63(F~n,C100~U)@EN -> Climp63(F~n,C100~P)@EN (1-
    H3Switch)*kcat6*DHC6free/KM6U/(1+ClimpC0ER/KM6+ClimpU0ER/KM6U+
    ClimpDim0/KM6dim+DHC6free/KM6U)
Climp63(F~y,C100~U)@EN -> Climp63(F~y,C100~P)@EN (1-
    H3Switch)*kcat6*DHC6free/KM6/(1+ClimpC0ER/KM6+ClimpU0ER/KM6U+
    ClimpDim0/KM6dim+DHC6free/KM6)
216 Climp63(F~y,C100~U)@PM -> Climp63(F~y,C100~P)@PM (1-
    H3Switch)*kcat2*DHC2PM/(ClimpC0PM+DHC2PM+KM2PM)
Climp63(F~y,C100~U)@CP -> Climp63(F~y,C100~P)@CP (1-
    H3Switch)*kcat2*DHC2CP/(ClimpC0CP+DHC2CP+KM2CP)
Dim(C~0)@EN -> Dim(C~1)@EN
                                (1-H3Switch)*kcat6*DHC6free/KM6dim/(1+
                                ClimpC0ER/KM6+ClimpU0ER/KM6U+ClimpDim0/KM6dim+DHC6free/KM6dim)
                                exclude_reactants(1,Dim(CL~2)@EN)
Dim(C~1,CL~0)@EN -> Dim(C~2,CL~0)@EN
                                (1-H3Switch)*kcat6*DHC6free/KM6dim/(1+ClimpC0ER/KM6+
                                ClimpU0ER/KM6U+ClimpDim0/KM6dim+DHC6free/KM6dim)

221 ####H3 LABELLING REACTIONS
Climp63(F~n,C100~U)@EN -> Climp63(F~n,C100~PL)@EN
    H3Switch*kcat6*DHC6free/KM6U/(1+ClimpC0ER/KM6+ClimpU0ER/KM6U+
    DHC6free/KM6U)
Climp63(F~y,C100~U)@EN -> Climp63(F~y,C100~PL)@EN
    H3Switch*kcat6*DHC6free/KM6/(1+ClimpC0ER/KM6+ClimpU0ER/KM6U+
    DHC6free/KM6)
Climp63(F~y,C100~U)@PM -> Climp63(F~y,C100~PL)@PM
    H3Switch*kcat2*DHC2PM/(ClimpC0PM+DHC2PM+KM2PM)
Climp63(F~y,C100~U)@CP -> Climp63(F~y,C100~PL)@CP
    H3Switch*kcat2*DHC2CP/(ClimpC0CP+DHC2CP+KM2CP)
226 Dim(CL~0)@EN -> Dim(CL~1)@EN
                                H3Switch*kcat6*DHC6free/KM6dim/(1+ClimpC0ER/KM6+
                                ClimpU0ER/KM6U+ClimpDim0/KM6dim+DHC6free/KM6dim)
                                exclude_reactants
                                (1,Dim(C~2)@EN)
Dim(C~0,CL~1)@EN -> Dim(C~0,CL~2)@EN
                                H3Switch*kcat6*DHC6free/KM6dim/(1+ClimpC0ER/KM6+ClimpU0ER/
                                KM6U+ClimpDim0/KM6dim+DHC6free/KM6dim)

```

```

#####DEPALMITOYLATION
Climp63(F~n,C100)@EN -> Climp63(F~n,C100~U)@EN          kcatB6*
    APTERfree/KMb6U/(1+ClimpC1ER/KMb6+ClimpU1ER/KMb6U+ClimpDim1/KMb6dim+
    APTERfree/KMb6U)          exclude_reactants(1,Climp63(C100~U)@EN)
231 Climp63(F~y,C100)@EN -> Climp63(F~y,C100~U)@EN          kcatB6*
    APTERfree/KMb6/(1+ClimpC1ER/KMb6+ClimpU1ER/KMb6U+ClimpDim1/KMb6dim+
    APTERfree/KMb6)          exclude_reactants(1,Climp63(C100~U)@EN
    )
Climp63(F~y,C100)@PM -> Climp63(F~y,C100~U)@PM          kcatB2*
    APTPM/(ClimpC1PM+APTPM+KMb2PM)  exclude_reactants(1,Climp63(C100~U)
    @PM)
Climp63(F~y,C100)@CP -> Climp63(F~y,C100~U)@CP          kcatB2*
    APTCP/(ClimpC1CP+APTCP+KMb2CP)  exclude_reactants(1,Climp63(C100~U)
    @CP)
Dim(C~2)@EN -> Dim(C~1)@EN
    kcatB6*APTERfree/KMb6dim/(1+ClimpC1ER/KMb6+
    ClimpU1ER/KMb6U+ClimpDim1/KMb6dim+APTERfree/KMb6dim)
Dim(C~1)@EN -> Dim(C~0)@EN
    kcatB6*APTERfree/KMb6dim/(1+ClimpC1ER/KMb6+
    ClimpU1ER/KMb6U+ClimpDim1/KMb6dim+APTERfree/KMb6dim)
236 Dim(CL~2)@EN -> Dim(CL~1)@EN
    kcatB6*APTERfree/KMb6dim/(1+ClimpC1ER/KMb6+ClimpU1ER
    /KMb6U+ClimpDim1/KMb6dim+APTERfree/KMb6dim)
Dim(CL~1)@EN -> Dim(CL~0)@EN
    kcatB6*APTERfree/KMb6dim/(1+ClimpC1ER/KMb6+ClimpU1ER
    /KMb6U+ClimpDim1/KMb6dim+APTERfree/KMb6dim)

#####DEGRADATION:
Climp63(F~n,C100~U)@EN -> Degradation()
    kdu0
241 Climp63(F~y,C100~U)@EN -> Degradation()
    kdC0ER
Climp63(F~y,C100~U)@PM -> Degradation()
    kdC0PM
Climp63(F~n,C100)@EN -> Degradation()
    kdu1  exclude_reactants(1,Climp63(C100~U)@EN)
Climp63(F~y,C100)@EN -> Degradation()
    kdC1ER  exclude_reactants(1,Climp63(C100~U)@EN)
Climp63(F~y,C100)@PM -> Degradation()
    kdC1PM  exclude_reactants(1,Climp63(C100~U)@PM)
246 Dim(C~2)@EN -> Degradation()
    kdDim2
Dim(CL~2)@EN -> Degradation()
    kdDim2
Dim(C~1,CL~1)@EN -> Degradation()
    kdDim2

```

```

Dim(C~1,CL~0)@EN -> Degradation()
                        kdDim1
Dim(C~0,CL~1)@EN -> Degradation()
                        kdDim1
251 Dim(C~0,CL~0)@EN -> Degradation()
                        kdDim0

###Dummy Protein
DHHC6(DHHC) <-> DHHC6Dum()      kdumB, kdumUB
256
end reaction rules

end model
generate_network({})
261 writeMfile()
    #writeMexfile()
    visualize({type=>"ruleviz_operation"})
    visualize({type=>"contactmap",suffix=>"optionalstring"})
    visualize({type=>"regulatory",background=>1,suffix=>"a"})
266 #simulate({method=>"ode", t_end=>10000,n_steps=>10000})

```

B.2 PARAMETER VALUES

Parameter name	Median	Standard deviation
ks	-2.0256008564790884	0.02795497206699842
kfnp	-0.22792859062646453	0.022046607247206022
kfp	-10.986950473902606	0.005399593434487685
kdim	3.7402185729924087	0.027681752352939425
kbdim	-4.231842471417499	0.03216878424271212
kcat6	6.025103304137952	0.026434575608411136
KM6	-4.746716337116787	0.0034480714554681233
KM6U	10.198106634092047	0.019444066596300564
KM6dim	0.9785463193841266	0.029990459517971576
kcatB6	-0.05811585963684825	0.013158089700177327
KMb6	-4.32050376160686	0.031677857248313936
KMb6U	10.108344729836057	0.016436027551300504
KMb6dim	0.8928453249088466	0.036185825025475474
kcat2	5.382777118578877	0.02789322782189646
KM2PM	-3.166805955034833	0.027053944411323923

KM2CP	10.833456759015743	0.01877041110658976
kcatB2	-1.2087648630629442	0.03399117682271176
KMb2PM	1.630652902788448	0.03545889699580471
KMb2CP	10.653272022177566	0.034434099530748066
kduo	-1.233330088867322	0.01894330159355836
kdu1	10.176669072016294	0.01851644078584501
kdCoER	-1.205196450359896	0.032804336812479594
kdC1ER	-1.6191482766437688	0.011800917761386505
kdCoPM	4.732382205950643	0.027137996232918057
kdC1PM	-1.7152663105956325	0.009440915906751658
kdDim2	-10.0	0.0
kdDim1	-10.0	0.0
kdDimo	-10.0	0.0
vol_EC	-2.88108253149888	0.034807244528799616
vol_PM	-1.3502992113047823	0.03134359475684689
vol_CP	0.771907515439177	0.03259859661418567
vol_ER	4.50675309468322	0.02729491746969013
vol_EX	1.6646941619765339	0.03454202510335197
knP_ER_CP	3.835998737967254	0.022373417708705975
knP_CP_ER	-1.1715411507942917	0.03374756408899087
knP_CP_PM	2.3469793738019913	0.02823334979919353
knP_PM_CP	1.5340031275777932	0.031693107472298365
kp_ER_CP	-1.1432433478405555	0.0159001012748975
kp_CP_ER	-1.0363486797559813	0.031077187208813493
kp_CP_PM	0.6644279120381184	0.035455470147886255
kp_PM_CP	-5.084879634217634	0.03292930021537078
kptoEX	-10.031049524497575	0.007235770994327002
kpfromEX	-10.993893136952806	0.0036536852038298565
knptoEX	-10.68548595092734	0.025130723423288284
knPfromEX	-10.85434070905005	0.02057406277453447
kpat2CP	-10.649071645554168	0.028116339721010446
kpat2PM	-10.121270199652798	0.015345187001212837
kapt2PM	-10.92831118947554	0.01389949731321634
kapt2CP	-10.21524882052428	0.02013574014720274
kdumB	0.8091328932319288	0.036037335425859116

kdumUB	3.0450585395082506	0.03324393531742347
PAT6	-3.163809543072885	0.028221967916535362
PAT2	-0.9805305538459812	0.028499975665107667
PAT2 CP	-10.346284174569686	0.028321940408701674
APT PM	-2.1473667508428544	0.03130224384066048
APT CP	-10.710104269479492	0.02504198650478478
APT ER	0.040325538638002104	0.03090132228661049
Si PAT6	3.959806373964418	0.027091604966799507
Si PAT2	2.391326207728498	0.031367135353299914
Ov PAT6	1.65	1.7808144545380086e-15

Table 7

C CMG₂ MODEL

C.1 MODEL STRUCTURE

Below is a copy of the rule-based modelling file used to generate the ODEs for the CMG₂ model. All parameter units related to time are in hours. Other parameters related to concentrations, volumes, surfaces *etc* are in arbitrary units. This is because no experimental data is available in absolute units.

```

begin model
begin parameters

4      S35switch      1                                #35S Labelling
      switch (0=off 1=on)
      H3switch        0                                #3H labelling
      switch (0=off 1=on)
      ks               13
      kf               5
      # Transport
9      k_go            1
      k_pm_0           1
      k_pm_1           1
      k_pm_2           1
      k_pm_3           1
14     k_to_end        1
      k_from_end       1

      # Palmitoylation
      kcat7            1
19     KM7_0_1         1
      KM7_0_2          1
      KM7_0_3          1
      KM7_1_2          1
      KM7_1_3          1
24     KM7_2_1         1
      KM7_2_3          1
      KM7_3_1          1
      KM7_3_2          1
      KM7_12           1
29     KM7_13          1
      KM7_23           1
      kcat3            1
      KM3_0_1          1

```

	KM3_0_2	1
34	KM3_0_3	1
	KM3_1_2	1
	KM3_1_3	1
	KM3_2_1	1
	KM3_2_3	1
39	KM3_3_1	1
	KM3_3_2	1
	KM3_12	1
	KM3_13	1
	KM3_23	1
44	# Depalmitoylation	
	kcat_APTer	1
	KM_APTer_1	1
	KM_APTer_2	1
49	KM_APTer_3	1
	KM_APTer_12_1	1
	KM_APTer_12_2	1
	KM_APTer_13_1	1
	KM_APTer_13_3	1
54	KM_APTer_23_2	1
	KM_APTer_23_3	1
	KM_APTer_123_1	1
	KM_APTer_123_2	1
	KM_APTer_123_3	1
59	kcat_APTgo	1
	KM_APTgo_1	1
	KM_APTgo_2	1
	KM_APTgo_3	1
64	KM_APTgo_12_1	1
	KM_APTgo_12_2	1
	KM_APTgo_13_1	1
	KM_APTgo_13_3	1
	KM_APTgo_23_2	1
69	KM_APTgo_23_3	1
	KM_APTgo_123_1	1
	KM_APTgo_123_2	1
	KM_APTgo_123_3	1
74	kcat_APTpm	1
	KM_APTpm_1	1
	KM_APTpm_2	1
	KM_APTpm_3	1
	KM_APTpm_12_1	1
79	KM_APTpm_12_2	1

	KM_APTpm_13_1	1
	KM_APTpm_13_3	1
	KM_APTpm_23_2	1
	KM_APTpm_23_3	1
84	KM_APTpm_123_1	1
	KM_APTpm_123_2	1
	KM_APTpm_123_3	1
	# Enzyme competition terms	
89	# PAT7_compet	1
	# PAT3_compet	1
	# APTer_compet	1
	# APTgo_compet	1
94	# Degradation	
	kd_nf_0_er	1
	kd_nf_1_er	1
	kd_nf_2_er	1
	kd_nf_3_er	1
99	kd_nf_12_er	1
	kd_nf_13_er	1
	kd_nf_23_er	1
	kd_nf_123_er	1
104	kd_f_0_er	1
	kd_f_1_er	1
	kd_f_2_er	1
	kd_f_3_er	1
	kd_f_12_er	1
109	kd_f_13_er	1
	kd_f_23_er	1
	kd_f_123_er	1
	# kd_nf_0_go	1
114	# kd_nf_1_go	1
	# kd_nf_2_go	1
	# kd_nf_3_go	1
	# kd_nf_12_go	1
	# kd_nf_13_go	1
119	# kd_nf_23_go	1
	# kd_nf_123_go	1
	kd_f_0_go	1
	kd_f_1_go	1
124	kd_f_2_go	1
	kd_f_3_go	1
	kd_f_12_go	1

```

kd_f_13_go      1
kd_f_23_go      1
129 kd_f_123_go   1

# kd_nf_0_pm     1
# kd_nf_1_pm     1
# kd_nf_2_pm     1
134 # kd_nf_3_pm     1
# kd_nf_12_pm    1
# kd_nf_13_pm    1
# kd_nf_23_pm    1
# kd_nf_123_pm   1

139 kd_f_0_pm      1
kd_f_1_pm      1
kd_f_2_pm      1
kd_f_3_pm      1
144 kd_f_12_pm     1
kd_f_13_pm     1
kd_f_23_pm     1
kd_f_123_pm    1
# Compartment volumes and surfaces:
149 vol_EC      1
A_PM           0.1
vol_CP         0.7
A_END          0.01
A_ER           1
154 A_GO         0.5
# Parameters relating to experimental conditions, not used when
# integrating the system of ODEs, only when modifying
# parameters and
# initial conditions as a function of experimental conditions (
# mutations
# etc.)
159 #KR_degrade   1
#R3             1
#R7             1
#APTer         1
#APTgo         1
164 #APTpm       1
#SiR3          1
#SiR7          1
end parameters

169 begin compartments
EC 3  vol_EC      # External environment
PM 2  A_PM  EC    # Plasma membrane

```

```

        CP 3    vol_CP  PM      # Cytoplasm
        END 2   A_END   CP      # Endosome
174      EN 2    A_ER    CP      # Endoplasmic reticulum
        G0     2       A_G0    CP      # Golgi apparatus
    end compartments

    begin molecule types
179      PAT7()
        PAT3()
        CMG2(L~off~on,F~y~n,C1~U~P~PL,C2~U~P~PL,C3~U~P~PL)
        APTpm()
        APTgo()
184      APTer()
        Synthesis()          # structureless molecule for species
            synthesis
        Degradation()        # structureless molecule for species
            degradation
    end molecule types

189  begin species
        @EN: PAT7()          0.1526
        @G0: PAT3()          0.1522
        @EN: APTer()         0.1523
        @G0: APTgo()         0.1523
194      @PM: APTpm()         0.1523
        @EN: CMG2(L~off,F~n,C1~U,C2~U,C3~U) 0
        @EN: Synthesis()     1
    end species

199  begin observables
        # Enzymes
        Molecules      PAT7      PAT7()@EN
        Molecules      PAT3      PAT3()@G0
        Molecules      APTer     APTer()@EN
204      Molecules      APTgo     APTgo()@G0
        Molecules      APTpm     APTpm()@PM
        # Palmit states
        Species        CMG2_0_er   CMG2(C1~U,C2~U,C3~U)@EN
        Species        CMG2_1_er   CMG2(C1~P,C2~U,C3~U)@EN, CMG2(
            C1~PL,C2~U,C3~U)@EN
209      Species        CMG2_2_er   CMG2(C1~U,C2~P,C3~U)@EN, CMG2(
            C1~U,C2~PL,C3~U)@EN
        Species        CMG2_3_er   CMG2(C1~U,C2~U,C3~P)@EN, CMG2(
            C1~U,C2~U,C3~PL)@EN
        Species        CMG2_12_er  CMG2(C1~P,C2~P,C3~U)@EN, CMG2(
            C1~P,C2~PL,C3~U)@EN, CMG2(C1~PL,C2~P,C3~U)@EN, CMG2(C1~PL,C2
            ~PL,C3~U)@EN

```

214	Species	CMG2_13_er	CMG2(C1~P,C2~U,C3~P)@EN, CMG2(C1~P,C2~U,C3~PL)@EN, CMG2(C1~PL,C2~U,C3~P)@EN, CMG2(C1~PL,C2~U,C3~PL)@EN
	Species	CMG2_23_er	CMG2(C1~U,C2~P,C3~P)@EN, CMG2(C1~U,C2~P,C3~PL)@EN, CMG2(C1~U,C2~PL,C3~P)@EN, CMG2(C1~U,C2~PL,C3~PL)@EN
	Species	CMG2_123_er	CMG2(C1~P,C2~P,C3~P)@EN, CMG2(C1~PL,C2~P,C3~P)@EN, CMG2(C1~P,C2~PL,C3~P)@EN, CMG2(C1~P,C2~P,C3~PL)@EN, CMG2(C1~PL,C2~PL,C3~P)@EN, CMG2(C1~PL,C2~P,C3~PL)@EN, CMG2(C1~P,C2~PL,C3~PL)@EN, CMG2(C1~PL,C2~PL,C3~PL)@EN
219	Species	CMG2_0_go	CMG2(C1~U,C2~U,C3~U)@G0
	Species	CMG2_1_go	CMG2(C1~P,C2~U,C3~U)@G0, CMG2(C1~PL,C2~U,C3~U)@G0
	Species	CMG2_2_go	CMG2(C1~U,C2~P,C3~U)@G0, CMG2(C1~U,C2~PL,C3~U)@G0
224	Species	CMG2_3_go	CMG2(C1~U,C2~U,C3~P)@G0, CMG2(C1~U,C2~U,C3~PL)@G0
	Species	CMG2_12_go	CMG2(C1~P,C2~P,C3~U)@G0, CMG2(C1~P,C2~PL,C3~U)@G0, CMG2(C1~PL,C2~P,C3~U)@G0, CMG2(C1~PL,C2~PL,C3~U)@G0
	Species	CMG2_13_go	CMG2(C1~P,C2~U,C3~P)@G0, CMG2(C1~P,C2~U,C3~PL)@G0, CMG2(C1~PL,C2~U,C3~P)@G0, CMG2(C1~PL,C2~U,C3~PL)@G0
229	Species	CMG2_23_go	CMG2(C1~U,C2~P,C3~P)@G0, CMG2(C1~U,C2~P,C3~PL)@G0, CMG2(C1~U,C2~PL,C3~P)@G0, CMG2(C1~U,C2~PL,C3~PL)@G0
	Species	CMG2_123_go	CMG2(C1~P,C2~P,C3~P)@G0, CMG2(C1~PL,C2~P,C3~P)@G0, CMG2(C1~P,C2~PL,C3~P)@G0, CMG2(C1~P,C2~P,C3~PL)@G0, CMG2(C1~PL,C2~PL,C3~P)@G0, CMG2(C1~PL,C2~P,C3~PL)@G0, CMG2(C1~P,C2~PL,C3~PL)@G0, CMG2(C1~PL,C2~PL,C3~PL)@G0
	Species	CMG2_0_pm	CMG2(C1~U,C2~U,C3~U)@PM
229	Species	CMG2_1_pm	CMG2(C1~P,C2~U,C3~U)@PM, CMG2(C1~PL,C2~U,C3~U)@PM
	Species	CMG2_2_pm	CMG2(C1~U,C2~P,C3~U)@PM, CMG2(C1~U,C2~PL,C3~U)@PM
	Species	CMG2_3_pm	CMG2(C1~U,C2~U,C3~P)@PM, CMG2(C1~U,C2~U,C3~PL)@PM
229	Species	CMG2_12_pm	CMG2(C1~P,C2~P,C3~U)@PM, CMG2(C1~P,C2~PL,C3~U)@PM, CMG2(C1~PL,C2~P,C3~U)@PM, CMG2(C1~PL,C2~PL,C3~U)@PM
	Species	CMG2_13_pm	CMG2(C1~P,C2~U,C3~P)@PM, CMG2(C1~P,C2~U,C3~PL)@PM, CMG2(C1~PL,C2~U,C3~P)@PM, CMG2(C1~PL,C2~U,C3~PL)@PM

```

Species      CMG2_23_pm      CMG2(C1~U,C2~P,C3~P)@PM, CMG2(
C1~U,C2~P,C3~PL)@PM, CMG2(C1~U,C2~PL,C3~P)@PM, CMG2(C1~U,C2~
PL,C3~PL)@PM
Species      CMG2_123_pm     CMG2(C1~P,C2~P,C3~P)@PM, CMG2(
C1~PL,C2~P,C3~P)@PM, CMG2(C1~P,C2~PL,C3~P)@PM, CMG2(C1~P,C2~
P,C3~PL)@PM, CMG2(C1~PL,C2~PL,C3~P)@PM, CMG2(C1~PL,C2~P,C3~
PL)@PM, CMG2(C1~P,C2~PL,C3~PL)@PM, CMG2(C1~PL,C2~PL,C3~PL)
@PM

234 Species      CMG2_0_end      CMG2(C1~U,C2~U,C3~U)@END
Species      CMG2_1_end      CMG2(C1~P,C2~U,C3~U)@END, CMG2(
C1~PL,C2~U,C3~U)@END
Species      CMG2_2_end      CMG2(C1~U,C2~P,C3~U)@END, CMG2(
C1~U,C2~PL,C3~U)@END
Species      CMG2_3_end      CMG2(C1~U,C2~U,C3~P)@END, CMG2(
C1~U,C2~U,C3~PL)@END
Species      CMG2_12_end     CMG2(C1~P,C2~P,C3~U)@END, CMG2(
C1~P,C2~PL,C3~U)@END, CMG2(C1~PL,C2~P,C3~U)@END, CMG2(C1~PL,
C2~PL,C3~U)@END
239 Species      CMG2_13_end     CMG2(C1~P,C2~U,C3~P)@END, CMG2(
C1~P,C2~U,C3~PL)@END, CMG2(C1~PL,C2~U,C3~P)@END, CMG2(C1~PL,
C2~U,C3~PL)@END
Species      CMG2_23_end     CMG2(C1~U,C2~P,C3~P)@END, CMG2(
C1~U,C2~P,C3~PL)@END, CMG2(C1~U,C2~PL,C3~P)@END, CMG2(C1~U,
C2~PL,C3~PL)@END
Species      CMG2_123_end    CMG2(C1~P,C2~P,C3~P)@END, CMG2(
C1~PL,C2~P,C3~P)@END, CMG2(C1~P,C2~PL,C3~P)@END, CMG2(C1~P,
C2~P,C3~PL)@END, CMG2(C1~PL,C2~PL,C3~P)@END, CMG2(C1~PL,C2~P,
C3~PL)@END, CMG2(C1~P,C2~PL,C3~PL)@END, CMG2(C1~PL,C2~PL,C3
~PL)@END
# Total species
Molecules    CMG2_er CMG2()@EN
244 Molecules    CMG2_go CMG2()@G0
Molecules    CMG2_pm CMG2()@PM
Molecules    CMG2_end      CMG2()@END
# Non-folded species
Molecules    CMG2_nf_er      CMG2(F~n)@EN
249 Molecules    CMG2_nf_go      CMG2(F~n)@G0
Molecules    CMG2_nf_pm      CMG2(F~n)@PM
Molecules    CMG2_nf_end     CMG2(F~n)@END
# 35S Labelled Species
Molecules    CMG2_35S      CMG2(L~on)
254 Molecules    CMG2_er_35S     CMG2(L~on)@EN
Molecules    CMG2_go_35S     CMG2(L~on)@G0
Molecules    CMG2_pm_35S     CMG2(L~on)@PM
Molecules    CMG2_end_35S   CMG2(L~on)@END
# 3H Labelled Species

```

```

259 Species      CMG2_1_3H      CMG2(C1~PL,C2~U,C3~U)
Species      CMG2_2_3H      CMG2(C1~U,C2~PL,C3~U)
Species      CMG2_3_3H      CMG2(C1~U,C2~U,C3~PL)
Species      CMG2_12_3H     CMG2(C1~PL,C2~PL,C3~U), CMG2(C1
~PL,C2~PL,C3~U), CMG2(C1~PL,C2~P,C3~U), CMG2(C1~P,C2~PL,C3~U
)
Species      CMG2_13_3H     CMG2(C1~PL,C2~U,C3~PL), CMG2(C1
~PL,C2~U,C3~PL), CMG2(C1~PL,C2~U,C3~P), CMG2(C1~P,C2~U,C3~PL
)
264 Species      CMG2_23_3H     CMG2(C1~U,C2~PL,C3~PL), CMG2(C1
~U,C2~PL,C3~PL), CMG2(C1~U,C2~PL,C3~P), CMG2(C1~U,C2~P,C3~PL
)
Species      CMG2_123_3H    CMG2(C1~PL,C2~PL,C3~PL), CMG2(
C1~PL,C2~PL,C3~PL), CMG2(C1~PL,C2~PL,C3~PL)
# Used for plotting surface half-life
Molecules    CMG2_pm_no35S   CMG2(L~off)@PM
# Used for plotting transport fluxes
269 Molecules    CMG2_er_folded CMG2(F~y)@EN
Species      CMG2_go_1c     CMG2(C1~P)@G0, CMG2(C1~PL)@G0
Species      CMG2_go_2c     CMG2(C2~P)@G0, CMG2(C2~PL)@G0
Species      CMG2_go_3c     CMG2(C3~P)@G0, CMG2(C3~PL)@G0

274 ## 35S labelled species
Species      CMG2_0_er_35S   CMG2(L~on,C1~U,C2~U,C3~U)@EN
Species      CMG2_1_er_35S   CMG2(L~on,C1~P,C2~U,C3~U)@EN,
CMG2(L~on,C1~PL,C2~U,C3~U)@EN
Species      CMG2_2_er_35S   CMG2(L~on,C1~U,C2~P,C3~U)@EN,
CMG2(L~on,C1~U,C2~PL,C3~U)@EN
Species      CMG2_3_er_35S   CMG2(L~on,C1~U,C2~U,C3~P)@EN,
CMG2(L~on,C1~U,C2~U,C3~PL)@EN
279 Species      CMG2_12_er_35S CMG2(L~on,C1~P,C2~P,C3~U)@EN,
CMG2(L~on,C1~P,C2~PL,C3~U)@EN, CMG2(L~on,C1~PL,C2~P,C3~U)@EN
, CMG2(L~on,C1~PL,C2~PL,C3~U)@EN
Species      CMG2_13_er_35S CMG2(L~on,C1~P,C2~U,C3~P)@EN,
CMG2(L~on,C1~P,C2~U,C3~PL)@EN, CMG2(L~on,C1~PL,C2~U,C3~P)@EN
, CMG2(L~on,C1~PL,C2~U,C3~PL)@EN
Species      CMG2_23_er_35S CMG2(L~on,C1~U,C2~P,C3~P)@EN,
CMG2(L~on,C1~U,C2~P,C3~PL)@EN, CMG2(L~on,C1~U,C2~PL,C3~P)@EN
, CMG2(L~on,C1~U,C2~PL,C3~PL)@EN
Species      CMG2_123_er_35S CMG2(L~on,C1~P,C2~P,C3~P)@EN,
CMG2(L~on,C1~PL,C2~P,C3~P)@EN, CMG2(L~on,C1~P,C2~PL,C3~P)@EN
, CMG2(L~on,C1~P,C2~P,C3~PL)@EN, CMG2(L~on,C1~PL,C2~PL,C3~P)
@EN, CMG2(L~on,C1~PL,C2~P,C3~PL)@EN, CMG2(L~on,C1~P,C2~PL,C3
~PL)@EN, CMG2(L~on,C1~PL,C2~PL,C3~PL)@EN
284 Species      CMG2_0_go_35S   CMG2(L~on,C1~U,C2~U,C3~U)@G0

```

Species CMG2_1_go_35S CMG2(L~on,C1~P,C2~U,C3~U)@G0,
CMG2(L~on,C1~PL,C2~U,C3~U)@G0

Species CMG2_2_go_35S CMG2(L~on,C1~U,C2~P,C3~U)@G0,
CMG2(L~on,C1~U,C2~PL,C3~U)@G0

Species CMG2_3_go_35S CMG2(L~on,C1~U,C2~U,C3~P)@G0,
CMG2(L~on,C1~U,C2~U,C3~PL)@G0

Species CMG2_12_go_35S CMG2(L~on,C1~P,C2~P,C3~U)@G0,
CMG2(L~on,C1~P,C2~PL,C3~U)@G0, CMG2(L~on,C1~PL,C2~P,C3~U)@G0
, CMG2(L~on,C1~PL,C2~PL,C3~U)@G0

289 Species CMG2_13_go_35S CMG2(L~on,C1~P,C2~U,C3~P)@G0,
CMG2(L~on,C1~P,C2~U,C3~PL)@G0, CMG2(L~on,C1~PL,C2~U,C3~P)@G0
, CMG2(L~on,C1~PL,C2~U,C3~PL)@G0

Species CMG2_23_go_35S CMG2(L~on,C1~U,C2~P,C3~P)@G0,
CMG2(L~on,C1~U,C2~P,C3~PL)@G0, CMG2(L~on,C1~U,C2~PL,C3~P)@G0
, CMG2(L~on,C1~U,C2~PL,C3~PL)@G0

Species CMG2_123_go_35S CMG2(L~on,C1~P,C2~P,C3~P)@G0,
CMG2(L~on,C1~PL,C2~P,C3~P)@G0, CMG2(L~on,C1~P,C2~PL,C3~P)@G0
, CMG2(L~on,C1~P,C2~P,C3~PL)@G0, CMG2(L~on,C1~PL,C2~PL,C3~P)
@G0, CMG2(L~on,C1~PL,C2~P,C3~PL)@G0, CMG2(L~on,C1~P,C2~PL,C3
~PL)@G0, CMG2(L~on,C1~PL,C2~PL,C3~PL)@G0

Species CMG2_0_pm_35S CMG2(L~on,C1~U,C2~U,C3~U)@PM

294 Species CMG2_1_pm_35S CMG2(L~on,C1~P,C2~U,C3~U)@PM,
CMG2(L~on,C1~PL,C2~U,C3~U)@PM

Species CMG2_2_pm_35S CMG2(L~on,C1~U,C2~P,C3~U)@PM,
CMG2(L~on,C1~U,C2~PL,C3~U)@PM

Species CMG2_3_pm_35S CMG2(L~on,C1~U,C2~U,C3~P)@PM,
CMG2(L~on,C1~U,C2~U,C3~PL)@PM

Species CMG2_12_pm_35S CMG2(L~on,C1~P,C2~P,C3~U)@PM,
CMG2(L~on,C1~P,C2~PL,C3~U)@PM, CMG2(L~on,C1~PL,C2~P,C3~U)@PM
, CMG2(L~on,C1~PL,C2~PL,C3~U)@PM

Species CMG2_13_pm_35S CMG2(L~on,C1~P,C2~U,C3~P)@PM,
CMG2(L~on,C1~P,C2~U,C3~PL)@PM, CMG2(L~on,C1~PL,C2~U,C3~P)@PM
, CMG2(L~on,C1~PL,C2~U,C3~PL)@PM

299 Species CMG2_23_pm_35S CMG2(L~on,C1~U,C2~P,C3~P)@PM,
CMG2(L~on,C1~U,C2~P,C3~PL)@PM, CMG2(L~on,C1~U,C2~PL,C3~P)@PM
, CMG2(L~on,C1~U,C2~PL,C3~PL)@PM

Species CMG2_123_pm_35S CMG2(L~on,C1~P,C2~P,C3~P)@PM,
CMG2(L~on,C1~PL,C2~P,C3~P)@PM, CMG2(L~on,C1~P,C2~PL,C3~P)@PM
, CMG2(L~on,C1~P,C2~P,C3~PL)@PM, CMG2(L~on,C1~PL,C2~PL,C3~P)
@PM, CMG2(L~on,C1~PL,C2~P,C3~PL)@PM, CMG2(L~on,C1~P,C2~PL,C3
~PL)@PM, CMG2(L~on,C1~PL,C2~PL,C3~PL)@PM

Species CMG2_0_end_35S CMG2(L~on,C1~U,C2~U,C3~U)@END

Species CMG2_1_end_35S CMG2(L~on,C1~P,C2~U,C3~U)@END,
CMG2(L~on,C1~PL,C2~U,C3~U)@END

```

304 Species      CMG2_2_end_35S  CMG2(L~on,C1~U,C2~P,C3~U)@END,
      CMG2(L~on,C1~U,C2~PL,C3~U)@END
Species      CMG2_3_end_35S  CMG2(L~on,C1~U,C2~U,C3~P)@END,
      CMG2(L~on,C1~U,C2~U,C3~PL)@END
Species      CMG2_12_end_35S CMG2(L~on,C1~P,C2~P,C3~U)@END,
      CMG2(L~on,C1~P,C2~PL,C3~U)@END, CMG2(L~on,C1~PL,C2~P,C3~U)
      @END, CMG2(L~on,C1~PL,C2~PL,C3~U)@END
Species      CMG2_13_end_35S CMG2(L~on,C1~P,C2~U,C3~P)@END,
      CMG2(L~on,C1~P,C2~U,C3~PL)@END, CMG2(L~on,C1~PL,C2~U,C3~P)
      @END, CMG2(L~on,C1~PL,C2~U,C3~PL)@END
Species      CMG2_23_end_35S CMG2(L~on,C1~U,C2~P,C3~P)@END,
      CMG2(L~on,C1~U,C2~P,C3~PL)@END, CMG2(L~on,C1~U,C2~PL,C3~P)
      @END, CMG2(L~on,C1~U,C2~PL,C3~PL)@END
309 Species      CMG2_123_end_35S      CMG2(L~on,C1~P,C2~P,C3~
      P)@END, CMG2(L~on,C1~PL,C2~P,C3~P)@END, CMG2(L~on,C1~P,C2~PL
      ,C3~P)@END, CMG2(L~on,C1~P,C2~P,C3~PL)@END, CMG2(L~on,C1~PL,
      C2~PL,C3~P)@END, CMG2(L~on,C1~PL,C2~P,C3~PL)@END, CMG2(L~on,
      C1~P,C2~PL,C3~PL)@END, CMG2(L~on,C1~PL,C2~PL,C3~PL)@END

# 3H labelled species
Species      CMG2_1_er_3H      CMG2(C1~PL,C2~U,C3~U)@EN
Species      CMG2_2_er_3H      CMG2(C1~U,C2~PL,C3~U)@EN
314 Species      CMG2_3_er_3H      CMG2(C1~U,C2~U,C3~PL)@EN
Species      CMG2_12_er_3H      CMG2(C1~PL,C2~PL,C3~U)@EN, CMG2
      (C1~PL,C2~PL,C3~U)@EN, CMG2(C1~PL,C2~P,C3~U)@EN, CMG2(C1~P,
      C2~PL,C3~U)@EN
Species      CMG2_13_er_3H      CMG2(C1~PL,C2~U,C3~PL)@EN, CMG2
      (C1~PL,C2~U,C3~PL)@EN, CMG2(C1~PL,C2~U,C3~P)@EN, CMG2(C1~P,
      C2~U,C3~PL)@EN
Species      CMG2_23_er_3H      CMG2(C1~U,C2~PL,C3~PL)@EN, CMG2
      (C1~U,C2~PL,C3~PL)@EN, CMG2(C1~U,C2~PL,C3~P)@EN, CMG2(C1~U,
      C2~P,C3~PL)@EN
Species      CMG2_123_er_3H      CMG2(C1~PL,C2~PL,C3~PL)@EN,
      CMG2(C1~PL,C2~PL,C3~PL)@EN, CMG2(C1~PL,C2~PL,C3~PL)@EN

319 Species      CMG2_1_go_3H      CMG2(C1~PL,C2~U,C3~U)@G0
Species      CMG2_2_go_3H      CMG2(C1~U,C2~PL,C3~U)@G0
Species      CMG2_3_go_3H      CMG2(C1~U,C2~U,C3~PL)@G0
Species      CMG2_12_go_3H      CMG2(C1~PL,C2~PL,C3~U)@G0, CMG2
      (C1~PL,C2~PL,C3~U)@G0, CMG2(C1~PL,C2~P,C3~U)@G0, CMG2(C1~P,
      C2~PL,C3~U)@G0
324 Species      CMG2_13_go_3H      CMG2(C1~PL,C2~U,C3~PL)@G0, CMG2
      (C1~PL,C2~U,C3~PL)@G0, CMG2(C1~PL,C2~U,C3~P)@G0, CMG2(C1~P,
      C2~U,C3~PL)@G0
Species      CMG2_23_go_3H      CMG2(C1~U,C2~PL,C3~PL)@G0, CMG2
      (C1~U,C2~PL,C3~PL)@G0, CMG2(C1~U,C2~PL,C3~P)@G0, CMG2(C1~U,
      C2~P,C3~PL)@G0

```



```

Species      CMG2_123_go_3H   CMG2(C1~PL,C2~PL,C3~PL)@G0,
              CMG2(C1~PL,C2~PL,C3~PL)@G0, CMG2(C1~PL,C2~PL,C3~PL)@G0

Species      CMG2_1_pm_3H     CMG2(C1~PL,C2~U,C3~U)@PM
329 Species    CMG2_2_pm_3H     CMG2(C1~U,C2~PL,C3~U)@PM
Species      CMG2_3_pm_3H     CMG2(C1~U,C2~U,C3~PL)@PM
Species      CMG2_12_pm_3H    CMG2(C1~PL,C2~PL,C3~U)@PM, CMG2
              (C1~PL,C2~PL,C3~U)@PM, CMG2(C1~PL,C2~P,C3~U)@PM, CMG2(C1~P,
              C2~PL,C3~U)@PM
Species      CMG2_13_pm_3H    CMG2(C1~PL,C2~U,C3~PL)@PM, CMG2
              (C1~PL,C2~U,C3~PL)@PM, CMG2(C1~PL,C2~U,C3~P)@PM, CMG2(C1~P,
              C2~U,C3~PL)@PM
Species      CMG2_23_pm_3H    CMG2(C1~U,C2~PL,C3~PL)@PM, CMG2
              (C1~U,C2~PL,C3~PL)@PM, CMG2(C1~U,C2~PL,C3~P)@PM, CMG2(C1~U,
              C2~P,C3~PL)@PM
334 Species    CMG2_123_pm_3H  CMG2(C1~PL,C2~PL,C3~PL)@PM,
              CMG2(C1~PL,C2~PL,C3~PL)@PM, CMG2(C1~PL,C2~PL,C3~PL)@PM

Species      CMG2_1_end_3H    CMG2(C1~PL,C2~U,C3~U)@END
Species      CMG2_2_end_3H    CMG2(C1~U,C2~PL,C3~U)@END
Species      CMG2_3_end_3H    CMG2(C1~U,C2~U,C3~PL)@END
339 Species    CMG2_12_end_3H  CMG2(C1~PL,C2~PL,C3~U)@END,
              CMG2(C1~PL,C2~PL,C3~U)@END, CMG2(C1~PL,C2~P,C3~U)@END, CMG2(
              C1~P,C2~PL,C3~U)@END
Species      CMG2_13_end_3H   CMG2(C1~PL,C2~U,C3~PL)@END,
              CMG2(C1~PL,C2~U,C3~PL)@END, CMG2(C1~PL,C2~U,C3~P)@END, CMG2(
              C1~P,C2~U,C3~PL)@END
Species      CMG2_23_end_3H   CMG2(C1~U,C2~PL,C3~PL)@END,
              CMG2(C1~U,C2~PL,C3~PL)@END, CMG2(C1~U,C2~PL,C3~P)@END, CMG2(
              C1~U,C2~P,C3~PL)@END
Species      CMG2_123_end_3H  CMG2(C1~PL,C2~PL,C3~PL)@END,
              CMG2(C1~PL,C2~PL,C3~PL)@END, CMG2(C1~PL,C2~PL,C3~PL)@END

344 end observables

begin functions
PAT7_compet() = CMG2_0_er/KM7_0_1 + CMG2_0_er/KM7_0_2 +
                CMG2_0_er/KM7_0_3 + CMG2_1_er/KM7_1_2 + CMG2_1_er/KM7_1_3 +
                CMG2_2_er/KM7_2_1 + CMG2_2_er/KM7_2_3 + CMG2_3_er/KM7_3_1 +
                CMG2_3_er/KM7_3_2 + CMG2_12_er/KM7_12 + CMG2_13_er/KM7_13 +
                CMG2_23_er/KM7_23
PAT3_compet() = CMG2_0_go/KM3_0_1 + CMG2_0_go/KM3_0_2 +
                CMG2_0_go/KM3_0_3 + CMG2_1_go/KM3_1_2 + CMG2_1_go/KM3_1_3 +
                CMG2_2_go/KM3_2_1 + CMG2_2_go/KM3_2_3 + CMG2_3_go/KM3_3_1 +
                CMG2_3_go/KM3_3_2 + CMG2_12_go/KM3_12 + CMG2_13_go/KM3_13 +
                CMG2_23_go/KM3_23

```

```

349     APTer_compet() = CMG2_1_er/KM_APTer_1 + CMG2_2_er/KM_APTer_2 +
        CMG2_3_er/KM_APTer_3 + CMG2_12_er/KM_APTer_12_1 + CMG2_12_er/
        /KM_APTer_12_2 + CMG2_13_er/KM_APTer_13_1 + CMG2_13_er/
        KM_APTer_13_3 + CMG2_23_er/KM_APTer_23_2 + CMG2_23_er/
        KM_APTer_23_3 + CMG2_123_er/KM_APTer_123_1 + CMG2_123_er/
        KM_APTer_123_2 + CMG2_123_er/KM_APTer_123_3
    APTgo_compet() = CMG2_1_go/KM_APTgo_1 + CMG2_2_go/KM_APTgo_2 +
        CMG2_3_go/KM_APTgo_3 + CMG2_12_go/KM_APTgo_12_1 + CMG2_12_go/
        /KM_APTgo_12_2 + CMG2_13_go/KM_APTgo_13_1 + CMG2_13_go/
        KM_APTgo_13_3 + CMG2_23_go/KM_APTgo_23_2 + CMG2_23_go/
        KM_APTgo_23_3 + CMG2_123_go/KM_APTgo_123_1 + CMG2_123_go/
        KM_APTgo_123_2 + CMG2_123_go/KM_APTgo_123_3
    APTpm_compet() = CMG2_1_pm/KM_APTpm_1 + CMG2_2_pm/KM_APTpm_2 +
        CMG2_3_pm/KM_APTpm_3 + CMG2_12_pm/KM_APTpm_12_1 + CMG2_12_pm/
        /KM_APTpm_12_2 + CMG2_13_pm/KM_APTpm_13_1 + CMG2_13_pm/
        KM_APTpm_13_3 + CMG2_23_pm/KM_APTpm_23_2 + CMG2_23_pm/
        KM_APTpm_23_3 + CMG2_123_pm/KM_APTpm_123_1 + CMG2_123_pm/
        KM_APTpm_123_2 + CMG2_123_pm/KM_APTpm_123_3

    end functions

354     begin reaction rules
    # Synthesis and Folding
    Synthesis() -> Synthesis()+CMG2(L~off,F~n,C1~U,C2~U,C3~U)@EN      (1-
        S35switch)*ks
    CMG2(F~n)@EN -> CMG2(F~y)@EN      kf

359     # S35 labelling
    Synthesis() -> Synthesis()+CMG2(L~on,F~n,C1~U,C2~U,C3~U)@EN
        S35switch*ks

    # Transport
364     CMG2(F~y)@EN -> CMG2(F~y)@G0      k_go

        CMG2(C1~U,C2~U,C3~U)@G0 -> CMG2(C1~U,C2~U,C3~U)@PM      k_pm_0
        CMG2(C1~P)@G0 -> CMG2(C1~P)@PM      k_pm_1
        CMG2(C1~PL)@G0 -> CMG2(C1~PL)@PM      k_pm_1
369     CMG2(C2~P)@G0 -> CMG2(C2~P)@PM      k_pm_2
        CMG2(C2~PL)@G0 -> CMG2(C2~PL)@PM      k_pm_2
        CMG2(C3~P)@G0 -> CMG2(C3~P)@PM      k_pm_3
        CMG2(C3~PL)@G0 -> CMG2(C3~PL)@PM      k_pm_3

374     CMG2()@PM <-> CMG2()@END      k_to_end, k_from_end

    # Palmitoylation ER
    CMG2(C1~U,C2~U,C3~U)@EN -> CMG2(C1~P,C2~U,C3~U)@EN      (1-
        H3switch)*kcat7*PAT7/KM7_0_1/(1+PAT7_compet()+PAT7/KM7_0_1)

```

```

CMG2(C1~U,C2~U,C3~U)@EN -> CMG2(C1~U,C2~P,C3~U)@EN          (1-
H3switch)*kcat7*PAT7/KM7_0_2/(1+PAT7_compet()+PAT7/KM7_0_2)
379 CMG2(C1~U,C2~U,C3~U)@EN -> CMG2(C1~U,C2~U,C3~P)@EN          (1-
H3switch)*kcat7*PAT7/KM7_0_3/(1+PAT7_compet()+PAT7/KM7_0_3)

CMG2(C2~U,C3~U)@EN -> CMG2(C2~P,C3~U)@EN          (1-H3switch)*
kcat7*PAT7/KM7_1_2/(1+PAT7_compet()+PAT7/KM7_1_2) exclude_reactants
(1,CMG2(C1~U))
CMG2(C2~U,C3~U)@EN -> CMG2(C2~U,C3~P)@EN          (1-H3switch)*
kcat7*PAT7/KM7_1_3/(1+PAT7_compet()+PAT7/KM7_1_3) exclude_reactants
(1,CMG2(C1~U))

384 CMG2(C1~U,C3~U)@EN -> CMG2(C1~P,C3~U)@EN          (1-H3switch)*
kcat7*PAT7/KM7_2_1/(1+PAT7_compet()+PAT7/KM7_2_1) exclude_reactants
(1,CMG2(C2~U))
CMG2(C1~U,C3~U)@EN -> CMG2(C1~U,C3~P)@EN          (1-H3switch)*
kcat7*PAT7/KM7_2_3/(1+PAT7_compet()+PAT7/KM7_2_3) exclude_reactants
(1,CMG2(C2~U))

CMG2(C1~U,C2~U)@EN -> CMG2(C1~P,C2~U)@EN          (1-H3switch)*
kcat7*PAT7/KM7_3_1/(1+PAT7_compet()+PAT7/KM7_3_1) exclude_reactants
(1,CMG2(C3~U))
CMG2(C1~U,C2~U)@EN -> CMG2(C1~U,C2~P)@EN          (1-H3switch)*
kcat7*PAT7/KM7_3_2/(1+PAT7_compet()+PAT7/KM7_3_2) exclude_reactants
(1,CMG2(C3~U))

389 CMG2(C3~U)@EN -> CMG2(C3~P)@EN          (1-H3switch)*kcat7*PAT7/KM7_12
/(1+PAT7_compet()+PAT7/KM7_12) exclude_reactants(1,CMG2(C1~U),
CMG2(C2~U))
CMG2(C2~U)@EN -> CMG2(C2~P)@EN          (1-H3switch)*kcat7*PAT7/KM7_13
/(1+PAT7_compet()+PAT7/KM7_13) exclude_reactants(1,CMG2(C1~U),
CMG2(C3~U))
CMG2(C1~U)@EN -> CMG2(C1~P)@EN          (1-H3switch)*kcat7*PAT7/KM7_23
/(1+PAT7_compet()+PAT7/KM7_23) exclude_reactants(1,CMG2(C2~U),
CMG2(C3~U))

394 # Palm labelling ER
CMG2(C1~U,C2~U,C3~U)@EN -> CMG2(C1~PL,C2~U,C3~U)@EN
H3switch*kcat7*PAT7/KM7_0_1/(1+PAT7_compet()+PAT7/KM7_0_1)
CMG2(C1~U,C2~U,C3~U)@EN -> CMG2(C1~U,C2~PL,C3~U)@EN
H3switch*kcat7*PAT7/KM7_0_2/(1+PAT7_compet()+PAT7/KM7_0_2)
CMG2(C1~U,C2~U,C3~U)@EN -> CMG2(C1~U,C2~U,C3~PL)@EN
H3switch*kcat7*PAT7/KM7_0_3/(1+PAT7_compet()+PAT7/KM7_0_3)

399 CMG2(C2~U,C3~U)@EN -> CMG2(C2~PL,C3~U)@EN          H3switch*kcat7*
PAT7/KM7_1_2/(1+PAT7_compet()+PAT7/KM7_1_2) exclude_reactants
(1,CMG2(C1~U))

```

```

CMG2(C2~U,C3~U)@EN -> CMG2(C2~U,C3~PL)@EN          H3switch*kcat7*
PAT7/KM7_1_3/(1+PAT7_compet()+PAT7/KM7_1_3)          exclude_reactants
(1,CMG2(C1~U))

CMG2(C1~U,C3~U)@EN -> CMG2(C1~PL,C3~U)@EN          H3switch*kcat7*
PAT7/KM7_2_1/(1+PAT7_compet()+PAT7/KM7_2_1)          exclude_reactants
(1,CMG2(C2~U))

CMG2(C1~U,C3~U)@EN -> CMG2(C1~U,C3~PL)@EN          H3switch*kcat7*
PAT7/KM7_2_3/(1+PAT7_compet()+PAT7/KM7_2_3)          exclude_reactants
(1,CMG2(C2~U))

404 CMG2(C1~U,C2~U)@EN -> CMG2(C1~PL,C2~U)@EN          H3switch*kcat7*
PAT7/KM7_3_1/(1+PAT7_compet()+PAT7/KM7_3_1)          exclude_reactants
(1,CMG2(C3~U))

CMG2(C1~U,C2~U)@EN -> CMG2(C1~U,C2~PL)@EN          H3switch*kcat7*
PAT7/KM7_3_2/(1+PAT7_compet()+PAT7/KM7_3_2)          exclude_reactants
(1,CMG2(C3~U))

CMG2(C3~U)@EN -> CMG2(C3~PL)@EN          H3switch*kcat7*PAT7/KM7_12/(1+
PAT7_compet()+PAT7/KM7_12)          exclude_reactants(1,CMG2(C1~U),
CMG2(C2~U))

409 CMG2(C2~U)@EN -> CMG2(C2~PL)@EN          H3switch*kcat7*PAT7/KM7_13/(1+
PAT7_compet()+PAT7/KM7_13)          exclude_reactants(1,CMG2(C1~U),
CMG2(C3~U))

CMG2(C1~U)@EN -> CMG2(C1~PL)@EN          H3switch*kcat7*PAT7/KM7_23/(1+
PAT7_compet()+PAT7/KM7_23)          exclude_reactants(1,CMG2(C2~U),
CMG2(C3~U))

# Palmitoylation G0
CMG2(C1~U,C2~U,C3~U)@G0 -> CMG2(C1~P,C2~U,C3~U)@G0          (1-
H3switch)*kcat3*PAT3/KM3_0_1/(1+PAT3_compet()+PAT3/KM3_0_1)

414 CMG2(C1~U,C2~U,C3~U)@G0 -> CMG2(C1~U,C2~P,C3~U)@G0          (1-
H3switch)*kcat3*PAT3/KM3_0_2/(1+PAT3_compet()+PAT3/KM3_0_2)

CMG2(C1~U,C2~U,C3~U)@G0 -> CMG2(C1~U,C2~U,C3~P)@G0          (1-
H3switch)*kcat3*PAT3/KM3_0_3/(1+PAT3_compet()+PAT3/KM3_0_3)

CMG2(C2~U,C3~U)@G0 -> CMG2(C2~P,C3~U)@G0          (1-H3switch)*
kcat3*PAT3/KM3_1_2/(1+PAT3_compet()+PAT3/KM3_1_2)          exclude_reactants
(1,CMG2(C1~U))

CMG2(C2~U,C3~U)@G0 -> CMG2(C2~U,C3~P)@G0          (1-H3switch)*
kcat3*PAT3/KM3_1_3/(1+PAT3_compet()+PAT3/KM3_1_3)          exclude_reactants
(1,CMG2(C1~U))

419 CMG2(C1~U,C3~U)@G0 -> CMG2(C1~P,C3~U)@G0          (1-H3switch)*
kcat3*PAT3/KM3_2_1/(1+PAT3_compet()+PAT3/KM3_2_1)          exclude_reactants
(1,CMG2(C2~U))

```

```

CMG2(C1~U,C3~U)@G0 -> CMG2(C1~U,C3~P)@G0          (1-H3switch)*
    kcat3*PAT3/KM3_2_3/(1+PAT3_compet()+PAT3/KM3_2_3)  exclude_reactants
    (1,CMG2(C2~U))

CMG2(C1~U,C2~U)@G0 -> CMG2(C1~P,C2~U)@G0          (1-H3switch)*
    kcat3*PAT3/KM3_3_1/(1+PAT3_compet()+PAT3/KM3_3_1)  exclude_reactants
    (1,CMG2(C3~U))
424 CMG2(C1~U,C2~U)@G0 -> CMG2(C1~U,C2~P)@G0          (1-H3switch)*
    kcat3*PAT3/KM3_3_2/(1+PAT3_compet()+PAT3/KM3_3_2)  exclude_reactants
    (1,CMG2(C3~U))

CMG2(C3~U)@G0 -> CMG2(C3~P)@G0          (1-H3switch)*kcat3*PAT3/KM3_12
    /(1+PAT3_compet()+PAT3/KM3_12)  exclude_reactants(1,CMG2(C1~U),
    CMG2(C2~U))
CMG2(C2~U)@G0 -> CMG2(C2~P)@G0          (1-H3switch)*kcat3*PAT3/KM3_13
    /(1+PAT3_compet()+PAT3/KM3_13)  exclude_reactants(1,CMG2(C1~U),
    CMG2(C3~U))
CMG2(C1~U)@G0 -> CMG2(C1~P)@G0          (1-H3switch)*kcat3*PAT3/KM3_23
    /(1+PAT3_compet()+PAT3/KM3_23)  exclude_reactants(1,CMG2(C2~U),
    CMG2(C3~U))
429
# Palm labelling G0
CMG2(C1~U,C2~U,C3~U)@G0 -> CMG2(C1~PL,C2~U,C3~U)@G0
    H3switch*kcat3*PAT3/KM3_0_1/(1+PAT3_compet()+PAT3/KM3_0_1)
CMG2(C1~U,C2~U,C3~U)@G0 -> CMG2(C1~U,C2~PL,C3~U)@G0
    H3switch*kcat3*PAT3/KM3_0_2/(1+PAT3_compet()+PAT3/KM3_0_2)
CMG2(C1~U,C2~U,C3~U)@G0 -> CMG2(C1~U,C2~U,C3~PL)@G0
    H3switch*kcat3*PAT3/KM3_0_3/(1+PAT3_compet()+PAT3/KM3_0_3)
434
CMG2(C2~U,C3~U)@G0 -> CMG2(C2~PL,C3~U)@G0          H3switch*kcat3*
    PAT3/KM3_1_2/(1+PAT3_compet()+PAT3/KM3_1_2)  exclude_reactants
    (1,CMG2(C1~U))
CMG2(C2~U,C3~U)@G0 -> CMG2(C2~U,C3~PL)@G0          H3switch*kcat3*
    PAT3/KM3_1_3/(1+PAT3_compet()+PAT3/KM3_1_3)  exclude_reactants
    (1,CMG2(C1~U))

CMG2(C1~U,C3~U)@G0 -> CMG2(C1~PL,C3~U)@G0          H3switch*kcat3*
    PAT3/KM3_2_1/(1+PAT3_compet()+PAT3/KM3_2_1)  exclude_reactants
    (1,CMG2(C2~U))
439 CMG2(C1~U,C3~U)@G0 -> CMG2(C1~U,C3~PL)@G0          H3switch*kcat3*
    PAT3/KM3_2_3/(1+PAT3_compet()+PAT3/KM3_2_3)  exclude_reactants
    (1,CMG2(C2~U))

CMG2(C1~U,C2~U)@G0 -> CMG2(C1~PL,C2~U)@G0          H3switch*kcat3*
    PAT3/KM3_3_1/(1+PAT3_compet()+PAT3/KM3_3_1)  exclude_reactants
    (1,CMG2(C3~U))

```

```

CMG2(C1~U,C2~U)@G0 -> CMG2(C1~U,C2~PL)@G0          H3switch*kcat3*
PAT3/KM3_3_2/(1+PAT3_compet()+PAT3/KM3_3_2)          exclude_reactants
(1,CMG2(C3~U))

444 CMG2(C3~U)@G0 -> CMG2(C3~PL)@G0          H3switch*kcat3*PAT3/KM3_12/(1+
PAT3_compet()+PAT3/KM3_12)          exclude_reactants(1,CMG2(C1~U),
CMG2(C2~U))
CMG2(C2~U)@G0 -> CMG2(C2~PL)@G0          H3switch*kcat3*PAT3/KM3_13/(1+
PAT3_compet()+PAT3/KM3_13)          exclude_reactants(1,CMG2(C1~U),
CMG2(C3~U))
CMG2(C1~U)@G0 -> CMG2(C1~PL)@G0          H3switch*kcat3*PAT3/KM3_23/(1+
PAT3_compet()+PAT3/KM3_23)          exclude_reactants(1,CMG2(C2~U),
CMG2(C3~U))

# Depalmitoylation ER
449 CMG2(C1~P,C2~U,C3~U)@EN -> CMG2(C1~U,C2~U,C3~U)@EN
kcat_APTer*APTter/KM_APTer_1/(1+APTter_compet()+APTter/KM_APTer_1)
exclude_reactants(1,CMG2(C1~U))
CMG2(C1~U,C2,C3~U)@EN -> CMG2(C1~U,C2~U,C3~U)@EN
kcat_APTer*APTter/KM_APTer_2/(1+APTter_compet()+APTter/KM_APTer_2)
exclude_reactants(1,CMG2(C2~U))
CMG2(C1~U,C2~U,C3)@EN -> CMG2(C1~U,C2~U,C3~U)@EN
kcat_APTer*APTter/KM_APTer_3/(1+APTter_compet()+APTter/KM_APTer_3)
exclude_reactants(1,CMG2(C3~U))

CMG2(C1,C3~U)@EN -> CMG2(C1~U,C3~U)@EN          kcat_APTer*APTter/
KM_APTer_12_1/(1+APTter_compet()+APTter/KM_APTer_12_2)
exclude_reactants(1,CMG2(C1~U),CMG2(C2~U))
454 CMG2(C2,C3~U)@EN -> CMG2(C2~U,C3~U)@EN          kcat_APTer*APTter/
KM_APTer_12_2/(1+APTter_compet()+APTter/KM_APTer_12_2)
exclude_reactants(1,CMG2(C1~U),CMG2(C2~U))

CMG2(C1,C2~U)@EN -> CMG2(C1~U,C2~U)@EN          kcat_APTer*APTter/
KM_APTer_13_1/(1+APTter_compet()+APTter/KM_APTer_13_1)
exclude_reactants(1,CMG2(C1~U),CMG2(C3~U))
CMG2(C2~U,C3)@EN -> CMG2(C2~U,C3~U)@EN          kcat_APTer*APTter/
KM_APTer_13_3/(1+APTter_compet()+APTter/KM_APTer_13_3)
exclude_reactants(1,CMG2(C1~U),CMG2(C3~U))

459 CMG2(C1~U,C2)@EN -> CMG2(C1~U,C2~U)@EN          kcat_APTer*APTter/
KM_APTer_23_2/(1+APTter_compet()+APTter/KM_APTer_23_2)
exclude_reactants(1,CMG2(C2~U),CMG2(C3~U))
CMG2(C1~U,C3)@EN -> CMG2(C1~U,C3~U)@EN          kcat_APTer*APTter/
KM_APTer_23_3/(1+APTter_compet()+APTter/KM_APTer_23_3)
exclude_reactants(1,CMG2(C2~U),CMG2(C3~U))

```

```

CMG2(C1)@EN -> CMG2(C1~U)@EN          kcat_APTer*APTer/KM_APTer_123_1
      /(1+APTer_compet()+APTer/KM_APTer_123_1) exclude_reactants(1,CMG2(C1
      ~U),CMG2(C2~U),CMG2(C3~U))
CMG2(C2)@EN -> CMG2(C2~U)@EN          kcat_APTer*APTer/KM_APTer_123_2
      /(1+APTer_compet()+APTer/KM_APTer_123_2) exclude_reactants(1,CMG2(C1
      ~U),CMG2(C2~U),CMG2(C3~U))
464 CMG2(C3)@EN -> CMG2(C3~U)@EN          kcat_APTer*APTer/KM_APTer_123_3
      /(1+APTer_compet()+APTer/KM_APTer_123_3) exclude_reactants(1,CMG2(C1
      ~U),CMG2(C2~U),CMG2(C3~U))

# Depalmitoylation G0
CMG2(C1,C2~U,C3~U)@G0 -> CMG2(C1~U,C2~U,C3~U)@G0
      kcat_APTgo*APTgo/KM_APTgo_1/(1+APTgo_compet()+APTgo/KM_APTgo_1)
      exclude_reactants(1,CMG2(C1~U))
CMG2(C1~U,C2,C3~U)@G0 -> CMG2(C1~U,C2~U,C3~U)@G0
      kcat_APTgo*APTgo/KM_APTgo_2/(1+APTgo_compet()+APTgo/KM_APTgo_2)
      exclude_reactants(1,CMG2(C2~U))
469 CMG2(C1~U,C2~U,C3)@G0 -> CMG2(C1~U,C2~U,C3~U)@G0
      kcat_APTgo*APTgo/KM_APTgo_3/(1+APTgo_compet()+APTgo/KM_APTgo_3)
      exclude_reactants(1,CMG2(C3~U))

CMG2(C1,C3~U)@G0 -> CMG2(C1~U,C3~U)@G0          kcat_APTgo*APTgo/
      KM_APTgo_12_1/(1+APTgo_compet()+APTgo/KM_APTgo_12_2)
      exclude_reactants(1,CMG2(C1~U),CMG2(C2~U))
CMG2(C2,C3~U)@G0 -> CMG2(C2~U,C3~U)@G0          kcat_APTgo*APTgo/
      KM_APTgo_12_2/(1+APTgo_compet()+APTgo/KM_APTgo_12_2)
      exclude_reactants(1,CMG2(C1~U),CMG2(C2~U))

474 CMG2(C1,C2~U)@G0 -> CMG2(C1~U,C2~U)@G0          kcat_APTgo*APTgo/
      KM_APTgo_13_1/(1+APTgo_compet()+APTgo/KM_APTgo_13_1)
      exclude_reactants(1,CMG2(C1~U),CMG2(C3~U))
CMG2(C2~U,C3)@G0 -> CMG2(C2~U,C3~U)@G0          kcat_APTgo*APTgo/
      KM_APTgo_13_3/(1+APTgo_compet()+APTgo/KM_APTgo_13_3)
      exclude_reactants(1,CMG2(C1~U),CMG2(C3~U))

CMG2(C1~U,C2)@G0 -> CMG2(C1~U,C2~U)@G0          kcat_APTgo*APTgo/
      KM_APTgo_23_2/(1+APTgo_compet()+APTgo/KM_APTgo_23_2)
      exclude_reactants(1,CMG2(C2~U),CMG2(C3~U))
CMG2(C1~U,C3)@G0 -> CMG2(C1~U,C3~U)@G0          kcat_APTgo*APTgo/
      KM_APTgo_23_3/(1+APTgo_compet()+APTgo/KM_APTgo_23_3)
      exclude_reactants(1,CMG2(C2~U),CMG2(C3~U))

479 CMG2(C1)@G0 -> CMG2(C1~U)@G0          kcat_APTgo*APTgo/KM_APTgo_123_1
      /(1+APTgo_compet()+APTgo/KM_APTgo_123_1) exclude_reactants(1,CMG2(C1
      ~U),CMG2(C2~U),CMG2(C3~U))

```

```

CMG2(C2)@G0 -> CMG2(C2~U)@G0          kcat_APTgo*APTgo/KM_APTgo_123_2
      /(1+APTgo_compet()+APTgo/KM_APTgo_123_2) exclude_reactants(1,CMG2(C1
      ~U),CMG2(C2~U),CMG2(C3~U))
CMG2(C3)@G0 -> CMG2(C3~U)@G0          kcat_APTgo*APTgo/KM_APTgo_123_3
      /(1+APTgo_compet()+APTgo/KM_APTgo_123_3) exclude_reactants(1,CMG2(C1
      ~U),CMG2(C2~U),CMG2(C3~U))

484 # Depalmitoylation PM
CMG2(C1,C2~U,C3~U)@PM -> CMG2(C1~U,C2~U,C3~U)@PM
      kcat_APTpm*APTpm/KM_APTpm_1/(1+APTpm_compet()+APTpm/KM_APTpm_1)
      exclude_reactants(1,CMG2(C1~U))
CMG2(C1~U,C2,C3~U)@PM -> CMG2(C1~U,C2~U,C3~U)@PM
      kcat_APTpm*APTpm/KM_APTpm_2/(1+APTpm_compet()+APTpm/KM_APTpm_2)
      exclude_reactants(1,CMG2(C2~U))
CMG2(C1~U,C2~U,C3)@PM -> CMG2(C1~U,C2~U,C3~U)@PM
      kcat_APTpm*APTpm/KM_APTpm_3/(1+APTpm_compet()+APTpm/KM_APTpm_3)
      exclude_reactants(1,CMG2(C3~U))

489 CMG2(C1,C3~U)@PM -> CMG2(C1~U,C3~U)@PM          kcat_APTpm*APTpm/
      KM_APTpm_12_1/(1+APTpm_compet()+APTpm/KM_APTpm_12_2)
      exclude_reactants(1,CMG2(C1~U),CMG2(C2~U))
CMG2(C2,C3~U)@PM -> CMG2(C2~U,C3~U)@PM          kcat_APTpm*APTpm/
      KM_APTpm_12_2/(1+APTpm_compet()+APTpm/KM_APTpm_12_2)
      exclude_reactants(1,CMG2(C1~U),CMG2(C2~U))

CMG2(C1,C2~U)@PM -> CMG2(C1~U,C2~U)@PM          kcat_APTpm*APTpm/
      KM_APTpm_13_1/(1+APTpm_compet()+APTpm/KM_APTpm_13_1)
      exclude_reactants(1,CMG2(C1~U),CMG2(C3~U))
CMG2(C2~U,C3)@PM -> CMG2(C2~U,C3~U)@PM          kcat_APTpm*APTpm/
      KM_APTpm_13_3/(1+APTpm_compet()+APTpm/KM_APTpm_13_3)
      exclude_reactants(1,CMG2(C1~U),CMG2(C3~U))

494 CMG2(C1~U,C2)@PM -> CMG2(C1~U,C2~U)@PM          kcat_APTpm*APTpm/
      KM_APTpm_23_2/(1+APTpm_compet()+APTpm/KM_APTpm_23_2)
      exclude_reactants(1,CMG2(C2~U),CMG2(C3~U))
CMG2(C1~U,C3)@PM -> CMG2(C1~U,C3~U)@PM          kcat_APTpm*APTpm/
      KM_APTpm_23_3/(1+APTpm_compet()+APTpm/KM_APTpm_23_3)
      exclude_reactants(1,CMG2(C2~U),CMG2(C3~U))

CMG2(C1)@PM -> CMG2(C1~U)@PM          kcat_APTpm*APTpm/KM_APTpm_123_1
      /(1+APTpm_compet()+APTpm/KM_APTpm_123_1) exclude_reactants(1,CMG2(C1
      ~U),CMG2(C2~U),CMG2(C3~U))
499 CMG2(C2)@PM -> CMG2(C2~U)@PM          kcat_APTpm*APTpm/KM_APTpm_123_2
      /(1+APTpm_compet()+APTpm/KM_APTpm_123_2) exclude_reactants(1,CMG2(C1
      ~U),CMG2(C2~U),CMG2(C3~U))

```



```

CMG2(C3)@PM -> CMG2(C3~U)@PM          kcat_APTpm*APTpm/KM_APTpm_123_3
      /(1+APTpm_compet()+APTpm/KM_APTpm_123_3) exclude_reactants(1,CMG2(C1
      ~U),CMG2(C2~U),CMG2(C3~U))

# Degradation ER
CMG2(F~n,C1~U,C2~U,C3~U)@EN -> Degradation()      kd_nf_0_er
504 CMG2(F~n,C2~U,C3~U)@EN -> Degradation()          kd_nf_1_er
      exclude_reactants(1,CMG2(C1~U))
CMG2(F~n,C1~U,C3~U)@EN -> Degradation()            kd_nf_2_er
      exclude_reactants(1,CMG2(C2~U))
CMG2(F~n,C1~U,C2~U)@EN -> Degradation()            kd_nf_3_er
      exclude_reactants(1,CMG2(C3~U))
CMG2(F~n,C3~U)@EN -> Degradation()                 kd_nf_12_er
      exclude_reactants(1,CMG2(C1~U),CMG2(C2~U))
CMG2(F~n,C2~U)@EN -> Degradation()                 kd_nf_13_er
      exclude_reactants(1,CMG2(C1~U),CMG2(C3~U))
509 CMG2(F~n,C1~U)@EN -> Degradation()              kd_nf_23_er
      exclude_reactants(1,CMG2(C2~U),CMG2(C3~U))
CMG2(F~n)@EN -> Degradation()                     kd_nf_123_er
      exclude_reactants(1,CMG2(C1~U),CMG2(C2~U),CMG2(C3~U))

CMG2(F~y,C1~U,C2~U,C3~U)@EN -> Degradation()      kd_f_0_er
CMG2(F~y,C2~U,C3~U)@EN -> Degradation()          kd_f_1_er
      exclude_reactants(1,CMG2(C1~U))
514 CMG2(F~y,C1~U,C3~U)@EN -> Degradation()        kd_f_2_er
      exclude_reactants(1,CMG2(C2~U))
CMG2(F~y,C1~U,C2~U)@EN -> Degradation()          kd_f_3_er
      exclude_reactants(1,CMG2(C3~U))
CMG2(F~y,C3~U)@EN -> Degradation()               kd_f_12_er
      exclude_reactants(1,CMG2(C1~U),CMG2(C2~U))
CMG2(F~y,C2~U)@EN -> Degradation()               kd_f_13_er
      exclude_reactants(1,CMG2(C1~U),CMG2(C3~U))
CMG2(F~y,C1~U)@EN -> Degradation()               kd_f_23_er
      exclude_reactants(1,CMG2(C2~U),CMG2(C3~U))
519 CMG2(F~y)@EN -> Degradation()                 kd_f_123_er
      exclude_reactants(1,CMG2(C1~U),CMG2(C2~U),CMG2(C3~U))

# Degradation non-folded GO
# CMG2(F~n,C1~U,C2~U,C3~U)@G0 -> Degradation()   kd_nf_0_go
# CMG2(F~n,C2~U,C3~U)@G0 -> Degradation()
      kd_nf_1_go      exclude_reactants(1,CMG2(C1~U))
524 # CMG2(F~n,C1~U,C3~U)@G0 -> Degradation()
      kd_nf_2_go      exclude_reactants(1,CMG2(C2~U))
# CMG2(F~n,C1~U,C2~U)@G0 -> Degradation()
      kd_nf_3_go      exclude_reactants(1,CMG2(C3~U))
# CMG2(F~n,C3~U)@G0 -> Degradation()
      kd_nf_12_go      exclude_reactants(1,CMG2(C1~U),CMG2(C2~U))

```

```

# CMG2(F~n,C2~U)@G0 -> Degradation()
    kd_nf_13_go      exclude_reactants(1,CMG2(C1~U),CMG2(C3~U))
# CMG2(F~n,C1~U)@G0 -> Degradation()
    kd_nf_23_go      exclude_reactants(1,CMG2(C2~U),CMG2(C3~U))
529 # CMG2(F~n)@G0 -> Degradation()
    kd_nf_123_go     exclude_reactants(1,CMG2(C1~U),CMG2(C2~U),CMG2(C3~U)
    )

# Degradation folded G0
CMG2(F~y,C1~U,C2~U,C3~U)@G0 -> Degradation()      kd_f_0_go
CMG2(F~y,C2~U,C3~U)@G0 -> Degradation()            kd_f_1_go
    exclude_reactants(1,CMG2(C1~U))
534 CMG2(F~y,C1~U,C3~U)@G0 -> Degradation()          kd_f_2_go
    exclude_reactants(1,CMG2(C2~U))
CMG2(F~y,C1~U,C2~U)@G0 -> Degradation()            kd_f_3_go
    exclude_reactants(1,CMG2(C3~U))
CMG2(F~y,C3~U)@G0 -> Degradation()                 kd_f_12_go
    exclude_reactants(1,CMG2(C1~U),CMG2(C2~U))
CMG2(F~y,C2~U)@G0 -> Degradation()                 kd_f_13_go
    exclude_reactants(1,CMG2(C1~U),CMG2(C3~U))
CMG2(F~y,C1~U)@G0 -> Degradation()                 kd_f_23_go
    exclude_reactants(1,CMG2(C2~U),CMG2(C3~U))
539 CMG2(F~y)@G0 -> Degradation()                   kd_f_123_go
    exclude_reactants(1,CMG2(C1~U),CMG2(C2~U),CMG2(C3~U))

# Degradation non-folded PM
# CMG2(F~n,C1~U,C2~U,C3~U)@PM -> Degradation()    kd_nf_0_pm
# CMG2(F~n,C2~U,C3~U)@PM -> Degradation()
    kd_nf_1_pm      exclude_reactants(1,CMG2(C1~U))
544 # CMG2(F~n,C1~U,C3~U)@PM -> Degradation()
    kd_nf_2_pm      exclude_reactants(1,CMG2(C2~U))
# CMG2(F~n,C1~U,C2~U)@PM -> Degradation()
    kd_nf_3_pm      exclude_reactants(1,CMG2(C3~U))
# CMG2(F~n,C3~U)@PM -> Degradation()
    kd_nf_12_pm     exclude_reactants(1,CMG2(C1~U),CMG2(C2~U))
# CMG2(F~n,C2~U)@PM -> Degradation()
    kd_nf_13_pm     exclude_reactants(1,CMG2(C1~U),CMG2(C3~U))
# CMG2(F~n,C1~U)@PM -> Degradation()
    kd_nf_23_pm     exclude_reactants(1,CMG2(C2~U),CMG2(C3~U))
549 # CMG2(F~n)@PM -> Degradation()
    kd_nf_123_pm    exclude_reactants(1,CMG2(C1~U),CMG2(C2~U),CMG2(C3~U)
    )

# Degradation folded PM
CMG2(F~y,C1~U,C2~U,C3~U)@PM -> Degradation()      kd_f_0_pm
CMG2(F~y,C2~U,C3~U)@PM -> Degradation()           kd_f_1_pm
    exclude_reactants(1,CMG2(C1~U))

```

```

554 CMG2(F~y,C1~U,C3~U)@PM -> Degradation()      kd_f_2_pm
      exclude_reactants(1,CMG2(C2~U))
CMG2(F~y,C1~U,C2~U)@PM -> Degradation()      kd_f_3_pm
      exclude_reactants(1,CMG2(C3~U))
CMG2(F~y,C3~U)@PM -> Degradation()      kd_f_12_pm
      exclude_reactants(1,CMG2(C1~U),CMG2(C2~U))
CMG2(F~y,C2~U)@PM -> Degradation()      kd_f_13_pm
      exclude_reactants(1,CMG2(C1~U),CMG2(C3~U))
CMG2(F~y,C1~U)@PM -> Degradation()      kd_f_23_pm
      exclude_reactants(1,CMG2(C2~U),CMG2(C3~U))
559 CMG2(F~y)@PM -> Degradation()      kd_f_123_pm
      exclude_reactants(1,CMG2(C1~U),CMG2(C2~U),CMG2(C3~U))

```

```
end reaction rules
```

```
end model
```

```

564 generate_network({}) # generates the .net file
writeMfile() # writes a matlab file for integrating the model

```

C.2 PARAMETER VALUES

Parameter name	Median	Standard deviation
ks	2.9733111288608667	0.027920059530934106
kf	0.01219677456363916	0.05410346576365027
k_go	-0.9845286384268077	0.035902197723798566
k_pm_o	-1.8401720097436984	0.0712426354517954
k_pm_1	-2.496218067175251	0.010789217689180683
k_pm_2	-0.6244036705101471	0.08572235338927409
k_pm_3	2.0311120462779315	0.08396284598984603
k_to_end	-0.5039683499863828	0.07718899427716996
k_from_end	1.578881853960695	0.08338243724808646
kcat7	1.1835689552612112	0.045310815385565764
KM7_o_1	0.979038242514041	0.06574736508805601
KM7_o_2	1.6549370845040283	0.05043754854787798
KM7_o_3	4.928255928520546	0.038680693368581424
KM7_1_2	1.6795367544047624	0.04093694318496363
KM7_1_3	4.95109555723578	0.04595578180621939
KM7_2_1	-0.9910372628986612	0.022236957138047377
KM7_2_3	4.924156248558496	0.03856957003314889

KM7_3_1	-0.5998224557257313	0.08420754967129002
KM7_3_2	-0.4838432829212574	0.08109218502980095
KM7_12	5.088138951206144	0.028797300031027043
KM7_13	0.8578577911892966	0.08163514547553484
KM7_23	1.6434276076569687	0.07583643556667664
kcat3	0.059675492426201654	0.05473846788428078
KM3_o_1	-0.2909939408417471	0.09577973257449182
KM3_o_2	1.971579731773251	0.03896424288631529
KM3_o_3	1.2231207560993997	0.08520450963128215
KM3_1_2	1.4889229697992832	0.025595673199029785
KM3_1_3	-1.9948712254342287	0.014473379512933192
KM3_2_1	0.5197578933772686	0.07926838894123675
KM3_2_3	1.0795682348804176	0.070267772566616
KM3_3_1	-1.4618125020322557	0.04571422125208792
KM3_3_2	-0.9478235890366769	0.05394610326763551
KM3_12	-0.014322784400398003	0.039109140516766026
KM3_13	-2.1213994552444113	0.05703541073831113
KM3_23	0.2562549070747192	0.08061510055504913
kcat_APter	-0.15443311447237013	0.07349003466832427
KM_APter_1	0.022849349937027953	0.07871354645811782
KM_APter_2	1.6411764773698336	0.07768876699413094
KM_APter_3	1.530696357359226	0.07416120625785763
KM_APter_12_1	1.1409861451406995	0.07704922847012824
KM_APter_12_2	0.4159214512427365	0.08621636721479653
KM_APter_13_1	1.578892126751533	0.07056733779715915
KM_APter_13_3	0.8833545470043069	0.06945215814440582
KM_APter_23_2	-0.42627588492270685	0.08316627182122546
KM_APter_23_3	0.08387526382800457	0.08226934557669625
KM_APter_123_1	-0.872907910038432	0.08134615058207811
KM_APter_123_2	-0.9882131831011234	0.032044519761609165
KM_APter_123_3	1.2260750125027722	0.08069210915283662
kcat_APTgo	-0.39216410096079574	0.08982773557126507
KM_APTgo_1	1.3470463336160505	0.0751720752717969
KM_APTgo_2	0.8322470095880281	0.07841147634041586
KM_APTgo_3	-0.5693164579788665	0.07988241987356269

KM_APTgo_12_1	0.5459684619422049	0.08458672302673154
KM_APTgo_12_2	0.20338435566468238	0.0887469164561708
KM_APTgo_13_1	0.9606313979053268	0.08019221032513213
KM_APTgo_13_3	1.3933851662001775	0.06209684964880542
KM_APTgo_23_2	-0.3741739718364717	0.07888458067622954
KM_APTgo_23_3	0.679988962024723	0.0795221561306065
KM_APTgo_123_1	-0.8716194559610755	0.08103058376802073
KM_APTgo_123_2	0.16104837404315597	0.08213380582036033
KM_APTgo_123_3	-0.3528007731719832	0.07809014301618852
kcat_APTpm	-0.18267877082765166	0.08004184228380844
KM_APTpm_1	0.8281326641000324	0.0761671212446328
KM_APTpm_2	0.5697681608323305	0.08937927465672832
KM_APTpm_3	0.7507163782773072	0.08140592482946268
KM_APTpm_12_1	-0.3006215016026226	0.07679886932616606
KM_APTpm_12_2	-0.4129902886297705	0.08146368351233885
KM_APTpm_13_1	-0.4860691512118437	0.0813805946991758
KM_APTpm_13_3	0.8456940545566671	0.07626569860467823
KM_APTpm_23_2	0.18241576436952123	0.0801036200214821
KM_APTpm_23_3	-0.7623656706397413	0.07587312688705178
KM_APTpm_123_1	0.54930646845919	0.08269186431772926
KM_APTpm_123_2	0.539023242522679	0.0810690163619241
KM_APTpm_123_3	-0.39727637598311205	0.08254665868184155
kd_nf_o_er	-1.483933704943151	0.03348899083134173
kd_nf_1_er	-1.2998983868851788	0.0793749408317424
kd_nf_2_er	-0.20341803867133487	0.06815327083405585
kd_nf_3_er	-0.016577215127033605	0.07964451460802216
kd_nf_12_er	-0.9313913589558943	0.05938002191443727
kd_nf_13_er	0.06422574616750096	0.07655903659197025
kd_nf_23_er	0.3877920613804088	0.08359967449307092
kd_nf_123_er	-0.7736982701317936	0.08211123285983542
kd_f_o_er	-0.450152670466341	0.04702374468638786
kd_f_1_er	-0.5260780437580073	0.03062902296823796
kd_f_2_er	-0.9936909760795654	0.016226162679596565
kd_f_3_er	1.010178554336585	0.08487997957638717
kd_f_12_er	-1.495624521318158	0.01353626929819027

kd_f_13_er	0.2264233863070702	0.0767251325591012
kd_f_23_er	-0.6529658876516717	0.0839112447590523
kd_f_123_er	-1.4731988735941042	0.0805627612228498
kd_f_o_go	-1.79370525290136	0.06845191767805724
kd_f_1_go	-1.9967094494450448	0.009630648179726855
kd_f_2_go	-1.195497987621465	0.020238722309934006
kd_f_3_go	-0.5997885542399919	0.07896680028135106
kd_f_12_go	-2.4803403001249085	0.03095809481277764
kd_f_13_go	-0.12633722134028966	0.07490841614133874
kd_f_23_go	-1.4393120090797922	0.07825812103938558
kd_f_123_go	-2.1468042924046076	0.04878883200680794
kd_f_o_pm	0.8234120797310378	0.07207908987482325
kd_f_1_pm	0.20894539654133015	0.08107756942956698
kd_f_2_pm	-1.49453738992834	0.018528403356093032
kd_f_3_pm	-0.6903334937914476	0.08104022296838799
kd_f_12_pm	-1.2239512467427742	0.09607516493457836
kd_f_13_pm	-1.9096760377681599	0.042218269622335935
kd_f_23_pm	-1.0763511837344568	0.07988241149679419
kd_f_123_pm	-0.5019957612802708	0.08448934117373644
vol_EC	-0.47222293654592584	0.08191574444692849
A_PM	1.178175326926488	0.07330589606153823
vol_CP	1.8510456018377046	0.06818444188683662
A_END	1.3871490478786903	0.07961674784121472
A_ER	2.201331156333392	0.07225535448100556
A_GO	0.34153575731935015	0.0831578843883073
KR_degrade	0.17204939654573254	0.07418478846293268
R3	1.9964452850857066	0.010709633743874989
R7	1.9893501659481467	0.020243723617874996
APTer	0.9924386508642908	0.08208885077113352
APTgo	0.11630596924138516	0.07951508469691128
APTpm	-0.8611932667683451	0.07508339572779274
SiR3	2.0525986526492033	0.09248154872736843
SiR7	0.3073462352557257	0.035614662051666905

Table 8

D E. COLI NETWORK

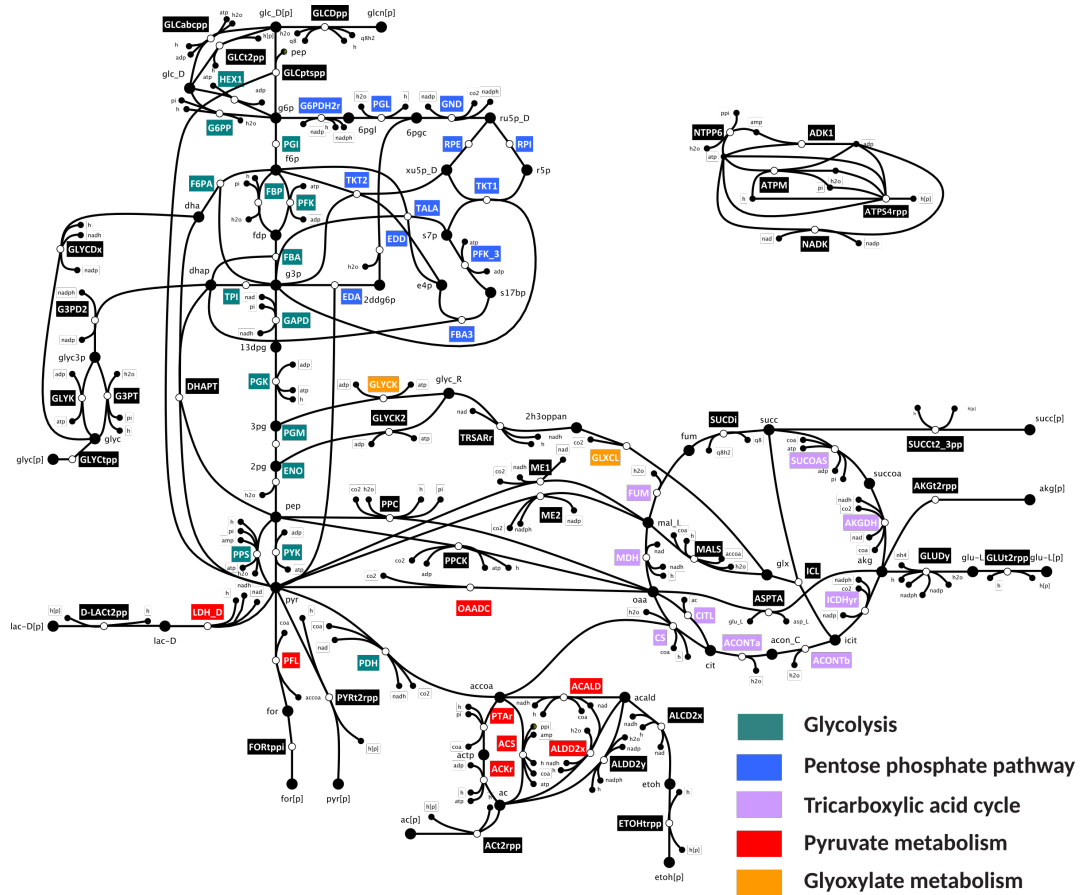


Figure 75: E.coli network diagram illustrating the reactions of the kinetic model. Diagram does not include all the reactions of the model. Overview of the reaction network, color-coding indicates to which subsystem a reaction belongs. Abbreviations are as follows: G/G - Glycolysis and Gluconeogenesis, PPP - Pentose Phosphate Pathway, TCA - Tricarboxylic Acid Cycle, Pyr - Pyruvate Metabolism, Glyox - Glyoxylate Metabolism. The reactions indicated in blue correspond to the Pentose phosphate pathway (PPP) subsystem, which was the focus of this study.

BIBLIOGRAPHY

- [1] N. Terranova, E. M. Tosca, E. Borella, E. Pesenti, M. Rocchetti, and P. Magni. "Modeling tumor growth inhibition and toxicity outcome after administration of anticancer agents in xenograft mice: A Dynamic Energy Budget (DEB) approach." en. In: *Journal of Theoretical Biology* 450 (Aug. 2018), pp. 1–14. ISSN: 0022-5193. DOI: [10.1016/j.jtbi.2018.04.012](https://doi.org/10.1016/j.jtbi.2018.04.012). URL: <http://www.sciencedirect.com/science/article/pii/S0022519318301723>.
- [2] Nagarjuna Nagaraj, Jacek R. Wisniewski, Tamar Geiger, Juergen Cox, Martin Kircher, Janet Kelso, Svante Pääbo, and Matthias Mann. "Deep proteome and transcriptome mapping of a human cancer cell line." eng. In: *Molecular Systems Biology* 7 (Nov. 2011), p. 548. ISSN: 1744-4292. DOI: [10.1038/msb.2011.81](https://doi.org/10.1038/msb.2011.81).
- [3] Longlong Liao, Kenli Li, Keqin Li, Canqun Yang, and Qi Tian. "A multiple kernel density clustering algorithm for incomplete datasets in bioinformatics." en. In: *BMC Systems Biology* 12.S6 (Nov. 2018). ISSN: 1752-0509. DOI: [10.1186/s12918-018-0630-6](https://doi.org/10.1186/s12918-018-0630-6). URL: <https://bmcsystbiol.biomedcentral.com/articles/10.1186/s12918-018-0630-6>.
- [4] Andrew S. Gaynor and Wilfred Chen. "Induced prodrug activation by conditional protein degradation." en. In: *Journal of Biotechnology* 260 (Oct. 2017), pp. 62–66. ISSN: 01681656. DOI: [10.1016/j.jbiotec.2017.09.005](https://doi.org/10.1016/j.jbiotec.2017.09.005). URL: <http://linkinghub.elsevier.com/retrieve/pii/S0168165617316450>.
- [5] Rasmus Agren, Adil Mardinoglu, Anna Asplund, Caroline Kampf, Mathias Uhlen, and Jens Nielsen. "Identification of anticancer drugs for hepatocellular carcinoma through personalized genome-scale metabolic modeling." In: *Molecular Systems Biology* 10.3 (Mar. 2014). ISSN: 1744-4292. DOI: [10.1002/msb.145122](https://doi.org/10.1002/msb.145122). URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4017677/>.
- [6] Kate Campbell, Jianye Xia, and Jens Nielsen. "The Impact of Systems Biology on Bioprocessing." In: *Trends in Biotechnology* 35.12 (Dec. 2017), pp. 1156–1168. ISSN: 0167-7799. DOI: [10.1016/j.tibtech.2017.08.011](https://doi.org/10.1016/j.tibtech.2017.08.011). URL: <http://www.sciencedirect.com/science/article/pii/S0167779917302305>.
- [7] Reinhard Renneberg, Viola Berkling, and Vanya Lorocho. "Chapter 4 - White Biotechnology: Cells as Synthetic Factories." In: *Biotechnology for Beginners (Second Edition)*. Boston: Academic Press, 2017, pp. 119–

163. ISBN: 978-0-12-801224-6. DOI: [10.1016/B978-0-12-801224-6.00004-7](https://doi.org/10.1016/B978-0-12-801224-6.00004-7). URL: <https://www.sciencedirect.com/science/article/pii/B9780128012246000047>.
- [8] A. J. Clark et al. "Biosynthetic human insulin in the treatment of diabetes. A double-blind crossover trial in established diabetic patients." eng. In: *Lancet (London, England)* 2.8294 (Aug. 1982), pp. 354–357. ISSN: 0140-6736. DOI: [10.1016/S0140-6736\(82\)90548-7](https://doi.org/10.1016/S0140-6736(82)90548-7).
- [9] D. V. Goeddel, D. G. Kleid, F. Bolivar, H. L. Heyneker, D. G. Yansura, R. Crea, T. Hirose, A. Kraszewski, K. Itakura, and A. D. Riggs. "Expression in *Escherichia coli* of chemically synthesized genes for human insulin." eng. In: *Proceedings of the National Academy of Sciences of the United States of America* 76.1 (Jan. 1979), pp. 106–110. ISSN: 0027-8424. DOI: [10.1073/pnas.76.1.106](https://doi.org/10.1073/pnas.76.1.106).
- [10] F. G. Banting, C. H. Best, J. B. Collip, W. R. Campbell, and A. A. Fletcher. "Pancreatic Extracts in the Treatment of Diabetes Mellitus: Preliminary Report." eng. In: *Canadian Medical Association Journal* 87.20 (Nov. 1962), pp. 1062–1067. ISSN: 0008-4409.
- [11] Benjamin Leader, Quentin J. Baca, and David E. Golan. "Protein therapeutics: a summary and pharmacological classification." en. In: *Nature Reviews Drug Discovery* 7.1 (Jan. 2008), pp. 21–39. ISSN: 1474-1784. DOI: [10.1038/nrd2399](https://doi.org/10.1038/nrd2399). URL: <https://www.nature.com/articles/nrd2399>.
- [12] James T. Yurkovich, Aarash Bordbar, Ólafur E. Sigurjónsson, and Bernhard O. Palsson. "Systems biology as an emerging paradigm in transfusion medicine." In: *BMC Systems Biology* 12 (Mar. 2018), p. 31. ISSN: 1752-0509. DOI: [10.1186/s12918-018-0558-x](https://doi.org/10.1186/s12918-018-0558-x). URL: <https://doi.org/10.1186/s12918-018-0558-x>.
- [13] M. Sturrock, I. S. Miller, G. Kang, N. Hannis Arba'ie, A. C. O'Farrell, A. Barat, G. Marston, P. L. Coletta, A. T. Byrne, and J. H. Prehn. "Anti-angiogenic drug scheduling optimisation with application to colorectal cancer." eng. In: *Scientific Reports* 8.1 (2018), p. 11182. ISSN: 2045-2322. DOI: [10.1038/s41598-018-29318-5](https://doi.org/10.1038/s41598-018-29318-5).
- [14] Gibin G. Powathil, Douglas J. A. Adamson, and Mark A. J. Chaplain. "Towards predicting the response of a solid tumour to chemotherapy and radiotherapy treatments: clinical insights from a computational model." eng. In: *PLoS computational biology* 9.7 (2013), e1003120. ISSN: 1553-7358. DOI: [10.1371/journal.pcbi.1003120](https://doi.org/10.1371/journal.pcbi.1003120).
- [15] Mark Chaplain and Alexander Anderson. "Mathematical Modelling of Tumour-induced Angiogenesis: Network Growth and Structure." en. In: *Angiogenesis in Brain Tumors*. Ed. by Matthias Kirsch and Peter McL. Black. Cancer Treatment and Research. Boston, MA: Springer

- US, 2004, pp. 51–75. ISBN: 978-1-4419-8871-3. DOI: [10.1007/978-1-4419-8871-3_3](https://doi.org/10.1007/978-1-4419-8871-3_3). URL: https://doi.org/10.1007/978-1-4419-8871-3_3.
- [16] Peter Hinow et al. “A spatial model of tumor-host interaction: application of chemotherapy.” eng. In: *Mathematical biosciences and engineering: MBE* 6.3 (July 2009), pp. 521–546. ISSN: 1547-1063. DOI: [10.3934/mbe.2009.6.521](https://doi.org/10.3934/mbe.2009.6.521).
- [17] Magdalena San Roman and Andreas Wagner. “An enormous potential for niche construction through bacterial cross-feeding in a homogeneous environment.” en. In: *PLOS Computational Biology* 14.7 (July 2018), e1006340. ISSN: 1553-7358. DOI: [10.1371/journal.pcbi.1006340](https://doi.org/10.1371/journal.pcbi.1006340). URL: <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1006340>.
- [18] Siu H. J. Chan, Elliot S. Friedman, Gary D. Wu, and Costas D. Maranas. “Predicting the Longitudinally and Radially Varying Gut Microbiota Composition using Multi-Scale Microbial Metabolic Modeling.” en. In: *Processes* 7.7 (July 2019), p. 394. DOI: [10.3390/pr7070394](https://doi.org/10.3390/pr7070394). URL: <https://www.mdpi.com/2227-9717/7/7/394>.
- [19] Won Jun Kim, Hyun Uk Kim, and Sang Yup Lee. “Current state and applications of microbial genome-scale metabolic models.” en. In: *Current Opinion in Systems Biology*. Regulatory and metabolic networks • Cancer and systemic diseases 2 (Apr. 2017), pp. 10–18. ISSN: 2452-3100. DOI: [10.1016/j.coisb.2017.03.001](https://doi.org/10.1016/j.coisb.2017.03.001). URL: <http://www.sciencedirect.com/science/article/pii/S2452310017300483>.
- [20] Maxime Durot, Pierre-Yves Bourguignon, and Vincent Schachter. “Genome-scale models of bacterial metabolism: reconstruction and applications.” eng. In: *FEMS microbiology reviews* 33.1 (Jan. 2009), pp. 164–190. ISSN: 0168-6445. DOI: [10.1111/j.1574-6976.2008.00146.x](https://doi.org/10.1111/j.1574-6976.2008.00146.x).
- [21] Edward J. O’Brien, Jonathan M. Monk, and Bernhard O. Palsson. “Using Genome-scale Models to Predict Biological Capabilities.” eng. In: *Cell* 161.5 (May 2015), pp. 971–987. ISSN: 1097-4172. DOI: [10.1016/j.cell.2015.05.019](https://doi.org/10.1016/j.cell.2015.05.019).
- [22] Michael T. Forrester, Douglas T. Hess, J. Will Thompson, Rainbo Hultman, M. Arthur Moseley, Jonathan S. Stamler, and Patrick J. Casey. “Site-specific analysis of protein S-acylation by resin-assisted capture.” en. In: *Journal of Lipid Research* 52.2 (Feb. 2011). Publisher: American Society for Biochemistry and Molecular Biology, pp. 393–398. ISSN: 0022-2275, 1539-7262. DOI: [10.1194/jlr.D011106](https://doi.org/10.1194/jlr.D011106). URL: <http://www.jlr.org/content/52/2/393>.

- [23] Patrick A. Sandoz and F. Gisou van der Goot. "How many lives does CLIMP-63 have?" en. In: *Biochemical Society Transactions* 43.2 (Apr. 2015), pp. 222–228. ISSN: 0300-5127, 1470-8752. DOI: [10.1042/BST20140272](https://doi.org/10.1042/BST20140272). URL: <http://www.biochemsoctrans.org/content/43/2/222>.
- [24] Sylvain Prigent et al. "Meneco, a Topology-Based Gap-Filling Tool Applicable to Degraded Genome-Wide Metabolic Networks." en. In: *PLOS Computational Biology* 13.1 (Jan. 2017), e1005276. ISSN: 1553-7358. DOI: [10.1371/journal.pcbi.1005276](https://doi.org/10.1371/journal.pcbi.1005276). URL: <http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005276>.
- [25] Ines Thiele and Bernhard Ø. Palsson. "A protocol for generating a high-quality genome-scale metabolic reconstruction." In: *Nature protocols* 5.1 (2010), pp. 93–121. ISSN: 1754-2189. DOI: [10.1038/nprot.2009.203](https://doi.org/10.1038/nprot.2009.203). URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3125167/>.
- [26] Christopher S. Henry, Matthew DeJongh, Aaron A. Best, Paul M. Frybarger, Ben Linsay, and Rick L. Stevens. "High-throughput generation, optimization and analysis of genome-scale metabolic models." en. In: *Nature Biotechnology* 28.9 (Sept. 2010). Number: 9 Publisher: Nature Publishing Group, pp. 977–982. ISSN: 1546-1696. DOI: [10.1038/nbt.1672](https://doi.org/10.1038/nbt.1672). URL: <https://www.nature.com/articles/nbt.1672>.
- [27] Andrew R. Joyce and Bernhard Ø Palsson. "Predicting gene essentiality using genome-scale in silico models." eng. In: *Methods in Molecular Biology (Clifton, N.J.)* 416 (2008), pp. 433–457. ISSN: 1064-3745. DOI: [10.1007/978-1-59745-321-9_30](https://doi.org/10.1007/978-1-59745-321-9_30).
- [28] Changdai Gu, Gi Bae Kim, Won Jun Kim, Hyun Uk Kim, and Sang Yup Lee. "Current status and applications of genome-scale metabolic models." In: *Genome Biology* 20.1 (June 2019), p. 121. ISSN: 1474-760X. DOI: [10.1186/s13059-019-1730-3](https://doi.org/10.1186/s13059-019-1730-3). URL: <https://doi.org/10.1186/s13059-019-1730-3>.
- [29] Matthew R Long, Wai Kit Ong, and Jennifer L Reed. "Computational methods in metabolic engineering for strain design." In: *Current Opinion in Biotechnology. Systems biology • Nanobiotechnology* 34 (Aug. 2015), pp. 135–141. ISSN: 0958-1669. DOI: [10.1016/j.copbio.2014.12.019](https://doi.org/10.1016/j.copbio.2014.12.019). URL: <http://www.sciencedirect.com/science/article/pii/S0958166914002341>.
- [30] A. Varma and B. O. Palsson. "Stoichiometric flux balance models quantitatively predict growth and metabolic by-product secretion in wild-type Escherichia coli W3110." en. In: *Applied and Environmental*

- Microbiology* 60.10 (Oct. 1994), pp. 3724–3731. ISSN: 0099-2240, 1098-5336. URL: <https://aem.asm.org/content/60/10/3724>.
- [31] Christopher S. Henry, Linda J. Broadbelt, and Vassily Hatzimanikatis. “Thermodynamics-Based Metabolic Flux Analysis.” en. In: *Biophysical Journal* 92.5 (Mar. 2007), pp. 1792–1805. ISSN: 00063495. DOI: [10.1529/biophysj.106.093138](https://doi.org/10.1529/biophysj.106.093138). URL: <http://linkinghub.elsevier.com/retrieve/pii/S0006349507709876>.
- [32] A. Kiparissides and V. Hatzimanikatis. “Thermodynamics-based Metabolite Sensitivity Analysis in metabolic networks.” In: *Metabolic Engineering* 39 (Jan. 2017), pp. 117–127. ISSN: 1096-7176. DOI: [10.1016/j.ymben.2016.11.006](https://doi.org/10.1016/j.ymben.2016.11.006). URL: <http://www.sciencedirect.com/science/article/pii/S109671761630218X>.
- [33] Meric Ataman and Vassily Hatzimanikatis. “Heading in the right direction: thermodynamics-based network analysis and pathway engineering.” In: *Current Opinion in Biotechnology. Pathway engineering* 36 (Dec. 2015), pp. 176–182. ISSN: 0958-1669. DOI: [10.1016/j.copbio.2015.08.021](https://doi.org/10.1016/j.copbio.2015.08.021). URL: <http://www.sciencedirect.com/science/article/pii/S0958166915001202>.
- [34] Pierre Salvy, Georgios Fengos, Meric Ataman, Thomas Pathier, Keng C. Soh, and Vassily Hatzimanikatis. “pyTFA and matTFA: a Python package and a Matlab toolbox for Thermodynamics-based Flux Analysis.” en. In: *Bioinformatics* 35.1 (Jan. 2019), pp. 167–169. ISSN: 1367-4803. DOI: [10.1093/bioinformatics/bty499](https://doi.org/10.1093/bioinformatics/bty499). URL: <https://academic.oup.com/bioinformatics/article/35/1/167/5047753>.
- [35] Ljubisa Miskovic and Vassily Hatzimanikatis. “Production of biofuels and biochemicals: in need of an ORACLE.” English. In: *Trends in Biotechnology* 28.8 (Aug. 2010), pp. 391–397. ISSN: 0167-7799, 1879-3096. DOI: [10.1016/j.tibtech.2010.05.003](https://doi.org/10.1016/j.tibtech.2010.05.003). URL: [http://www.cell.com/trends/biotechnology/abstract/S0167-7799\(10\)00084-3](http://www.cell.com/trends/biotechnology/abstract/S0167-7799(10)00084-3).
- [36] Anirikh Chakrabarti, Ljubisa Miskovic, Keng Cher Soh, and Vassily Hatzimanikatis. “Towards kinetic modeling of genome-scale metabolic networks without sacrificing stoichiometric, thermodynamic and physiological constraints.” en. In: *Biotechnology Journal* 8.9 (2013), pp. 1043–1057. ISSN: 1860-7314. DOI: [10.1002/biot.201300091](https://doi.org/10.1002/biot.201300091). URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/biot.201300091>.
- [37] Christine Reder. “Metabolic control theory: A structural approach.” en. In: *Journal of Theoretical Biology* 135.2 (Nov. 1988), pp. 175–201. ISSN: 00225193. DOI: [10.1016/S0022-5193\(88\)80073-0](https://doi.org/10.1016/S0022-5193(88)80073-0). URL: <https://linkinghub.elsevier.com/retrieve/pii/S0022519388800730>.

- [38] L Wang and V Hatzimanikatis. "Metabolic engineering under uncertainty. I: Framework development." en. In: *Metabolic Engineering* 8.2 (Mar. 2006), pp. 133–141. ISSN: 10967176. DOI: [10.1016/j.ymben.2005.11.003](https://doi.org/10.1016/j.ymben.2005.11.003). URL: <http://linkinghub.elsevier.com/retrieve/pii/S1096717605000996>.
- [39] Vassily Hatzimanikatis. "Nonlinear Metabolic Control Analysis." In: *Metabolic Engineering* 1.1 (Jan. 1999), pp. 75–87. ISSN: 1096-7176. DOI: [10.1006/mben.1998.0108](https://doi.org/10.1006/mben.1998.0108). URL: <http://www.sciencedirect.com/science/article/pii/S1096717698901082>.
- [40] Liqing Wang, Inanç Birol, and Vassily Hatzimanikatis. "Metabolic Control Analysis under Uncertainty: Framework Development and Case Studies." In: *Biophysical Journal* 87.6 (Dec. 1, 2004), pp. 3750–3763. ISSN: 0006-3495. DOI: [10.1529/biophysj.104.048090](https://doi.org/10.1529/biophysj.104.048090). URL: <http://www.sciencedirect.com/science/article/pii/S0006349504738436>.
- [41] Vassily Hatzimanikatis and Liqing Wang. "The systems engineering of cellular processes." In: *Computer Aided Chemical Engineering*. Ed. by W. Marquardt and C. Pantelides. Vol. 21. 16th European Symposium on Computer Aided Process Engineering and 9th International Symposium on Process Systems Engineering. Elsevier, Jan. 2006, pp. 71–80. DOI: [10.1016/S1570-7946\(06\)80028-3](https://doi.org/10.1016/S1570-7946(06)80028-3). URL: <http://www.sciencedirect.com/science/article/pii/S1570794606800283>.
- [42] Vassily Hatzimanikatis and James E. Bailey. "MCA Has More to Say." In: *Journal of Theoretical Biology* 182.3 (Oct. 1996), pp. 233–242. ISSN: 0022-5193. DOI: [10.1006/jtbi.1996.0160](https://doi.org/10.1006/jtbi.1996.0160). URL: <http://www.sciencedirect.com/science/article/pii/S0022519396901605>.
- [43] Tiziano Dallavilla, Laurence Abrami, Patrick A. Sandoz, Georgios Savoglidis, Vassily Hatzimanikatis, and F. Gisou van der Goot. "Model-Driven Understanding of Palmitoylation Dynamics: Regulated Acylation of the Endoplasmic Reticulum Chaperone Calnexin." In: *PLOS Comput Biol* 12.2 (Feb. 2016), e1004774. ISSN: 1553-7358. DOI: [10.1371/journal.pcbi.1004774](https://doi.org/10.1371/journal.pcbi.1004774). URL: <http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1004774>.
- [44] Asvin KK Lakkaraju, Laurence Abrami, Thomas Lemmin, Sanja Blaskovic, Béatrice Kunz, Akio Kihara, Matteo Dal Peraro, and Françoise Gisou van der Goot. "Palmitoylated calnexin is a key component of the ribosome–translocon complex." In: *The EMBO Journal* 31.7 (Apr. 2012), pp. 1823–1835. ISSN: 0261-4189. DOI: [10.1038/emboj.2012.15](https://doi.org/10.1038/emboj.2012.15). URL: <https://www.embopress.org/doi/full/10.1038/emboj.2012.15>.
- [45] Desmond J. Higham. "Modeling and Simulating Chemical Reactions." en. In: *SIAM Review* 50.2 (Jan. 2008), pp. 347–368. ISSN: 0036-1445, 1095-7200. DOI: [10.1137/060666457](https://doi.org/10.1137/060666457). URL: <http://epubs.siam.org/doi/10.1137/060666457>.

- [46] Daniel T Gillespie. "A general method for numerically simulating the stochastic time evolution of coupled chemical reactions." en. In: *Journal of Computational Physics* 22.4 (Dec. 1976), pp. 403–434. ISSN: 0021-9991. DOI: [10.1016/0021-9991\(76\)90041-3](https://doi.org/10.1016/0021-9991(76)90041-3). URL: <http://www.sciencedirect.com/science/article/pii/0021999176900413>.
- [47] Daniel T. Gillespie and Daniel T. Gillespie. *Exact Stochastic Simulation of Coupled Chemical Reactions*.
- [48] Péter Érdi and János Tóth. *Mathematical Models of Chemical Reactions: Theory and Applications of Deterministic and Stochastic Models*. en. Google-Books-ID: iDu8AAAAIAAJ. Manchester University Press, 1989. ISBN: 978-0-7190-2208-1.
- [49] Dmitry Yu. Murzin and Tapio Salmi. "Chapter 6 - Enzymatic Kinetics." In: *Catalytic Kinetics (Second Edition)*. Amsterdam: Elsevier, 2016, pp. 281–343. ISBN: 978-0-444-63753-6. DOI: [10.1016/B978-0-444-63753-6.00006-3](https://doi.org/10.1016/B978-0-444-63753-6.00006-3). URL: <https://www.sciencedirect.com/science/article/pii/B9780444637536000063>.
- [50] Lee A. Segel. "On the validity of the steady state assumption of enzyme kinetics." en. In: *Bulletin of Mathematical Biology* 50.6 (Jan. 1988), pp. 579–593. ISSN: 0092-8240. DOI: [10.1016/S0092-8240\(88\)80057-0](https://doi.org/10.1016/S0092-8240(88)80057-0). URL: <http://www.sciencedirect.com/science/article/pii/S0092824088800570>.
- [51] Kenneth A. Johnson and Roger S. Goody. *The Original Michaelis Constant: Translation of the 1913 Michaelis–Menten Paper*. Sept. 9, 2011. DOI: [10.1021/bi201284u](https://doi.org/10.1021/bi201284u). URL: <https://pubs.acs.org/doi/suppl/10.1021/bi201284u>.
- [52] George Edward Briggs and John Burdon Sanderson Haldane. "A Note on the Kinetics of Enzyme Action." In: *Biochemical Journal* 19.2 (1925), pp. 338–339. ISSN: 0264-6021. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1259181/>.
- [53] Morten Gram Pedersen, Alberto M. Bersani, Enrico Bersani, and Giuliana Cortese. "The total quasi-steady-state approximation for complex enzyme reactions." In: *Mathematics and Computers in Simulation*. 5th Vienna International Conference on Mathematical Modelling/-Workshop on Scientific Computing in Electronic Engineering of the 2006 International Conference on Computational Science/Structural Dynamical Systems: Computational Aspects 79.4 (Dec. 2008), pp. 1010–1019. ISSN: 0378-4754. DOI: [10.1016/j.matcom.2008.02.009](https://doi.org/10.1016/j.matcom.2008.02.009). URL: <http://www.sciencedirect.com/science/article/pii/S0378475408001109>.
- [54] A Tzafriri. "Michaelis–Menten kinetics at high enzyme concentrations." en. In: *Bulletin of Mathematical Biology* 65.6 (Nov. 2003), pp. 1111–1129. ISSN: 00928240. DOI: [10.1016/S0092-8240\(03\)00059-4](https://doi.org/10.1016/S0092-8240(03)00059-4). URL: [http://link.springer.com/10.1016/S0092-8240\(03\)00059-4](http://link.springer.com/10.1016/S0092-8240(03)00059-4).

- [55] Sanja Blaskovic, Mathieu Blanc, and F. Gisou van der Goot. "What does S-palmitoylation do to membrane proteins?" In: *The FEBS journal* 280.12 (June 2013), pp. 2766–2774. ISSN: 1742-4658. DOI: [10.1111/febs.12263](https://doi.org/10.1111/febs.12263).
- [56] Luke H. Chamberlain and Michael J. Shipston. "The Physiology of Protein S-acylation." In: *Physiological Reviews* 95.2 (Apr. 2015), pp. 341–376. ISSN: 0031-9333. DOI: [10.1152/physrev.00032.2014](https://doi.org/10.1152/physrev.00032.2014). URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4551212/>.
- [57] Brent R. Martin and Benjamin F. Cravatt. "Large-scale profiling of protein palmitoylation in mammalian cells." In: *Nature Methods* 6.2 (Feb. 2009), pp. 135–138. ISSN: 1548-7091. DOI: [10.1038/nmeth.1293](https://doi.org/10.1038/nmeth.1293). URL: <http://www.nature.com/nmeth/journal/v6/n2/full/nmeth.1293.html>.
- [58] Fiona Plain, Jacqueline Howie, Jennifer Kennedy, Elaine Brown, Michael J. Shattock, Niall J. Fraser, and William Fuller. "Control of protein palmitoylation by regulating substrate recruitment to a zDHHC-protein acyltransferase." en. In: *Communications Biology* 3.1 (July 2020). Number: 1 Publisher: Nature Publishing Group, pp. 1–10. ISSN: 2399-3642. DOI: [10.1038/s42003-020-01145-3](https://doi.org/10.1038/s42003-020-01145-3). URL: <https://www.nature.com/articles/s42003-020-01145-3>.
- [59] Amy F. Roth, Junmei Wan, Aaron O. Bailey, Beimeng Sun, Jason A. Kuchar, William N. Green, Brett S. Phinney, John R. Yates, and Nicholas G. Davis. "Global Analysis of Protein Palmitoylation in Yeast." en. In: *Cell* 125.5 (June 2006), pp. 1003–1013. ISSN: 0092-8674. DOI: [10.1016/j.cell.2006.03.042](https://doi.org/10.1016/j.cell.2006.03.042). URL: <http://www.sciencedirect.com/science/article/pii/S0092867406005150>.
- [60] Mathieu Blanc, Sanja Blaskovic, and F. Gisou van der Goot. "Palmitoylation, pathogens and their host." en. In: *Biochemical Society Transactions* 41.1 (Feb. 2013), pp. 84–88. ISSN: 0300-5127, 1470-8752. DOI: [10.1042/BST20120337](https://doi.org/10.1042/BST20120337). URL: <http://www.biochemsoctrans.org/content/41/1/84>.
- [61] Sanja Blaskovic, Alexander Adibekian, Mathieu Blanc, and Gisou F. van der Goot. "Mechanistic effects of protein palmitoylation and the cellular consequences thereof." In: *Chemistry and Physics of Lipids* 180 (May 2014), pp. 44–52. ISSN: 0009-3084. DOI: [10.1016/j.chemphyslip.2014.02.001](https://doi.org/10.1016/j.chemphyslip.2014.02.001). URL: <http://www.sciencedirect.com/science/article/pii/S0009308414000127>.
- [62] Justyna Sobocińska, Paula Roszczenko-Jasińska, Anna Ciesielska, and Katarzyna Kwiatkowska. "Protein Palmitoylation and Its Role in Bacterial and Viral Infections." In: *Frontiers in Immunology* 8 (Jan. 2018). ISSN: 1664-3224. DOI: [10.3389/fimmu.2017.02003](https://doi.org/10.3389/fimmu.2017.02003). URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5780409/>.

- [63] T. Putilina, P. Wong, and S. Gentleman. "The DHHC domain: A new highly conserved cysteine-rich motif." en. In: *Molecular and Cellular Biochemistry* 195.1-2 (May 1999), pp. 219–226. ISSN: 0300-8177, 1573-4919. DOI: [10.1023/A:1006932522197](https://doi.org/10.1023/A:1006932522197). URL: <https://link.springer.com/article/10.1023/A:1006932522197>.
- [64] Amy F. Roth, Ying Feng, Linyi Chen, and Nicholas G. Davis. "The yeast DHHC cysteine-rich domain protein Akr1p is a palmitoyl transferase." In: *The Journal of Cell Biology* 159.1 (Oct. 14, 2002), pp. 23–28. ISSN: 0021-9525. DOI: [10.1083/jcb.200206120](https://doi.org/10.1083/jcb.200206120). URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2173492/>.
- [65] Wei Yang, Dolores Di Vizio, Marc Kirchner, Hanno Steen, and Michael R. Freeman. "Proteome scale characterization of human S-acylated proteins in lipid raft-enriched and non-raft membranes." In: *Molecular & cellular proteomics: MCP* 9.1 (Jan. 2010), pp. 54–70. ISSN: 1535-9484. DOI: [10.1074/mcp.M800448-MCP200](https://doi.org/10.1074/mcp.M800448-MCP200).
- [66] C. Y. Huang and J. E. Ferrell. "Ultrasensitivity in the mitogen-activated protein kinase cascade." In: *Proceedings of the National Academy of Sciences* 93.19 (Sept. 17, 1996), pp. 10078–10083. ISSN: 0027-8424, 1091-6490. URL: <http://www.pnas.org/content/93/19/10078>.
- [67] Robert H. Rice and Gary E. Means. "Radioactive Labeling of Proteins in Vitro." en. In: *Journal of Biological Chemistry* 246.3 (Feb. 1971). Publisher: American Society for Biochemistry and Molecular Biology, pp. 831–832. ISSN: 0021-9258, 1083-351X. URL: <http://www.jbc.org/content/246/3/831>.
- [68] Frederick D. Tsai, Joseph P. Wynne, Ian M. Ahearn, and Mark R. Philips. "Metabolic labeling of Ras with tritiated palmitate to monitor palmitoylation and depalmitoylation." In: *Methods in molecular biology (Clifton, N.J.)* 1120 (2014), pp. 33–41. ISSN: 1064-3745. DOI: [10.1007/978-1-62703-791-4_3](https://doi.org/10.1007/978-1-62703-791-4_3). URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4065800/>.
- [69] PubChem. *Sulfur-35*. en. URL: <https://pubchem.ncbi.nlm.nih.gov/compound/61783>.
- [70] Gary L. Miessler. *Inorganic chemistry*. eng. Upper Saddle River, N.J. : Pearson Education, 2004. ISBN: 978-0-13-035471-6 978-0-13-120198-9. URL: <http://archive.org/details/inorganicchemist03edmies>.
- [71] Maya Shamir, Yinon Bar-On, Rob Phillips, and Ron Milo. "Snap-Shot: Timescales in Cell Biology." en. In: *Cell* 164.6 (Mar. 2016), 1302–1302.e1. ISSN: 00928674. DOI: [10.1016/j.cell.2016.02.058](https://doi.org/10.1016/j.cell.2016.02.058). URL: <https://linkinghub.elsevier.com/retrieve/pii/S0092867416302082>.

- [72] William F. Scherer, Jerome T. Syverton, and George O. Gey. "STUDIES ON THE PROPAGATION IN VITRO OF POLIOMYELITIS VIRUSES." In: *The Journal of Experimental Medicine* 97.5 (May 1953), pp. 695–710. ISSN: 0022-1007. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2136303/>.
- [73] Pamela Bulzomi, Paola Galluzzo, Alessandro Bolli, Stefano Leone, Filippo Acconcia, and Maria Marino. "The pro-apoptotic effect of quercetin in cancer cell lines requires ERbeta-dependent signals." en. In: *Journal of Cellular Physiology* 227.5 (2012). _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/jcp.22917>. pp. 1891–1898. ISSN: 1097-4652. DOI: <https://doi.org/10.1002/jcp.22917>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/jcp.22917>.
- [74] Patrick A. Sandoz et al. "The architecture of the endoplasmic reticulum is regulated by the reversible lipid modification of the shaping protein CLIMP-63." en. In: *bioRxiv* (Oct. 2018), p. 431106. DOI: [10.1101/431106](https://doi.org/10.1101/431106). URL: <https://www.biorxiv.org/content/early/2018/10/01/431106>.
- [75] Emily M. Lynes et al. "Palmitoylation is the switch that assigns calnexin to quality control or ER Ca²⁺ signaling." en. In: *Journal of Cell Science* 126.17 (Sept. 2013). Publisher: The Company of Biologists Ltd Section: Research Article, pp. 3893–3903. ISSN: 0021-9533, 1477-9137. DOI: [10.1242/jcs.125856](https://doi.org/10.1242/jcs.125856). URL: <https://jcs.biologists.org/content/126/17/3893>.
- [76] A. Goldbeter and D. E. Koshland. "An amplified sensitivity arising from covalent modification in biological systems." en. In: *Proceedings of the National Academy of Sciences* 78.11 (Nov. 1981). Publisher: National Academy of Sciences Section: Research Article, pp. 6840–6844. ISSN: 0027-8424, 1091-6490. DOI: [10.1073/pnas.78.11.6840](https://doi.org/10.1073/pnas.78.11.6840). URL: <https://www.pnas.org/content/78/11/6840>.
- [77] Richard J. Orton, Oliver E. Sturm, Vladislav Vyshemirsky, Muffy Calder, David R. Gilbert, and Walter Kolch. "Computational modelling of the receptor-tyrosine-kinase-activated MAPK pathway." In: *Biochemical Journal* 392.Pt 2 (Dec. 2005), pp. 249–261. ISSN: 0264-6021. DOI: [10.1042/BJ20050908](https://doi.org/10.1042/BJ20050908). URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1316260/>.
- [78] Walter Kolch, Muffy Calder, and David Gilbert. "When kinases meet mathematics: the systems biology of MAPK signalling." en. In: *FEBS Letters* 579.8 (2005). _eprint: <https://febs.onlinelibrary.wiley.com/doi/pdf/10.1016/j.febslet.2005.02.002>. pp. 1891–1895. ISSN: 1873-3468. DOI: [10.1016/j.febslet.2005.02.002](https://doi.org/10.1016/j.febslet.2005.02.002). URL: <https://febs.onlinelibrary.wiley.com/doi/abs/10.1016/j.febslet.2005.02.002>.

- [79] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. "A fast and elitist multiobjective genetic algorithm: NSGA-II." en. In: *IEEE Transactions on Evolutionary Computation* 6.2 (Apr. 2002), pp. 182–197. ISSN: 1089778X. DOI: [10.1109/4235.996017](https://doi.org/10.1109/4235.996017). URL: <http://ieeexplore.ieee.org/document/996017/>.
- [80] Bertrand Iooss and Paul Lemaître. "A Review on Global Sensitivity Analysis Methods." In: *Uncertainty Management in Simulation-Optimization of Complex Systems*. Ed. by Gabriella Dellino and Carlo Meloni. Vol. 59. Boston, MA: Springer US, 2015, pp. 101–122. ISBN: 978-1-4899-7546-1 978-1-4899-7547-8. URL: http://link.springer.com/10.1007/978-1-4899-7547-8_5.
- [81] Arvind Varma, Massimo Morbidelli, and Hua Wu. *Parametric Sensitivity in Chemical Systems*. Cambridge ; New York: Cambridge University Press, Sept. 15, 2005. 360 pp. ISBN: 978-0-521-01984-2.
- [82] A. Saltelli, Marco Ratto, Terry Andres, Francesca Campolongo, Jessica Cariboni, Debora Gatelli, Michaela Saisana, and Stefano Tarantola. *Global Sensitivity Analysis: The Primer*. 1 edition. Chichester, England ; Hoboken, NJ: Wiley-Interscience, Feb. 11, 2008. 304 pp. ISBN: 978-0-470-05997-5.
- [83] A. Schweizer, M. Ericsson, T. Bächli, G. Griffiths, and H. P. Hauri. "Characterization of a novel 63 kDa membrane protein. Implications for the organization of the ER-to-Golgi pathway." eng. In: *Journal of Cell Science* 104 (Pt 3) (Mar. 1993), pp. 671–683. ISSN: 0021-9533.
- [84] D. I. Mundy and G. Warren. "Mitosis and inhibition of intracellular transport stimulate palmitoylation of a 62-kD protein." en. In: *Journal of Cell Biology* 116.1 (Jan. 1992). Publisher: The Rockefeller University Press, pp. 135–146. ISSN: 0021-9525. DOI: [10.1083/jcb.116.1.135](https://doi.org/10.1083/jcb.116.1.135). URL: <https://rupress.org/jcb/article/116/1/135/14296/Mitosis-and-inhibition-of-intracellular-transport>.
- [85] Tahir M. Razzaq, Rosemary Bass, David J. Vines, Finn Werner, Simon A. Whawell, and Vincent Ellis. "Functional Regulation of Tissue Plasminogen Activator on the Surface of Vascular Smooth Muscle Cells by the Type-II Transmembrane Protein p63 (CKAP4)." en. In: *Journal of Biological Chemistry* 278.43 (Oct. 2003). Publisher: American Society for Biochemistry and Molecular Biology, pp. 42679–42685. ISSN: 0021-9258, 1083-351X. DOI: [10.1074/jbc.M305695200](https://doi.org/10.1074/jbc.M305695200). URL: <http://www.jbc.org/content/278/43/42679>.
- [86] Nisha Gupta, Yefim Manevich, Altaf S. Kazi, Jian-Qin Tao, Aron B. Fisher, and Sandra R. Bates. "Identification and characterization of p63 (CKAP4/ERGIC-63/CLIMP-63), a surfactant protein A binding protein, on type II pneumocytes." In: *American Journal of Physiology-Lung Cellular and Molecular Physiology* 291.3 (Sept. 2006). Publisher:

- American Physiological Society, pp. L436–L446. ISSN: 1040-0605. DOI: [10.1152/ajplung.00415.2005](https://doi.org/10.1152/ajplung.00415.2005). URL: <https://journals.physiology.org/doi/full/10.1152/ajplung.00415.2005>.
- [87] Thomas P. Conrads, Gillian M. Tocci, Brian L. Hood, Chen-Ou Zhang, Li Guo, Kristopher R. Koch, Christopher J. Michejda, Timothy D. Veenstra, and Susan K. Keay. “CKAP4/p63 Is a Receptor for the Frizzled-8 Protein-related Antiproliferative Factor from Interstitial Cystitis Patients.” en. In: *Journal of Biological Chemistry* 281.49 (Dec. 2006). Publisher: American Society for Biochemistry and Molecular Biology, pp. 37836–37843. ISSN: 0021-9258, 1083-351X. DOI: [10.1074/jbc.M604581200](https://doi.org/10.1074/jbc.M604581200). URL: <http://www.jbc.org/content/281/49/37836>.
- [88] Chen-min Sun, Jiang Geng, Yang Yan, Xudong Yao, and Min Liu. “Overexpression of CKAP4 is Associated with Poor Prognosis in Clear Cell Renal Cell Carcinoma and Functions via Cyclin B Signaling.” In: *Journal of Cancer* 8.19 (Oct. 2017), pp. 4018–4026. ISSN: 1837-9664. DOI: [10.7150/jca.21226](https://doi.org/10.7150/jca.21226). URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5706004/>.
- [89] Min-hong Li et al. “Expression of cytoskeleton-associated protein 4 is related to lymphatic metastasis and indicates prognosis of intra-hepatic cholangiocarcinoma patients after surgery resection.” en. In: *Cancer Letters* 337.2 (Sept. 2013), pp. 248–253. ISSN: 0304-3835. DOI: [10.1016/j.canlet.2013.05.003](https://doi.org/10.1016/j.canlet.2013.05.003). URL: <https://www.sciencedirect.com/science/article/pii/S0304383513003716>.
- [90] Shuang-Xi Li et al. “Prognostic significance of cytoskeleton-associated membrane protein 4 and its palmitoyl acyltransferase DHHC2 in hepatocellular carcinoma.” en. In: *Cancer* 120.10 (2014). _eprint: [https://doi.org/10.1002/cncr.28593](https://acsjournals.onlinelibrary.wiley.com/doi/abs/10.1002/cncr.28593). URL: <https://acsjournals.onlinelibrary.wiley.com/doi/abs/10.1002/cncr.28593>.
- [91] Feng Zhao, Tongyun Chen, and Nan Jiang. “CDR1as/miR-7/CKAP4 axis contributes to the pathogenesis of abdominal aortic aneurysm by regulating the proliferation and apoptosis of primary vascular smooth muscle cells.” In: *Experimental and Therapeutic Medicine* 19.6 (June 2020). Publisher: Spandidos Publications, pp. 3760–3766. ISSN: 1792-0981. DOI: [10.3892/etm.2020.8622](https://doi.org/10.3892/etm.2020.8622). URL: <http://www.spandidos-publications.com/10.3892/etm.2020.8622/abstract>.
- [92] Zsuzsanna Dosztányi, Veronika Csizmok, Peter Tompa, and István Simon. “IUPred: web server for the prediction of intrinsically unstructured regions of proteins based on estimated energy content.” eng. In: *Bioinformatics (Oxford, England)* 21.16 (Aug. 2005), pp. 3433–3434. ISSN: 1367-4803. DOI: [10.1093/bioinformatics/bti541](https://doi.org/10.1093/bioinformatics/bti541).

- [93] Leonard J. Foster, Carmen L. de Hoog, Yanling Zhang, Yong Zhang, Xiaohui Xie, Vamsi K. Mootha, and Matthias Mann. "A Mammalian Organelle Map by Protein Correlation Profiling." en. In: *Cell* 125.1 (Apr. 2006), pp. 187–199. ISSN: 0092-8674. DOI: [10.1016/j.cell.2006.03.022](https://doi.org/10.1016/j.cell.2006.03.022). URL: <http://www.sciencedirect.com/science/article/pii/S0092867406003692>.
- [94] Martin Beck, Alexander Schmidt, Johan Malmstroem, Manfred Claassen, Alessandro Ori, Anna Szymborska, Franz Herzog, Oliver Rinner, Jan Ellenberg, and Ruedi Aebersold. "The quantitative proteome of a human cell line." In: *Molecular Systems Biology* 7.1 (Jan. 2011). Publisher: John Wiley & Sons, Ltd, p. 549. ISSN: 1744-4292. DOI: [10.1038/msb.2011.82](https://doi.org/10.1038/msb.2011.82). URL: <https://www.embopress.org/doi/full/10.1038/msb.2011.82>.
- [95] Jun Zhang, Sonia L. Planey, Carolina Ceballos, Stanley M. Stevens, Susan K. Keay, and David A. Zacharias. "Identification of CKAP4/p63 as a Major Substrate of the Palmitoyl Acyltransferase DHHC2, a Putative Tumor Suppressor, Using a Novel Proteomics Method." In: *Molecular & Cellular Proteomics : MCP* 7.7 (July 2008), pp. 1378–1388. ISSN: 1535-9476. DOI: [10.1074/mcp.M800069-MCP200](https://doi.org/10.1074/mcp.M800069-MCP200). URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2493380/>.
- [96] James R. Faeder, Michael L. Blinov, and William S. Hlavacek. "Rule-Based Modeling of Biochemical Systems with BioNetGen." en. In: *Systems Biology. Methods in Molecular Biology*. Humana Press, 2009, pp. 113–167. ISBN: 978-1-934115-64-0 978-1-59745-525-1. DOI: [10.1007/978-1-59745-525-1_5](https://doi.org/10.1007/978-1-59745-525-1_5). URL: https://link.springer.com/protocol/10.1007/978-1-59745-525-1_5.
- [97] Leonard A. Harris, Justin S. Hogg, José-Juan Tapia, John A. P. Sekar, Sanjana Gupta, Ilya Korsunsky, Arshi Arora, Dipak Barua, Robert P. Sheehan, and James R. Faeder. "BioNetGen 2.2: advances in rule-based modeling." en. In: *Bioinformatics* 32.21 (Nov. 2016). Publisher: Oxford Academic, pp. 3366–3368. ISSN: 1367-4803. DOI: [10.1093/bioinformatics/btw469](https://doi.org/10.1093/bioinformatics/btw469). URL: <https://academic.oup.com/bioinformatics/article/32/21/3366/2415298>.
- [98] Carlos F. Lopez, Jeremy L. Muhlich, John A. Bachman, and Peter K. Sorger. "Programming biological models in Python using PySB." In: *Molecular Systems Biology* 9.1 (Jan. 1, 2013), p. 646. ISSN: 1744-4292, 1744-4292. DOI: [10.1038/msb.2013.1](https://doi.org/10.1038/msb.2013.1). URL: <http://msb.embopress.org/content/9/1/646>.
- [99] CMA-ES. en. Page Version ID: 967047119. July 2020. URL: <https://en.wikipedia.org/w/index.php?title=CMA-ES&oldid=967047119>.

- [100] Alan C. Hindmarsh, Peter N. Brown, Keith E. Grant, Steven L. Lee, Radu Serban, Dan E. Shumaker, and Carol S. Woodward. "SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers." In: *ACM Transactions on Mathematical Software* 31.3 (Sept. 1, 2005), pp. 363–396. ISSN: 00983500. DOI: [10.1145/1089014.1089020](https://doi.org/10.1145/1089014.1089020). URL: <http://portal.acm.org/citation.cfm?doid=1089014.1089020>.
- [101] Jeff Bezanson, Jiahao Chen, Stefan Karpinski, Viral Shah, and Alan Edelman. "Array Operators Using Multiple Dispatch: A Design Methodology for Array Implementations in Dynamic Languages." In: *Proceedings of ACM SIGPLAN International Workshop on Libraries, Languages, and Compilers for Array Programming*. ARRAY'14. New York, NY, USA: ACM, 2014, 56:56–56:61. ISBN: 978-1-4503-2937-8. DOI: [10.1145/2627373.2627383](https://doi.org/10.1145/2627373.2627383). URL: <https://doi.org/10.1145/2627373.2627383>.
- [102] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. "Julia: A fresh approach to numerical computing." In: *SIAM Review* 59.1 (2017), pp. 65–98. DOI: [10.1137/141000671](https://doi.org/10.1137/141000671).
- [103] Francesco Zappa Nardelli, Julia Belyakova, Artem Pelenitsyn, Benjamin Chung, Jeff Bezanson, and Jan Vitek. "Julia Subtyping: A Rational Reconstruction." In: *Proc. ACM Program. Lang.* 2.OOPSLA (Oct. 2018), 113:1–113:27. ISSN: 2475-1421. DOI: [10.1145/3276483](https://doi.org/10.1145/3276483). URL: <https://doi.org/10.1145/3276483>.
- [104] Jeff Bezanson, Jiahao Chen, Benjamin Chung, Stefan Karpinski, Viral B. Shah, Jan Vitek, and Lionel Zoubritzky. "Julia: Dynamism and Performance Reconciled by Design." In: *Proc. ACM Program. Lang.* 2.OOPSLA (Oct. 2018), 120:1–120:23. ISSN: 2475-1421. DOI: [10.1145/3276490](https://doi.org/10.1145/3276490). URL: <https://doi.org/10.1145/3276490>.
- [105] Christopher Rackauckas and Qing Nie. "DifferentialEquations.jl – A Performant and Feature-Rich Ecosystem for Solving Differential Equations in Julia." In: *Journal of Open Research Software* 5.1 (May 25, 2017). ISSN: 2049-9647. DOI: [10.5334/jors.151](https://doi.org/10.5334/jors.151). URL: <http://openresearchsoftware.metajnl.com/articles/10.5334/jors.151/>.
- [106] Jérôme Bürgi et al. "CMG2/ANTXR2 regulates extracellular collagen VI which accumulates in hyaline fibromatosis syndrome." en. In: *Nature Communications* 8 (June 2017), ncomms15861. ISSN: 2041-1723. DOI: [10.1038/ncomms15861](https://doi.org/10.1038/ncomms15861). URL: <https://www.nature.com/articles/ncomms15861>.
- [107] Claire V. Reeves, Xing Wang, Pelisa C. Charles-Horvath, Joy Y. Vink, Valeriya Y. Borisenko, John A. T. Young, and Jan K. Kitajewski. "Anthrax toxin receptor 2 functions in ECM homeostasis of the murine reproductive tract and promotes MMP activity." eng. In: *PloS One*

- 7.4 (2012), e34862. ISSN: 1932-6203. DOI: [10.1371/journal.pone.0034862](https://doi.org/10.1371/journal.pone.0034862).
- [108] Sandra Hanks et al. "Mutations in the gene encoding capillary morphogenesis protein 2 cause juvenile hyaline fibromatosis and infantile systemic hyalinosis." eng. In: *American Journal of Human Genetics* 73.4 (Oct. 2003), pp. 791–800. ISSN: 0002-9297. DOI: [10.1086/378418](https://doi.org/10.1086/378418).
- [109] Oonagh Dowling et al. "Mutations in capillary morphogenesis gene-2 result in the allelic disorders juvenile hyaline fibromatosis and infantile systemic hyalinosis." eng. In: *American Journal of Human Genetics* 73.4 (Oct. 2003), pp. 957–966. ISSN: 0002-9297. DOI: [10.1086/378781](https://doi.org/10.1086/378781).
- [110] Nazneen Rahman et al. "The gene for juvenile hyaline fibromatosis maps to chromosome 4q21." eng. In: *American Journal of Human Genetics* 71.4 (Oct. 2002), pp. 975–980. ISSN: 0002-9297. DOI: [10.1086/342776](https://doi.org/10.1086/342776).
- [111] Joseph TC Shieh, H. Eugene Hoyme, and Laura T. Arbour. "Hyaline Fibromatosis Syndrome." eng. In: *GeneReviews®*. Ed. by Margaret P. Adam, Holly H. Ardinger, Roberta A. Pagon, Stephanie E. Wallace, Lora JH Bean, Karen Stephens, and Anne Amemiya. Seattle (WA): University of Washington, Seattle, 1993. URL: <http://www.ncbi.nlm.nih.gov/books/NBK1525/>.
- [112] Heather M. Scobie, G. Jonah A. Rainey, Kenneth A. Bradley, and John A. T. Young. "Human capillary morphogenesis protein 2 functions as an anthrax toxin receptor." eng. In: *Proceedings of the National Academy of Sciences of the United States of America* 100.9 (Apr. 2003), pp. 5170–5174. ISSN: 0027-8424. DOI: [10.1073/pnas.0431098100](https://doi.org/10.1073/pnas.0431098100).
- [113] K. A. Bradley, J. Mogridge, M. Mourez, R. J. Collier, and J. A. Young. "Identification of the cellular receptor for anthrax toxin." eng. In: *Nature* 414.6860 (Nov. 2001), pp. 225–229. ISSN: 0028-0836. DOI: [10.1038/n35101999](https://doi.org/10.1038/n35101999).
- [114] Sarah Friebe, F. Gisou van der Goot, and Jérôme Bürgi. "The Ins and Outs of Anthrax Toxin." eng. In: *Toxins* 8.3 (Mar. 2016). ISSN: 2072-6651. DOI: [10.3390/toxins8030069](https://doi.org/10.3390/toxins8030069).
- [115] Laurence Abrami, Stephen H. Leppla, and F. Gisou van der Goot. "Receptor palmitoylation and ubiquitination regulate anthrax toxin endocytosis." eng. In: *The Journal of Cell Biology* 172.2 (Jan. 2006), pp. 309–320. ISSN: 0021-9525. DOI: [10.1083/jcb.200507067](https://doi.org/10.1083/jcb.200507067).
- [116] Jérôme Bürgi, Bin Xue, Vladimir N. Uversky, and F. Gisou van der Goot. "Intrinsic Disorder in Transmembrane Proteins: Roles in Signaling and Topology Prediction." eng. In: *PloS One* 11.7 (2016), e0158594. ISSN: 1932-6203. DOI: [10.1371/journal.pone.0158594](https://doi.org/10.1371/journal.pone.0158594).

- [117] Sanja Blaskovic. "Mechanism and Function of S-palmitoylation in Capillary Morphogenesis Gene 2." en. PhD thesis. 2014. DOI: [10.5075/epfl-thesis-6027](https://infoscience.epfl.ch/record/195168). URL: <https://infoscience.epfl.ch/record/195168>.
- [118] Julie Deuquet, Ekkehart Lausch, Andrea Superti-Furga, and F. Gisou van der Goot. "The dark sides of capillary morphogenesis gene 2." en. In: *The EMBO Journal* 31.1 (Jan. 2012), pp. 3–13. ISSN: 0261-4189, 1460-2075. DOI: [10.1038/emboj.2011.442](https://doi.org/10.1038/emboj.2011.442). URL: <http://emboj.embopress.org/content/31/1/3>.
- [119] The UniProt Consortium. "UniProt: a worldwide hub of protein knowledge." en. In: *Nucleic Acids Research* 47.D1 (Jan. 2019). Publisher: Oxford Academic, pp. D506–D515. ISSN: 0305-1048. DOI: [10.1093/nar/gky1049](https://doi.org/10.1093/nar/gky1049). URL: <https://academic.oup.com/nar/article/47/D1/D506/5160987>.
- [120] The UniProt Consortium. "UniProt: the universal protein knowledgebase." en. In: *Nucleic Acids Research* 46.5 (Mar. 2018). Publisher: Oxford Academic, pp. 2699–2699. ISSN: 0305-1048. DOI: [10.1093/nar/gky092](https://doi.org/10.1093/nar/gky092). URL: <https://academic.oup.com/nar/article/46/5/2699/4841658>.
- [121] Mathias Uhlen et al. "Tissue-based map of the human proteome." en. In: *Science* 347.6220 (Jan. 2015). Publisher: American Association for the Advancement of Science Section: Research Article. ISSN: 0036-8075, 1095-9203. DOI: [10.1126/science.1260419](https://doi.org/10.1126/science.1260419). URL: <https://science.sciencemag.org/content/347/6220/1260419>.
- [122] Mathias Uhlen et al. "A pathology atlas of the human cancer transcriptome." en. In: *Science* 357.6352 (Aug. 2017). Publisher: American Association for the Advancement of Science Section: Research Article. ISSN: 0036-8075, 1095-9203. DOI: [10.1126/science.aan2507](https://doi.org/10.1126/science.aan2507). URL: <https://science.sciencemag.org/content/357/6352/eaan2507>.
- [123] Peter J. Thul et al. "A subcellular map of the human proteome." en. In: *Science* 356.6340 (May 2017). Publisher: American Association for the Advancement of Science Section: Research Article. ISSN: 0036-8075, 1095-9203. DOI: [10.1126/science.aal3321](https://doi.org/10.1126/science.aal3321). URL: <https://science.sciencemag.org/content/356/6340/eaal3321>.
- [124] Martin Göttle, Stefan Dove, and Roland Seifert. "Bacillus anthracis Edema Factor Substrate Specificity: Evidence for New Modes of Action." en. In: *Toxins* 4.7 (July 2012). Number: 7 Publisher: Molecular Diversity Preservation International, pp. 505–535. DOI: [10.3390/toxins4070505](https://doi.org/10.3390/toxins4070505). URL: <https://www.mdpi.com/2072-6651/4/7/505>.

- [125] Jennifer Greaves and Luke H. Chamberlain. “DHHC palmitoyl transferases: substrate interactions and (patho)physiology.” English. In: *Trends in Biochemical Sciences* 36.5 (May 2011). Publisher: Elsevier, pp. 245–253. ISSN: 0968-0004. DOI: [10.1016/j.tibs.2011.01.003](https://doi.org/10.1016/j.tibs.2011.01.003). URL: [https://www.cell.com/trends/biochemical-sciences/abstract/S0968-0004\(11\)00014-4](https://www.cell.com/trends/biochemical-sciences/abstract/S0968-0004(11)00014-4).
- [126] Lisa Jeske, Sandra Placzek, Ida Schomburg, Antje Chang, and Dietmar Schomburg. “BRENDA in 2019: a European ELIXIR core data resource.” In: *Nucleic Acids Research* 47.D1 (Nov. 2018), pp. D542–D549. DOI: [10.1093/nar/gky1048](https://doi.org/10.1093/nar/gky1048). URL: <https://doi.org/10.1093%2Fnar%2Fgky1048>.
- [127] U. Wittig et al. “SABIO-RK–database for biochemical reaction kinetics.” In: *Nucleic Acids Research* 40.D1 (Nov. 2011), pp. D790–D796. DOI: [10.1093/nar/gkr1046](https://doi.org/10.1093/nar/gkr1046). URL: <https://doi.org/10.1093%2Fnar%2Fgkr1046>.
- [128] The UniProt Consortium. “UniProt: a worldwide hub of protein knowledge.” In: *Nucleic Acids Research* 47.D1 (Nov. 2018), pp. D506–D515. DOI: [10.1093/nar/gky1049](https://doi.org/10.1093/nar/gky1049). URL: <https://doi.org/10.1093%2Fnar%2Fgky1049>.
- [129] Dan Davidi, Elad Noor, Wolfram Liebermeister, Arren Bar-Even, Avi Flamholz, Katja Tummler, Uri Barenholz, Miki Goldenfeld, Tomer Shlomi, and Ron Milo. “Global characterization of in vivo enzyme catalytic rates and their correspondence to in vitro measurements.” In: *Proceedings of the National Academy of Sciences* 113.12 (Mar. 2016), pp. 3401–3406. DOI: [10.1073/pnas.1514240113](https://doi.org/10.1073/pnas.1514240113). URL: <https://doi.org/10.1073%2Fpnas.1514240113>.
- [130] Ljubisa Miskovic, Jonas Beal, Michael Moret, and Vassily Hatzimanikatis. *Model Classification for Uncertainty Reduction in Biochemical Kinetic Models*. en. preprint. Systems Biology, Sept. 2018. DOI: [10.1101/427716](https://doi.org/10.1101/427716). URL: <http://biorxiv.org/lookup/doi/10.1101/427716>.
- [131] Ali Khodayari and Costas D. Maranas. “A genome-scale Escherichia coli kinetic metabolic model k-ecoli457 satisfying flux data for multiple mutant strains.” In: *Nature Communications* 7 (Dec. 2016), p. 13806. DOI: [10.1038/ncomms13806](https://doi.org/10.1038/ncomms13806). URL: <https://doi.org/10.1038%2Fncomms13806>.
- [132] Linh M. Tran, Matthew L. Rizk, and James C. Liao. “Ensemble Modeling of Metabolic Networks.” In: *Biophysical Journal* 95.12 (Dec. 2008), pp. 5606–5617. DOI: [10.1529/biophysj.108.135442](https://doi.org/10.1529/biophysj.108.135442). URL: <https://doi.org/10.1529%2Fbiophysj.108.135442>.

- [133] Ljubisa Miskovic and Vassily Hatzimanikatis. "Production of biofuels and biochemicals: in need of an ORACLE." In: *Trends in Biotechnology* 28.8 (Aug. 2010), pp. 391–397. DOI: [10.1016/j.tibtech.2010.05.003](https://doi.org/10.1016/j.tibtech.2010.05.003). URL: <https://doi.org/10.1016%2Fj.tibtech.2010.05.003>.
- [134] Stefano Andreozzi, Ljubisa Miskovic, and Vassily Hatzimanikatis. "iSCHRUNK – In Silico Approach to Characterization and Reduction of Uncertainty in the Kinetic Models of Genome-scale Metabolic Networks." In: *Metabolic Engineering* 33 (Jan. 2016), pp. 158–168. DOI: [10.1016/j.ymben.2015.10.002](https://doi.org/10.1016/j.ymben.2015.10.002). URL: <https://doi.org/10.1016%2Fj.ymben.2015.10.002>.
- [135] Ryan N. Gutenkunst, Joshua J. Waterfall, Fergal P. Casey, Kevin S. Brown, Christopher R. Myers, and James P. Sethna. "Universally Sloppy Parameter Sensitivities in Systems Biology Models." In: *PLoS Computational Biology* 3.10 (2007), e189. DOI: [10.1371/journal.pcbi.0030189](https://doi.org/10.1371/journal.pcbi.0030189). URL: <https://doi.org/10.1371%2Fjournal.pcbi.0030189>.
- [136] A. Saltelli, S. Tarantola, and K. P.-S. Chan. "A Quantitative Model-Independent Method for Global Sensitivity Analysis of Model Output." In: *Technometrics* 41.1 (Feb. 1999), pp. 39–56. DOI: [10.1080/00401706.1999.10485594](https://doi.org/10.1080/00401706.1999.10485594). URL: <https://doi.org/10.1080%2F00401706.1999.10485594>.
- [137] A. Kiparissides, S. S. Kucherenko, A. Mantalaris, and E. N. Pistikopoulos. "Global Sensitivity Analysis Challenges in Biological Systems Modeling." In: *Industrial & Engineering Chemistry Research* 48.15 (Aug. 2009), pp. 7168–7180. DOI: [10.1021/ie900139x](https://doi.org/10.1021/ie900139x). URL: <https://doi.org/10.1021%2Fie900139x>.
- [138] A. Raue, T. Maiwald, J. Timmer, C. Kreutz, and U. Klingmüller. "Addressing parameter identifiability by model-based experimentation." In: *IET Systems Biology* 5.2 (Mar. 2011), pp. 120–130. DOI: [10.1049/iet-syb.2010.0061](https://doi.org/10.1049/iet-syb.2010.0061). URL: <https://doi.org/10.1049%2Fiet-syb.2010.0061>.
- [139] Alexandros Kiparissides, Christos Georgakis, Athanasios Mantalaris, and Efstratios N. Pistikopoulos. "Design of In Silico Experiments as a Tool for Nonlinear Sensitivity Analysis of Knowledge-Driven Models." In: *Industrial & Engineering Chemistry Research* 53.18 (Mar. 2014), pp. 7517–7525. DOI: [10.1021/ie4032154](https://doi.org/10.1021/ie4032154). URL: <https://doi.org/10.1021%2Fie4032154>.
- [140] A. Kiparissides and V. Hatzimanikatis. "Thermodynamics-based Metabolite Sensitivity Analysis in metabolic networks." In: *Metabolic Engineering* 39 (Jan. 2017), pp. 117–127. DOI: [10.1016/j.ymben.2016.11.006](https://doi.org/10.1016/j.ymben.2016.11.006). URL: <https://doi.org/10.1016%2Fj.ymben.2016.11.006>.

- [141] Christine Reder. “Metabolic control theory: A structural approach.” In: *Journal of Theoretical Biology* 135.2 (Nov. 1988), pp. 175–201. DOI: [10.1016/s0022-5193\(88\)80073-0](https://doi.org/10.1016/s0022-5193(88)80073-0). URL: <https://doi.org/10.1016%2Fs0022-5193%2888%2980073-0>.
- [142] Vassily Hatzimanikatis and James E. Bailey. “MCA Has More to Say.” In: *Journal of Theoretical Biology* 182.3 (Oct. 1996), pp. 233–242. DOI: [10.1006/jtbi.1996.0160](https://doi.org/10.1006/jtbi.1996.0160). URL: <https://doi.org/10.1006%2Fjtbi.1996.0160>.
- [143] Liqing Wang, Inanç Birol, and Vassily Hatzimanikatis. “Metabolic Control Analysis under Uncertainty: Framework Development and Case Studies.” In: *Biophysical Journal* 87.6 (Dec. 2004), pp. 3750–3763. DOI: [10.1529/biophysj.104.048090](https://doi.org/10.1529/biophysj.104.048090). URL: <https://doi.org/10.1529%2Fbiophysj.104.048090>.
- [144] Meric Ataman, Daniel F. Hernandez Gardiol, Georgios Fengos, and Vassily Hatzimanikatis. “redGEM: Systematic reduction and analysis of genome-scale metabolic reconstructions for development of consistent core metabolic models.” In: *PLOS Computational Biology* 13.7 (July 2017). Ed. by Ryan S Senger, e1005444. DOI: [10.1371/journal.pcbi.1005444](https://doi.org/10.1371/journal.pcbi.1005444). URL: <https://doi.org/10.1371%2Fjournal.pcbi.1005444>.
- [145] Meric Ataman and Vassily Hatzimanikatis. “lumpGEM: Systematic generation of subnetworks and elementally balanced lumped reactions for the biosynthesis of target metabolites.” In: *PLOS Computational Biology* 13.7 (July 2017). Ed. by Satoru Miyano, e1005513. DOI: [10.1371/journal.pcbi.1005513](https://doi.org/10.1371/journal.pcbi.1005513). URL: <https://doi.org/10.1371%2Fjournal.pcbi.1005513>.
- [146] J. D. Orth, T. M. Conrad, J. Na, J. A. Lerman, H. Nam, A. M. Feist, and B. O. Palsson. “A comprehensive genome-scale reconstruction of Escherichia coli metabolism–2011.” In: *Molecular Systems Biology* 7.1 (Apr. 2014), pp. 535–535. DOI: [10.1038/msb.2011.65](https://doi.org/10.1038/msb.2011.65). URL: <https://doi.org/10.1038%2Fmsb.2011.65>.
- [147] I.M Sobol. “Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates.” In: *Mathematics and Computers in Simulation* 55.1-3 (Feb. 2001), pp. 271–280. DOI: [10.1016/s0378-4754\(00\)00270-6](https://doi.org/10.1016/s0378-4754(00)00270-6). URL: <https://doi.org/10.1016%2Fs0378-4754%2800%2900270-6>.
- [148] S Schnell. “Enzyme Kinetics at High Enzyme Concentration.” In: *Bulletin of Mathematical Biology* 62.3 (May 2000), pp. 483–499. DOI: [10.1006/bulm.1999.0163](https://doi.org/10.1006/bulm.1999.0163). URL: <https://doi.org/10.1006%2Fbulm.1999.0163>.

- [149] L SEGEL. "On the validity of the steady state assumption of enzyme kinetics." In: *Bulletin of Mathematical Biology* 50.6 (1988), pp. 579–593. DOI: [10.1016/S0092-8240\(88\)80057-0](https://doi.org/10.1016/S0092-8240(88)80057-0). URL: <https://doi.org/10.1016%2Fs0092-8240%2888%2980057-0>.
- [150] Lee A. Segel and Marshall Slemrod. "The Quasi-Steady-State Assumption: A Case Study in Perturbation." In: *SIAM Review* 31.3 (Sept. 1989), pp. 446–477. DOI: [10.1137/1031091](https://doi.org/10.1137/1031091). URL: <https://doi.org/10.1137%2F1031091>.
- [151] A Varma and BO Palsson. "Stoichiometric flux balance models quantitatively predict growth and metabolic by-product secretion in wild-type *Escherichia coli* W3110." In: *Appl Environ Microbiol* 60 (Oct. 1994), pp. 3724–31.
- [152] J Pramanik and JD Keasling. "Stoichiometric model of *Escherichia coli* metabolism: incorporation of growth-rate dependent biomass composition and mechanistic energy requirements." In: *Biotechnol Bioeng* 56 (Nov. 1997), pp. 398–421.
- [153] Christopher S. Henry, Linda J. Broadbelt, and Vassily Hatzimanikatis. "Thermodynamics-Based Metabolic Flux Analysis." In: *Biophysical Journal* 92.5 (Mar. 2007), pp. 1792–1805. DOI: [10.1529/biophysj.106.093138](https://doi.org/10.1529/biophysj.106.093138). URL: <https://doi.org/10.1529%2Fbiophysj.106.093138>.
- [154] Meric Ataman and Vassily Hatzimanikatis. "Heading in the right direction: thermodynamics-based network analysis and pathway engineering." In: *Current Opinion in Biotechnology* 36 (Dec. 2015), pp. 176–182. DOI: [10.1016/j.copbio.2015.08.021](https://doi.org/10.1016/j.copbio.2015.08.021). URL: <https://doi.org/10.1016%2Fj.copbio.2015.08.021>.
- [155] T Hameri, G Fengos, M Ataman, L Miskovic, and V Hatzimanikatis. "Kinetic models of metabolism that consider alternative steady-state solutions of intracellular fluxes and concentrations." In: *Metab Eng* 52 (Mar. 2019), pp. 29–41.
- [156] Stefano Andreozzi, Ljubisa Miskovic, and Vassily Hatzimanikatis. "iSCHRUNK – In Silico Approach to Characterization and Reduction of Uncertainty in the Kinetic Models of Genome-scale Metabolic Networks." In: *Metabolic Engineering* 33 (Jan. 2016), pp. 158–168. DOI: [10.1016/j.ymben.2015.10.002](https://doi.org/10.1016/j.ymben.2015.10.002). URL: <https://doi.org/10.1016%2Fj.ymben.2015.10.002>.
- [157] KC Soh, L Miskovic, and V Hatzimanikatis. "From network models to network responses: integration of thermodynamic and kinetic properties of yeast genome-scale metabolic networks." In: *FEMS Yeast Res* 12 (Mar. 2012), pp. 129–43.

- [158] L Wang and V Hatzimanikatis. "Metabolic engineering under uncertainty. I: framework development." In: *Metab Eng* 8 (Mar. 2006), pp. 133–41.
- [159] Douglas McCloskey, Jon A. Gangoiti, Zachary A. King, Robert K. Naviaux, Bruce A. Barshop, Bernhard O. Palsson, and Adam M. Feist. "A model-driven quantitative metabolomics analysis of aerobic and anaerobic metabolism in *E. coli* K-12 MG1655 that is biochemically and thermodynamically consistent." In: *Biotechnology and Bioengineering* 111.4 (Nov. 2013), pp. 803–815. DOI: [10.1002/bit.25133](https://doi.org/10.1002/bit.25133). URL: <https://doi.org/10.1002%2Fbit.25133>.
- [160] Fanny Sarrazin, Francesca Pianosi, and Thorsten Wagener. "Global Sensitivity Analysis of environmental models: Convergence and validation." In: *Environmental Modelling & Software* 79 (May 2016), pp. 135–152. DOI: [10.1016/j.envsoft.2016.02.005](https://doi.org/10.1016/j.envsoft.2016.02.005). URL: <https://doi.org/10.1016%2Fj.envsoft.2016.02.005>.
- [161] Majdi Awad, Tristan Senga Kiese, Zainab Assaghir, and Anne Ventura. "Convergence of sensitivity analysis methods for evaluating combined influences of model inputs." In: *Reliability Engineering & System Safety* 189 (Sept. 2019), pp. 109–122. DOI: [10.1016/j.res.2019.03.050](https://doi.org/10.1016/j.res.2019.03.050). URL: <https://doi.org/10.1016%2Fj.res.2019.03.050>.
- [162] Nan-Hung Hsieh, Brad Reisfeld, Frederic Y. Bois, and Weihsueh A. Chiu. "Applying a Global Sensitivity Analysis Workflow to Improve the Computational Efficiencies in Physiologically-Based Pharmacokinetic Modeling." In: *Frontiers in Pharmacology* 9 (June 2018). DOI: [10.3389/fphar.2018.00588](https://doi.org/10.3389/fphar.2018.00588). URL: <https://doi.org/10.3389%2Ffphar.2018.00588>.
- [163] Francesca Pianosi and Thorsten Wagener. "A simple and efficient method for global sensitivity analysis based on cumulative distribution functions." In: *Environmental Modelling & Software* 67 (May 2015), pp. 1–11. DOI: [10.1016/j.envsoft.2015.01.004](https://doi.org/10.1016/j.envsoft.2015.01.004). URL: <https://doi.org/10.1016%2Fj.envsoft.2015.01.004>.
- [164] Stefano Andreozzi, Ljubisa Miskovic, and Vassily Hatzimanikatis. "iSCHRUNK – In Silico Approach to Characterization and Reduction of Uncertainty in the Kinetic Models of Genome-scale Metabolic Networks." In: *Metabolic Engineering* 33 (Jan. 2016), pp. 158–168. DOI: [10.1016/j.ymben.2015.10.002](https://doi.org/10.1016/j.ymben.2015.10.002). URL: <https://doi.org/10.1016%2Fj.ymben.2015.10.002>.
- [165] V Hatzimanikatis and JE Bailey. "Effects of spatiotemporal variations on metabolic control: approximate analysis using (log)linear kinetic models." In: *Biotechnol Bioeng* 54 (Apr. 1997), pp. 91–104.

- [166] R. Heinrich, S. M. Rapoport, and T. A. Rapoport. "Metabolic Regulation and Mathematical Models." en. In: *Progress in Biophysics and Molecular Biology*. Ed. by J. A. V. Butler and D. Noble. Pergamon, Jan. 1978, pp. 1–82. ISBN: 978-0-08-020295-2. DOI: [10.1016/B978-0-08-020295-2.50004-0](https://doi.org/10.1016/B978-0-08-020295-2.50004-0). URL: <http://www.sciencedirect.com/science/article/pii/B9780080202952500040>.
- [167] H. Kacser, J. A. Burns, H. Kacser, and D. A. Fell. "The control of flux." In: *Biochemical Society Transactions* 23.2 (May 1995), pp. 341–366. DOI: [10.1042/bst0230341](https://doi.org/10.1042/bst0230341). URL: <https://doi.org/10.1042%2Fbst0230341>.
- [168] V Hatzimanikatis and JE Bailey. "MCA has more to say." In: *J Theor Biol* 182 (Oct. 1996), pp. 233–42.
- [169] L Wang and V Hatzimanikatis. "Metabolic engineering under uncertainty. I: Framework development." In: *Metabolic Engineering* 8.2 (Mar. 2006), pp. 133–141. DOI: [10.1016/j.ymben.2005.11.003](https://doi.org/10.1016/j.ymben.2005.11.003). URL: <https://doi.org/10.1016%2Fj.ymben.2005.11.003>.
- [170] L Mišković and V Hatzimanikatis. "Modeling of uncertainties in biochemical reactions." In: *Biotechnol Bioeng* 108 (Feb. 2011), pp. 413–23.
- [171] Reinhart Heinrich and Stefan Schuster. *The Regulation of Cellular Systems*. Springer US, 1996. DOI: [10.1007/978-1-4613-1161-4](https://doi.org/10.1007/978-1-4613-1161-4). URL: <https://doi.org/10.1007%2F978-1-4613-1161-4>.
- [172] David A. Fell and Herbert M. Sauro. "Metabolic control and its analysis. Additional relationships between elasticities and control coefficients." In: *European Journal of Biochemistry* 148.3 (May 1985), pp. 555–561. DOI: [10.1111/j.1432-1033.1985.tb08876.x](https://doi.org/10.1111/j.1432-1033.1985.tb08876.x). URL: <https://doi.org/10.1111%2Fj.1432-1033.1985.tb08876.x>.
- [173] Irene Bauer, Hans Georg Bock, Stefan Körkel, and Johannes P. Schlöder. "Numerical methods for optimum experimental design in DAE systems." In: *Journal of Computational and Applied Mathematics* 120.1 (Aug. 2000), pp. 1–25. ISSN: 0377-0427. DOI: [10.1016/S0377-0427\(00\)00300-9](https://doi.org/10.1016/S0377-0427(00)00300-9). URL: <http://www.sciencedirect.com/science/article/pii/S0377042700003009>.
- [174] Samuel Bandara, Johannes P Schlöder, Roland Eils, Hans Bock, and Tobias Meyer. "Optimal Experimental Design for Parameter Estimation of a Cell Signaling Model." In: *PLoS computational biology* 5 (Nov. 2009), e1000558. DOI: [10.1371/journal.pcbi.1000558](https://doi.org/10.1371/journal.pcbi.1000558).
- [175] R. J. Flassig and K. Sundmacher. "Optimal design of stimulus experiments for robust discrimination of biochemical reaction networks." en. In: *Bioinformatics* 28.23 (Dec. 2012), pp. 3089–3096. ISSN: 1367-4803. DOI: [10.1093/bioinformatics/bts585](https://doi.org/10.1093/bioinformatics/bts585). URL: <https://academic.oup.com/bioinformatics/article/28/23/3089/194080>.

- [176] Nathan Braniff and Brian Ingalls. “New opportunities for optimal design of dynamic experiments in systems and synthetic biology.” In: *Current Opinion in Systems Biology*. Mathematic modelling 9 (June 2018), pp. 42–48. ISSN: 2452-3100. DOI: [10.1016/j.coisb.2018.02.005](https://doi.org/10.1016/j.coisb.2018.02.005). URL: <http://www.sciencedirect.com/science/article/pii/S2452310017302202>.
- [177] Mike Innes, Alan Edelman, Keno Fischer, Chris Rackauckas, Elliot Saba, Viral B. Shah, and Will Tebbutt. “A Differentiable Programming System to Bridge Machine Learning and Scientific Computing.” In: *arXiv:1907.07587 [cs]* (July 2019). arXiv: 1907.07587. URL: <http://arxiv.org/abs/1907.07587>.
- [178] Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, and Ali Ramadhan. “Universal Differential Equations for Scientific Machine Learning.” en. In: *arXiv:2001.04385 [cs, math, q-bio, stat]* (Jan. 2020). arXiv: 2001.04385. URL: <http://arxiv.org/abs/2001.04385>.
- [179] Dongbin Xiu. *Numerical Methods for Stochastic Computations: A Spectral Method Approach*. Princeton University Press, 2010. ISBN: 978-0-691-14212-8. DOI: [10.2307/j.ctv7h0skv](https://doi.org/10.2307/j.ctv7h0skv). URL: <https://www.jstor.org/stable/j.ctv7h0skv>.
- [180] Thierry Crestaux, Olivier Le Maître, and Jean-Marc Martinez. “Polynomial chaos expansion for sensitivity analysis.” en. In: *Reliability Engineering & System Safety*. Special Issue on Sensitivity Analysis 94.7 (July 2009), pp. 1161–1172. ISSN: 0951-8320. DOI: [10.1016/j.ress.2008.10.008](https://doi.org/10.1016/j.ress.2008.10.008). URL: <http://www.sciencedirect.com/science/article/pii/S0951832008002561>.
- [181] Bruno Sudret. “Global sensitivity analysis using polynomial chaos expansions.” en. In: *Reliability Engineering & System Safety*. Bayesian Networks in Dependability 93.7 (July 2008), pp. 964–979. ISSN: 0951-8320. DOI: [10.1016/j.ress.2007.04.002](https://doi.org/10.1016/j.ress.2007.04.002). URL: <http://www.sciencedirect.com/science/article/pii/S0951832007001329>.
- [182] Dongbin Xiu and Jan S. Hesthaven. “High-Order Collocation Methods for Differential Equations with Random Inputs.” In: *SIAM Journal on Scientific Computing* 27.3 (Jan. 2005). Publisher: Society for Industrial and Applied Mathematics, pp. 1118–1139. ISSN: 1064-8275. DOI: [10.1137/040615201](https://doi.org/10.1137/040615201). URL: <https://epubs.siam.org/doi/10.1137/040615201>.
- [183] G. Deman, K. Konakli, B. Sudret, J. Kerrou, P. Perrochet, and H. Benabderrahmane. “Using sparse polynomial chaos expansions for the global sensitivity analysis of groundwater lifetime expectancy in a multi-layered hydrogeological model.” en. In: *Reliability Engineering & System Safety* 147 (Mar. 2016), pp. 156–169. ISSN: 09518320. DOI:

- 10.1016/j.res.2015.11.005. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0951832015003324>.
- [184] Daniele Bigoni. "Uncertainty Quantification with Applications to Engineering Problems." PhD thesis. Feb. 2015.
- [185] G.J.A. Loeven, Jeroen Witteveen, and Hester Bijl. "Probabilistic Collocation: An Efficient Non-Intrusive Approach for Arbitrarily Distributed Parametric Uncertainties." In: *Collection of Technical Papers - 45th AIAA Aerospace Sciences Meeting*. Vol. 6. Jan. 2007. ISBN: 978-1-62410-012-3. DOI: 10.2514/6.2007-317.
- [186] Francesca Pianosi and Thorsten Wagener. "A simple and efficient method for global sensitivity analysis based on cumulative distribution functions." In: *Environmental Modelling & Software* 67 (May 2015), pp. 1–11. ISSN: 1364-8152. DOI: 10.1016/j.envsoft.2015.01.004. URL: <http://www.sciencedirect.com/science/article/pii/S1364815215000237>.