UNIVERSITÉ D'ORLÉANS

ÉCOLE DOCTORALE SANTÉ, SCIENCES BIOLOGIQUES ET CHIMIE DU VIVANT

CENTRE DE BIOPHYSIQUE MOLÉCULAIRE

THÈSE présentée par :

Thibault FILLION

soutenue le : **19 juillet 2024**

pour obtenir le grade de : **Docteur de l'Université d'Orléans**

Discipline/ Spécialité : Biologie

Non-equilibrium effects in cell biology: Mathematical and computational modeling of G-protein signal transduction and other biochemical reaction-diffusion networks

Effets de non-équilibre en biologie cellulaire : Modélisation mathématique et informatique
de la transduction du signal par les protéines G et autres réseaux biochimiques de réaction-diffusion

THÈSE dirigée par :

Francesco PIAZZA	Professeur des Universités, Université de Florence
Josef HAMACEK	Professeur des Universités, Université d'Orléans

RAPPORTEURS :

Francesca DI PATTI	Professeure des Universités, Université de Perugia
Alessandro BARDUCCI	Directeur de recherche, HDR

JURY :

Francesco PIAZZA	Professeur des Universités, Université de Florence
Josef HAMACEK	Professeur des Universités, Université d'Orléans
Francesca DI PATTI	Professeure des Universités, Université de Perugia, **(Présidente)**
Alessandro BARDUCCI	Directeur de recherche, HDR

Je ne remercierai jamais le Professeur Francesco Piazza, Professeur des universités associé à l'Université de Florence, mon directeur de thèse, pour son soutien et son attention, pour s'être toujours rendu disponible pour m'aiguiller et faire le point sur l'avancée des projets, pour sa patience infinie et sa bienveillance, et surtout pour avoir su me transmettre le goût de la physique, que j'espère ne jamais cesser de cultiver. Je lui suis également extrêmement reconnaissant pour tout le temps passé à relire et corriger ce manuscrit.

Je remercie également le Professeur Josef Hamacek, Professeur des universités à l'université d'Orléans, pour avoir accepté de co-encadrer ma thèse, et pour toujours s'être montré bienveillant et attentif à mon égard.

Un grand merci à la Professeure Francesca Di Patti, Professeure des universités à l'université de Perugia, et au Docteur Alessandro Barducci, Directeur de recherche au CBS, pour avoir acceptés de faire partie du jury de thèse en tant que rapporteuse et rapporteur.

Je tiens à remercier chaleureusement le Docteur Matthieu Réfrégiers, Directeur du Centre de Biophysique Moléculaire (CBM), et la Docteure Hélène Bénédetti, Directrice de recherche au CBM et directrice-adjointe de l'école doctorale SSBCV, pour leur attention et leur bienveillance. Ils ont su me guider et me rassurer, en particulier durant la rédaction de ce manuscrit.

Je remercie également le Professeur Duccio Fanelli, Professeur des universités à l'université de Florence, et la Professeure Annarosa Arcangeli, Professeure des universités à l'université de Florence, pour m'avoir accepté au sein de leurs départements respectifs, le Département de Physique & Astronomie et le Département de Médecine Expérimentale et Clinique, de l'Université de Florence, en Italie. Ce fut une expérience riche, aussi bien scientifiquement qu'humainement.

Je remercie également la Docteure Ginevra Chioccioli Altadonna pour m'avoir enseigné la culture cellulaire et pour la patience dont elle a fait preuve à cette occasion.

Je tiens par ailleurs à remercier tous mes collègues du CBM à Orléans, ainsi que ceux du Département de Physique & Astronomie et du Département de Médecine Expérimentale et Clinique, de l'Université de Florence. Je pense notamment à Éric Kaya, Josipa Cecić Vidoš et Giuliano Migliorini, mes collègues du CBM. J'ai sincèrement apprécié leur bonne humeur et leur compagnie pendant le temps que j'ai passé au CBM. Je ne remercierai jamais assez non plus ma famille, en particulier mes parents Joël Fillion et Marie Fillion et mon petit frère Grégoire Fillion, qui m'ont toujours soutenu et encouragé tout au long ces quatre années. Pour leur affection, leur écoute, leurs conseils, et pour m'avoir aidé si souvent à y voir plus clair. Je n'en serai pas là sans eux.

Je me dois également de remercier tous mes amis, qui ont su m'encourager durant ma thèse. Je pense notamment à Shagnik Chaudhuri et à Quentin Mura, avec qui j'ai eu la chance d'échanger durant cette thèse. Je remercie également chaleureusement Clément, Alice, Anaël, James et Anne Catherine pour leur présence et leur soutien, notamment au moment de la soutenance.

Contents

Chapter 0

Résumé en français

0.1 Introduction

Les propriétés du vivant, à l'échelle de la cellule, émergent de réseaux de réaction-diffusion complexes. Une cellule est un système thermodynamique ouvert, au sens ou il est délimité par une membrane partiellement-perméable, au niveau de laquelle se joue un échange de matière avec l'extérieur. Un tel système tend généralement vers un état stationnaire, qui peut être d'équilibre ou de non-équilibre. l'état d'équilibre est caractérisé par une absence de courants de réaction, condition aussi appelée bilan détaillé, ou réversibilité microscopique. Dans la cellule, l'activité métabolique convertit des sources d'énergie externes en adénosine tri-phosphate (ATP), et autres sources d'énergie utilisables par les processus cellulaires, ce qui conduit le système loin de l'équilibre, rendant possible l'activité cellulaire. Pour comprendre précisément comment cette propriété thermodynamique impacte des processus précis, il est possible d'étudier des modèles théoriques, consistant en des ensembles d'équation stoechiométriques gouvernées par des constantes cinétiques, et dont l'évolution temporelle peut être simulée en résolvant l'équation des taux ou l'équation maîtresse [22] correspondante. C'est en partie le sujet du premier chapitre. Cependant, un tel réseau de réaction ne satisfait pas automatiquement le bilan détaillé. Il est nécessaire d'en contraindre les paramètres pour que ce soit le cas [8]. Cette consistance thermodynamique est nécessaire pour caractériser correctement les bilans énergétiques des systèmes.

Dans le second chapitre, on s'intéresse à l'exemple concret de la transduction du signal par les GTPases. On cherche notamment à caractériser les effets de non-équilibre impliqués dans leur régulation, afin de pouvoir envisager des systèmes plus complexes comme celui des récepteurs couplés aux protéines G (GPCRs). Ce travail implique donc la construction des réseaux de réaction cinétiques modélisant chaque système, ainsi que la résolution de leurs contraintes thermodynamiques. Il donnent lieu à un éclairage théorique intéressant sur le fonctionnement des GTPases, en particulier dans la transduction du signal par les GPCRs, pour lequel nous proposons un modèle détaillé et facilement paramétrable.

Ces travaux nécessitent des outils informatiques spécifiques, en particulier pour la modélisation et la simulation des systèmes de réaction-diffusion biochimiques. Nous avons donc développé

STReNGTHS, un nouvel outil de modélisation et de simulation sous la forme d'un package Python, que nous présentons dans le premier chapitre, en complément d'un rappel sur les bases théoriques associées.

0.2 STReNGTHS

Beaucoup de propriétés des organismes vivants peuvent émerger de réseaux de réaction-diffusion complexes. Une façon de représenter de tels réseaux est par le biais de systèmes d'équations stoechiométriques, gouvernées par des constantes cinétiques. Ces réseaux de réactions peuvent être couplés à des processus de diffusion: on parle de système de réaction-diffusion.

Il existe alors différentes façons de simuler l'évolution des systèmes chimiques correspondants dans le temps: Les méthodes déterministes, consistant à intégrer les équations de taux associées au système, et les méthodes stochastiques, telles que l'algorithme de Gillespie [22], qui permettent notamment de tenir compte des quantités finies de particules/molécules dans le système et des fluctuations qui en découlent [22], et donc d'étudier des systèmes multi-stables [5].

Plus un système est grand et complexe, plus il est fastidieux et source d'erreur de re-implémenter "from scratch" un logiciel de simulation. Nous présentons donc STReNGTHS, acronyme de "Simulation and modeling Tool for Reaction-diffusion Networks in Graphs and Tridimensional Heterogeneous Systems", un package Python libre et open-source développé durant cette thèse, destiné à la modélisation et la simulation de systèmes de réaction-diffusion inhomogènes [18].

Pour cet outil, nous avons choisi une interface de programmation afin de le rendre le plus versatile possible. Le choix du Langage Python repose sur sa riche diversité de bibliothèques de traitement et de visualisation des données, ainsi que sa forte popularité dans le milieu scientifique. Les algorithmes de simulation sous-jacents sont implémentés en C++ et compilés sous forme de librairie dynamique partagée, interfacée avec python via sa librairie standard. Le choix d'un langage compilé comme le C++ pour les calculs lourds (simulations) permet une meilleur performance (vitesse d'exécution). L'intégralité du code source de l'outil ainsi que de sa documentation sont publiquement accessibles sur le dépôt GitHub du projet. Ce dernier est publié sous une licence permissive MIT, affin de permettre à chacun de participer à son développement.

Bien que de nombreux logiciels riches existent déjà [26] [29] [64] [27], nous pensons que STReNGTHS trouve sa force dans sa simplicité d'utilisation et dans son extensibilité.

0.2.1 Modéliser un système de réactions chimiques

Dans cette section, on commence par présenter les bases théoriques de la description de réseaux de réaction par des systèmes d'équations stoechiométriques. Une espèce chimique est un type de molécule ou de complexe, diffusant et réagissant comme une seule entité. Une réaction, ou transformation chimique, est un évènement localisé impliquant la consommation et/ou la production d'espèces chimiques. Elle peut être décrite à l'aide d'une équation stoechiométrique, ou chaque côté peut s'écrire comme une somme sur l'ensemble des espèces, impliquant différent coefficient stoechiométriques. Une seule équation stoechimoétrique peut décrire une transformation réversible, donc deux réactions. En plus, pour chaque direction de transformation, l'équation stoechiométrique donne une constante cinétique, qui décrit la cadence à laquelle la réaction se réalise. Un réseau de réaction couple un ensemble de réactions, décrites par différentes équations stoechiométriques,

qui opèrent sur un même ensemble d'espèces. Un système chimique associe un réseau de réaction à un état de système, qui consiste en un vecteur associant une quantité de matière à chaque espèce chimique.

Dans un second temps, on présente comment les différent concepts précédemment évoqués sont implémentés dans STReNGTHS, et on montre comment modéliser pas à pas un système de réaction, en présentant progressivement l'interface de programmation (API) Python des classes impliquées. En particulier, les concepts de espèce, de réaction, de réseau de réactions et de système chimique Une espèces chimique est représentée par un objet de la classes "Species", associant à l'espèce un nom unique et diverses propriétés chimiques. De même, la classe "Reaction" représente une réaction chimique, définissant ses substrats et produits, sa stoechiométrie et ses paramètres cinétiques. Le réseau de réaction lui-même est représentée par la classe "RDNetwork" qui contient une liste d'espèces et une liste de réactions. Enfin, le système chimique est représenté par la classe "RDSystem", et contient un réseau de réaction ("RDNetwork") et un état du système correspondant. Les réseaux et systèmes de réactions peuvent être initialisés par construction, via les constructeurs des différentes classes, mais également plus simplement via un dictionnaire Python ou un fichier JSON.

0.2.2 Modéliser un système de réaction-diffusion

Les systèmes biologiques présentent souvent plusieurs compartiments (organites, membranes, cytoplasme, espace extracellulaire ...). Par conséquent, il est nécessaire d'étendre les réseaux de réactions avec de la diffusion. La première étape est de diviser le volume de réaction en plusieurs sous-volumes interconnectés, entre lesquels les molécules vont pouvoir diffuser. Dans STReNGTHS, on envisage deux approches. Le plus simple consiste utiliser une grille de sous-volumes. Chaque cellule de la grille a la même taille et possède 1 à 6 voisins selon sa position et la taille de la grille. Cependant, utiliser une grille contraint à appliquer la même résolution spatiale à tout le système ce qui peut être inefficace dans certains cas. C'est pourquoi on considère l'autre cas de figure, qui consiste à utiliser un Graphe plus générique de sous-volumes. Chaque sous-volume est un noeud avec son propre volume, et est connecté à un nombre arbitraire de noeuds voisins, reliés par distance et une surface d'échange données. Cette seconde approche, bien que plus complexe à mettre en place, permet donc une modélisation plus optimale.

Puisqu'on a divisé le système en sous-volumes, il convient également de différencier les populations d'espèces de chaque sous-volume. On redéfinit donc l'état du système comme une matrice, avec les espèces en lignes et les sous-volumes/noeuds en colonnes. On peut modéliser la diffusion comme un processus de saut de particules/molécules d'un noeud à son voisin (réaction réversible du premier ordre) [4]. Le réseau de réaction-diffusion contient les réactions possibles dans chaque noeud, ainsi que toutes les réactions diffusives entre noeuds voisins.

Dans STReNGHTS, l'espace de diffusion est représenté par les classes "RDGridSpace" et "RDGraphSpace", correspondant aux deux cas de figures évoqués précédemment. Du reste, les classes impliquées son les mêmes, mais des arguments et méthodes supplémentaires permettent de définir les propriétés diffusives du système. En particulier, tout objet "RDSystème" contient notamment un objet "RDGridSpace" ou "RDGraphSpace" définissant sa topologie.

0.2.3 Simuler des trajectoires de système

Une fois le système chimique modélisé, on peut simuler son évolution dans le temps. Une simulation génère donc à proprement parler une trajectoire de l'état du système, c'est à dire l'état du système en fonction du temps, depuis un état initial. Les deux types d'approches supportées par STReNGTHS actuellement sont l'approche déterministe et les approches stochastiques.

L'approche déterministe consiste simplement à exprimer puis résoudre les équations de taux du système. On utilise la méthode d'Euler, qui est l'approche la plus basique pour cela.

Les approches déterministes sont l'algorithme de Gillespie [22] et son approximation, l'algorithme τ-leap [21]. La première, l'approche originale, consiste à effectuer une réaction aléatoire à la fois, avec un pas temporel aléatoire associé [21]. Elle consiste en fait à résoudre exactement, étape par étape, l'équation maîtresse du système [22]. L'algorithme τ-leap en est une approximation où le pas de temps peut être fixe et ou plusieurs réaction sont réalisées en même temps, mais toujours en proportions aléatoires [22].

STReNGTHS implémente ces trois méthodes de simulation. Les simulations peuvent être réalisées en via la fonction générique "simulate", ou bien en manipulant directement un moteur de simulation. Un moteur de simulation ("RDEngine") est une classe proposant l'implémentation d'une méthode de simulation spécifique avec une interface générale abstraite, indépendante de la méthode implémentée. Une telle approche, on l'espère, devrait permettre d'étendre plus facilement la palette d'outils de STReNGHTHS.

Une simulation, une fois complète, retourne un objet de la classe "RDTrajectory", représentant la trajectoire obtenue, et contenant tout les éléments nécessaires à as mise en contexte et à son analyse. Elle permet également de sauvegarder/charger facilement les résultats de simulation.

0.2.4 Fonctionnalités importantes

Afin de répondre aux exigences spécifiques des systèmes biologiques, STReNGTHS offre certaines fonctionnalités.

Des compartiments biologiques peuvent être modélisés via différents types de sous-volumes avec des propriétés diffusives différentes. On définit d'abord les différents environnements de diffusion (types de sous-volumes) dans le réseau de réaction. Les coefficients de diffusion et densités initiales des espèces chimiques peuvent notamment être définies en fonction de l'environnement de diffusion. En suite, on attribue à chaque sous-volume un environnement de diffusion. Par exemple, on peut définir des environnements "membrane", "cytoplasme" et "extérieur" pour représenter une section de membrane plasmique, avec certaines espèces ne diffusant que d'un côté, ou bien seulement au voisinage, de la membrane. Ce système est central car il permet précisément de représenter les systèmes spatialement inhomogènes propres à la biologie cellulaire.

Les systèmes biologiques fonctionnent typiquement loin de l'équilibre thermodynamique, du fait de l'activité métabolique qui maintient les concentrations de certaines espèces chimiques fixées. Cela peut être implémenté dans les modèles par un système de chémostat. Une espèce chémostatée voit sa quantité fixée durant les simulations. Les chémostats peuvent être définis dans un système de trois façons. D'abord, il peuvent être définis globalement. Les quantité d'espèces concernées sont alors fixées dans tout le système. Les chémostats peuvent également être définis selon l'environnement de diffusion. Les quantités des espèces concernées ne seront alors fixées que dans les environnements

en question. Cela permet d'intégrer facilement plus de finesse aux modèles. Enfin, ils peuvent être définis sous-volume par sous-volume, si nécessaire, ce qui garantit un contrôle total sur leur disposition.

0.2.5 Optimisation

Comme évoqué plus tôt, les systèmes de type graphe peuvent être plus avantageux du point de vue computationnel, mais sont aussi plus difficiles à définir. Pour palier à ce problème, STReNGTHS offre une fonctionnalité de coarse-graining de grille en graphe. Cette fonctionnalité repose sur l'utilisation d'une carte de réindexation, qui réassigne à chaque sous-volume de la grille de départ un nouveau noeud. Comme plusieurs sous-volumes de la grille peuvent être fusionnés en un même noeud, on obtient bien un graphe où la taille des noeuds, la distance qui les séparent, et la surface qui les connectent, peuvent varier. Une autre application de cette fonctionnalité est de filtrer les noeuds inutiles dans certains systèmes, ce qui à pour effet d'augmenter la vitesse des simulations.

0.2.6 Exemples Illustratifs

Nous présentons deux exemples illustrant l'utilisation de STReNGTHS, tirés directement de l'article qui lui est dédié [18].

Le premier exemple représente une scénario de transduction du signal dans un système 2D hétérogène représentant une coupe de cellule avec un milieu extracellulaire, une membrane, un cytoplasme, et un noyau. Un récepteur membranaire peut fixer un ligand extracellulaire, et le complexe en résultant peut à son tour activer un effecteur intracellulaire. On simule l'évolution de ce système avec les différentes méthodes (déterministes et stochastiques) supportées par STReNGTHS. On visualise ensuite la trajectoire globale de l'effecteur actif, ainsi que sa distribution spatiale à différents temps.

Le deuxième exemple montre la formation de motifs spatiaux de réaction-diffusion sous différentes topologies. Il utilise un réseau de réaction capable de faire de tels motifs spatiaux, similaire au modèle de Gray-Scott [38] ou à celui utilisé dans la référence [45]. On utilise l'algorithme τ-leap [21] pour rendre compte de la formation spontanée de motifs aléatoires depuis un état initial homogène. On étudie l'évolution de systèmes appliquant ce réseau de réaction sur différentes géométries, en 1D, dans une grille 2D plaine, dans une forme en 2D et en 3D sur une sphère. Dans les deux derniers cas, on intègre également des sources de réactifs supplémentaires pour complexifier les motifs.

0.3 GTPases

La transduction du signal est la fonction d'un système biologique consistant à produire un signal de sortie en réponse à un signal d'entrée. Dans les cellules, les systèmes de transduction du signal sont implémentés dans des réseaux de réaction-diffusion, et les signaux peuvent correspondre à la densité d'espèces chimiques spécifiques. Il est donc essentiel de bien identifier les espèces concernées.

Les GTPases de la superfamille de protéines RAS jouent un rôle important dans la transduction du signal [7]. C'est notamment le cas des petites GTPases [53] et de la sous-unité α des protéines G, partenaires des récepteurs couplés aux protéines G (GPCRs) [7].

Une GTPase est une hydrolase qui catalyse l'hydrolyse de la guanosine tri-phosphate (GTP) en guanosine di-phosphate (GDP), libérant un phosphate inorganique P_i. Liée au GTP, une GTPase

est capable d'interagir avec un effecteur en aval de la cascade de signalisation [44] : c'est la forme active [49]. C'est ainsi qu'une fois activée, une GTPase peut propager le signal. Cette étape requiert donc en amont l'activation de la GTPase depuis son état basal, liée au GDP [7].

La régulation de l'activité des GTPases est principalement assurée par trois types de partenaires protéiques, dont deux que nous traitons ici : Les facteurs d'échange nucléotidiques de la guanine (GEFs), activant les GTPases en accélérant l'échange de nucléotide [58], et les protéines activatrices de GTPases (GAPs), qui, contrairement à ce que leur nom suggère, inactivent les GTPases en renforçant leur activité enzymatique (hydrolytique) [58]. Avec les effecteurs de signalisation en aval, ces protéines constituent les partenaires des GTPases.

Dans ce chapitre, on étudie la transduction du signal par les protéines G dans différents système, tout en s'assurant de la consistance thermodynamique des modèles permettant les considérations énergétiques. Nous étudions notamment les mécanismes de régulation des GTPases responsables des propriétés de signalisation.

Nous proposons en particulier un modèle détaillé de transduction du signal par les GPCRs. Par rapport à des modèles cinétiques précédents ([28], [31], [51]), celui-ci à la particularité de simultanément tenir compte de la dissociation de G_α et $G_{\beta\gamma}$ en plus d'être thermodynamiquement consistent, et également d'avoir un jeu de constantes cinétiques contrôlé par seulement 3 paramètres libres.

0.3.1 Modélisation des réseaux de réaction des systèmes de transduction du signal par les GTPases

Dans cette section, nous présentons différents réseaux de réactions associés a des systèmes de transduction du signal de plus en plus complexes. Chaque réseau de réaction est décrit par un système d'équations stoechiométriques réversibles, où chaque réaction représente la formation d'un complexe.

Le premier système considéré est le plus simple. Il s'agit du cas d'une GTPase seule (sans partenaire protéique), interagissant avec ses substrats nucléotidiques. Le réseau comporte 4 réactions: L'association du GTP et du GDP avec la GTPase, l'hydrolyse catalysée du GTP lié ([65] [16]), et l'hydrolyses non catalysée du GTP libre. Ces quatre réactions forment le cycle GTPase.

Ce système peut être étendu par l'ajout d'un partenaire protéique.

Cela peut être un partenaire générique, capable de s'associer à la GTPase quel-que-soit son état. En résulte un système avec deux cycles de GTPase, couplés par les réactions d'association du partenaire.

Le partenaire peut aussi être strictement spécifique d'un état donné de la GTPase (liée au GTP ou au GDP). Dans ce cas, il n'y à pas de cycle GTPase supplémentaire : le réseau est simplement étendu par une seule réaction.

On peut généraliser cette logique à un nombre arbitraire de partenaires génériques ou strictement spécifiques, en obtenant des réseaux plus ou moins larges selon le type et le nombre de partenaires.

Finalement, on établit la structure du réseau de réaction associé à la transduction du signal par les GPCRs. Ce système peut être modélisé comme un système de GTPase [28]. C'en est un. Ce dernier réseau est considérablement plus complexe que les précédents.

La GTPase G_α [7] peut interagir avec le récepteur R, qui peut lui même se lier à son ligand agoniste L. L'activation du récepteur par son ligand lui confère une fonction de GEF, le rendant capable d'activer la sous-unité G_α [11] [52]. G_α peut également interagir avec la sous-unité $G_{\beta\gamma}$ pour former la protéine G trimérique, ayant une conformation plus appropriée pour interagir avec le récepteur [48]. L'activation de G_α entraîne sa dissociation de $G_{\beta\gamma}$ [61], et chaque sous-unité s'associe avec son propre effecteur de signalisation en aval, E et E' pour G_α et $G_{\beta\gamma}$ respectivement. Le modèle suppose que ces interactions sont compétitives. De plus, E est traité comme un partenaire strictement spécifique de G_α-GTP, du fait de la spécificité des effecteurs de GTPases pour leur état active [44].

0.3.2 Mise en place des contraintes thermodynamiques

Les processus biochimiques dans les cellules fonctionnent généralement loin de l'équilibre thermodynamique. Pour comprendre les mécanismes biochimiques dans leur dimension énergétique, il est donc nécessaire de bien identifier les sources d'énergies pouvant conduire ces système loin de l'équilibre. Pour ce faire, le principe du bilan détaillé, ou de la réversibilité microscopique, doit être respecté. Celui-ci impose que toutes les réactions soient réversible et que le système admette un état d'état stationnaire d'équilibre où chaque transformation à un courant net nul.

Une approche proposée par Colquhoun et al consiste à dériver des cycles de réaction fondamentaux des contraintes entre constantes cinétique (ou entre constantes d'équilibre) [8]. Pour assurer le respect du bilan détaillé dans les différents modèles, Nous utilisons cette méthode, étendue avec une approche composite pour plus facilement résoudre manuellement les contraintes en hiérarchisant les réseaux de réaction. On commence par traiter les réseaux les plus simples. On arrive alors facilement à résoudre et appliquer leurs contraintes pour définir leurs constantes d'équilibre contraintes en fonction de celles libres. Ceci étant fait, on traite les réseaux légèrement plus complexes. Ces derniers contiennent une ou plusieurs instances des réseaux plus simple, et donc leurs contraintes respectives. Autrement dit, l'ensemble des contraintes sur un réseau donné peut s'écrire comme la somme des contraintes de chacun de ses sous-réseaux (déjà caractérisés), plus des contraintes de couplage. Ainsi, en procédant des réseaux les plus simples (cycle GTPase) vers les réseaux les plus complexes (système GPCR), nous déterminons et résolvons manuellement l'ensemble des contraintes thermodynamiques associées à chaque réseau de réaction beaucoup plus facilement.

Pour chaque réseau de réaction, on vérifie également l'exactitude des contraintes numériquement, en attestant de la disparition des courants à l'état stationnaire avec les contraintes.

0.3.3 Valeurs de paramètres

On souhaite étudier le fonctionnent des modèles avec des paramètres réalistes du point de vue biologique. Pour cela, nous avons rassemblé des valeurs issues de la littérature scientifique. En particulier, les concentrations des espèces conservées (GTPases, récepteurs, effecteurs) et chémostatées (GTP, GDP, P_i), et des paramètres cinétiques de base à partir desquels interpoler les autres. On rapporte et justifie le choix des valeurs.

0.3.4 Transduction du signal

Comme évoqué plus tôt, un système de transduction du signal comporte un signal d'entrée et un (deux dans le cas des GPCRs) signal de sortie. Ce type de système peut être décrit par une fonction

de transduction du signal $\mathcal{S}$ exprimant le signal de sortie en fonction du signal d'entrée et du temps (et éventuellement des autres paramètres du système). Typiquement, le signal de sortie correspond à la densité normalisée d'une espèce, prenant donc une valeur entre 0 (pas de signal) et 1 (signal maximal). Le choix de l'espèce dépend du système.

Dans le cas du système GTPase le plus simple, il convient naturellement de prendre la densité de GTPases actives, normalisée par la densité totale de GTPases [50].

Si le système inclut seulement une GTPase et un effecteur de signalisation, alors il peut être plus pertinent de prendre la densité d'effecteurs actifs (liés aux GTPases actives), normalisée par la densité totale de GTPases.

Si le système contient également d'autres partenaires, alors il faut prendre en compte toutes les formes de complexe GTPase-effecteur susceptibles de propager le signal, ce qui implique de tenir compte des inhibitions.

Notamment, dans les systèmes où l'effecteur n'est par explicitement présent, il convient tout de même de prendre en compte la potentielle compétition des autres partenaires avec ce dernier.

La nature du signal d'entrée dépend d'avantage du système. Il peut s'agir de la densité de GEF introduite dans un système par exemple.

Dans le cas spécifique du système GPCR, le signal d'entrée est la densité de ligand agoniste introduite dans le système, et les deux signaux de sortie sont les densités de G_α et $G_{\beta\gamma}$ liés à leurs effecteurs respectifs, normalisées par la densité totale des protéines G.

0.3.5 Résultats et discussion

0.3.5.1 Régulation des GTPases

Dans un premier temps, on étudie les mécanismes de régulation des GTPases. Notamment, on introduit la notion importante de régime. Un régime de paramètres cinétiques est une région des paramètres d'un modèle correspondant à une propriété biologique particulière.

On se penche d'abord sur le cas de la GTPase seule. Son régime, en absence de régulation et avec les paramètres de la littérature, est appelé régime de base. Comme attendu, on observe que se régime de base donne un signal basal bas, proche de 0. À partir du régime de base, on voit ensuite comment une variation des paramètres cinétique, dans la mesure permise par les contraintes thermodynamiques, permet l'accès à de nouveaux régimes. On voit notamment comment différents régimes associés à l'effet d'une GEF, c'est à dire l'accélération de l'échange de nucléotide, aboutissent en effet à une activation des GTPses, tandis que l'accélération de l'activité hydrolytique, associée au GAP, aboutissent bien à l'opposé à leur inactivation. On montre notamment que ces effets sont purement dû aux conditions de non-équilibre, et n'existent pas à l'équilibre thermodynamique. Ces observations posent les bases de la compréhension de la régulation des GTPases, nécessaires pour comprendre des systèmes plus complexes.

On affine ensuite cette compréhension en étudiant des systèmes où les changements de régime sont introduits par l'association de la GTPase avec des partenaires protéiques. On étudie notamment à nouveau le cas des GEFs et GAPs sous cet angle, en tenant compte notamment des effets de compétition potentielle avec les effecteurs de signalisation. On relève qu'un effet inhibiteur possible

des GEF, relevé dans une publication antérieure [50], est lié d'avantage à l'intensité de l'activité des GEFs qu'à leur quantité, ainsi que à leur compétition avec l'effecteur de signalisation.

Un partenaire générique de GTPase peut tout-de-même présenter une préférence pour un certain état de la GTPase. Une conséquence extrêmement importante découlant des contraintes thermodynamiques est qu'introduire une préférence du partenaire pour un état de la GTPase (lié au GTP ou au GDP) requiert d'introduire cette même préférence pour le complexe vis-à-vis du nucléotide correspondant.

On montre également que les partenaires strictement état-spécifiques sont de plutôt bonnes approximations de leurs équivalents génériques avec préférence forte pour un état de la GTPase.

0.3.5.2 Le système GPCR

À la lumière des éléments de compréhension issus dés étapes précédentes, on peut désormais s'atteler à l'étude de la transduction du signal par les GPCRs et protéines G. Dans ce système, il y a un signal d'entrée, la quantité de ligand, et deux signaux de sortie, les proportions de G_α actif et $G_{\beta\gamma}$ liés à leurs effecteurs de signalisation respectifs.

L'ensemble des paramètres cinétiques du modèle est contrôlé (notamment par l'intermédiaire des contraintes thermodynamiques) par trois paramètres libres: ϕ, δ et γ. ϕ mesure la force de l'activité de GEF du GPCR actif. Plus il est élevé par rapport à 1, plus l'activité GEF est forte. δ détermine la préférence de $G_{\beta\gamma}$ pour G_α-GDP par rapport à G_α-GTP. Plus il est élevé par rapport à 1, plus cette préférence est forte. γ reflète la cohésion de la protéine G trimérique, soit l'affinité du complexe G_α-GDP-$G_{\beta\gamma}$. Plus il est élevé par rapport à 1, plus cette affinité est forte par rapport à celle de G_α pour son effecteur E.

Comme le système à deux signaux de sortie, il existe 3 façons de maximiser le signal. Soit on maximise seulement le signal de G_α, soit seulement celui de $G_{\beta\gamma}$, ou bien soit les deux en même temps.

À chaque façon de maximiser le signal, on peut théoriquement associer un régime. On les nomme régimes α, β et $\alpha\beta$ respectivement. Par minimisation stochastique du signal, on parvient à confirmer l'existence de ces trois régimes distincts et à les localiser dans l'espace des paramètres.

Dés lors, on peut caractériser chaque régime. Les régimes α et β sont opposés, tandis que le régime $\alpha\beta$ est entre les deux, tant en terme de signal que de valeurs de paramètres.

On commence par établir le profil du signal en fonction du signal d'entrée et du temps, pour chaque régime. Les régimes diffèrent notamment par leur signal de base. On note que ce profil du signal peut être réutilisé dans des modèles plus simples.

Comme montré avec les systèmes précédents, la transduction du signal par les GPCRs, qui vue plus simplement consiste en la régulation d'une GTPase par une GEF, dépend des forces de non-équilibre. On mesure les variations de signal ainsi que la consommation en GTP du système en fonction de la force de conduction hors-équilibre imposée au système par la densité de GTP chémostatée. Sur un échelle logarithmique en abscisse, les profils des signaux et de la consommation de GTP sont des courbes sigmoïdes, trouvant leur point d'inflexion aux abords du ratio cytoplasmique. En deçà, il n'y a pas de signal, ni de consommation d'ATP. On montre alors que les conditions énergétiques cytoplasmiques sont optimales pour la signalisation, et ce quel que soit le signal et le régime.

On étudie également les fluctuations du signal en simulant l'évolution du système avec l'algorithme de Gillespie [22]. On étudie en particulier l'effet de la variation des quantités totales des différentes espèces fondamentales du système sur les fluctuations.

Enfin, on s'intéresse à la variation de la densité de $G_{\beta\gamma}$ par rapport à celle de G_α. Jusqu'ici, on considérait toujours des situations où les deux sous-unités existaient en proportions égales. On montre que tandis que le signal $\beta\gamma$ et proportionnel à la densité de $G_{\beta\gamma}$, le signal α évolue de façon non-linéaire et particulièrement contre-intuitive. En effet, celui-ci augmente à mesure que la concentration en $G_{\beta\gamma}$ diminue, avant de disparaître quand elle devient nulle.

0.4 Conclusion

Nous avons développé STReNGHTS, un outil de simulation facile à utiliser. Ce dernier propose une interface intuitive pour modéliser des réseaux de réaction-diffusion et simuler leur évolution dans le temps avec des méthodes déterministes et stochastiques. Son interface abstraite pour les algorithmes de simulation le rend propice à l'ajout de nouveaux algorithmes ou de nouvelles implémentations. La possibilité d'émuler la présence de différents compartiments diffusifs dans les réseaux et systèmes de réaction-diffusion, et de définir finement la résolution spatiale en font un outil particulièrement approprié pour étudier les systèmes biochimiques. Cependant, il reste encore beaucoup à faire pour améliorer ce dernier, comme en particulier mettre en place

- des méthodes plus avancées de résolution des équation de taux,
- des pas de temps adaptatifs, en particulier pour l'algorithme τ-leap [21],
- des implémentations parallèles via l'utilisation du processeur graphique (GPU), pour accélérer les simulations avec beaucoup de sous-volumes.

La seconde partie concerne l'étude des systèmes de transduction du signal par les GTPases. Nous avons vu comment l'activité des GEFs et des GAPs était essentiellement un processus de non-équilibre. Nous avons également discuté des conséquences des contraintes thermodynamiques sur la préférence des partenaires enzymatiques pour un état de la GTPase plutôt qu'un autre, à savoir comment cette préférence se répercutait sur l'affinité de la GTPase pour ses différents substrats. Nous avons ensuite pu étudier le cas plus complexe de la transduction du signal par les GPCR. Nous avons caractérisé les 3 régimes maximisant l'un, l'autre ou les deux signaux de sortie du système. Après avoir discuté des différences de signalisation entre les régimes, nous avons pu monter comment le système avait un fonctionnement optimal dans les conditions de non-équilibre cytoplasmiques. Nous avons également montré que, de manière contre-intuitive, diminuer la densité $G_{\beta\gamma}$ augmentait le signal de G_α, jusqu'à un certain point. Pour étudier tout ces systèmes, nous avons construit des modèles de complexités diverses, et en particulier un modèle complet de signalisation par les GPCRs.

Cependant, ces travaux restent théoriques et bénéficieraient grandement d'une perspective expérimentale complémentaire. La suite logique de ce travail serait de considérer l'impact de la diffusion sur ces systèmes. Dans le cas des GPCR, il serait également intéressant d'étudier le trafic membranaire [62] [1](848-849) et la régulation du réseau par les arrestines et autres protéines régulatrices [1](pages 832, 848-849).

Pour assurer la consistance thermodynamique de nos modèles, nous avons étendus une méthode

issue de travaux précédents [8], via une logique de hiérarchisation des sous-réseaux et de leurs contraintes. Cependant, bien que cette approche offre des avantages, elle nécessite toujours de résoudre manuellement certaines contraintes. Aussi, une approche plus automatique serait la bienvenue pour les travaux futurs.

Il serait également intéressant de considérer d'autres critères de consistance physique des réseaux de réaction-diffusion, comme les limites des constantes d'association dues à la diffusion.

Chapter 1

Introduction

In this thesis, we study far-from-equilibrium biochemical systems with physical models. The living relies on a complex chemistry, which involves the diffusion of diverse species in compartments delimited by phospholipid bilayers. These species commonly include aminoacids, proteins, lipids, carbohydrates, nucleotides, ribonucleic acids, ions and other types of molecules. Biochemical processes work far from thermodynamic equilibrium, due to the level of some chemical species being tightly regulated in the cytoplasm by the cell metabolism. This is the case for instance in signal transduction systems, which allow a cell to sense and respond to some chemical signal. The way driving forces impact biochemical systems is complex. It requires identifying where they come from, which species are involved and in which quantity. Moreover, in a system, all non-equilibrium phenomena are driven by some well-defined and identified driving force. When those forces are turned off, the system should relax to an equilibrium state.

In this work, we aim to provide a thermodynamically consistent perspective on signal transduction systems involving GTPases and G Protein Coupled Receptors (GPCRs). Our goal is to understand how the different regulating effects in those systems depend on the non-equilibrium conditions. We will have to build thermodynamically consistent kinetic models for different systems involving GTPase and their signaling pathways and effectors. Furthermore, we aim to provide a characterization of the different non-equilibrium regimes of those signal transduction systems. To study this kind of biochemical systems, specific software are necessary, especially for modeling and simulation. In this thesis, we developed our own tools to respond to the specific requirements of our studies. In particular, we developed a Python package for modeling and simulating such biochemical reaction-diffusion networks, allowing to define complex landscapes with diffusion in multiple compartments.

In cell biology, signal transduction refers to the production of an intracellular output chemical signal as a response to some different input signal, usually sensed from the outside of the cell by some receptor in the plasma membrane [9]. Those systems comprise cascades of reactions, involving mostly enzymes, and are at the base of signaling pathways, which are a central topic in cell biology. A particularly interesting case of signal transduction is eukaryotic chemotaxis, which consists in cells migrating following the gradient of a chemical signal, perceived through membrane receptors. Signal transduction is involved in multiple functions of cells, such as growth, proliferation, differentiation

or polarisation, and generally has consequences on gene expression and protein trafficking [9] [46].

As open systems, in the thermodynamic sense, i.e. exchanging molecules with the outside, these biochemical reaction networks usually evolve towards a state, which can be either thermodynamic equilibrium or a non-equilibrium steady state. At equilibrium, all reactions are reversible, and there are no net currents, neither inside the system nor between the system and the exterior. This condition requires microscopic reversibility of all the underlying reactions, which in equilibrium are balanced in the forward and backward directions. This condition is known thermodynamically as detailed balance (DB).

Biochemical systems mostly work far from equilibrium. Cells consume nutrients to regenerate an internal gradient of ATP and ADP which then drives other cellular processes into their working, far-from-equilibrium behavior. However, if those chemostats are turned off, the systems should relax to a state of equilibrium, where DB is respected. To build thermodynamically consistent models of biochemical processes it is necessary to account for this specific property.

The study of biochemical systems requires thermodynamically consistent modeling approaches as well as appropriate methods to simulate their behavior. When a system can be described through a finite number of chemical species and transformations, one way to represent those interactions is as a reaction-diffusion network. Chemical reaction networks are usually described with a set of coupled stoichiometric equations [56]. From these stoichiometric equations, one may derive both rate equations and the master equation of the system. The trajectories obtained with this approach are stochastic in nature. On the one hand, this means that individual realizations may be substantially different even when they start off from identical initial conditions. Furthermore, the average values of certain molecular species in the non-equilibrium steady state may be different from the steady state values of the associated deterministic rate equations. Cellular conditions typically provide environments that are determined by fluctuations, making stochastic methods the biologically relevant investigation tools for signal transduction. For example, bistable systems or systems with multiple attraction basins, typically display noise-driven dynamical phase transitions. These, absent by construction in a rate-equation-based modeling approach, may have important biological implications, which can only be investigated through stochastic methods.

Signal transduction systems involve not only the formation of complexes occurring in a cascade of steps that essentially follow the patterns of non-equilibrium steady currents in specific, intricate biochemical networks. They are also built around the mobility of selected molecular species, which can be passive diffusive transport or active, energy-consuming directed transport, across multiple compartments separated by biological phospholipid membranes. The fine-tuned nature of cellular responses relies on the subtle intertwining of chemical reactions and mass transport with the local environment, where chemical potential gradients (e.g. di and triphosphate nucleotides) are usually sustained. It appeared thus natural, in line with the objectives of this Ph.D. project, to set up more and more complex reaction-diffusion simulations, both stochastic and deterministic, to investigate the subtle interplay of regulatory dynamics in chemical and physical space. This effort has culminated with a complete, flexible and powerful software package that we designed and built in order to allow one to easily set up elaborated simulations of many-body reaction-diffusion (RD) systems in complex landscapes (i.e. multiple biological compartments). We termed our simulation tool STReNGTHS, and made it available as an open-source Python package, internally optimized with compiled C++ implementations of the most computationally demanding processes and designed to be easily extended with new methods. This software, along with the underlying

concepts of reaction-diffusion network modeling and simulation, are the object of chapter one of this manuscript.

Signal transduction systems involve many different chemical species. However, while some are generic, involved in most biochemical processes (i.e. phosphate nucleotides), other enzymes are characteristic of signal transduction systems, where they play a central role. Among the enzymes heavily involved in signal transduction, two types of enzymes stand out: GTPases from the Ras superfamily [7] and G Protein Coupled Receptors (GRCRs) [11]. The former are GTP hydrolases which are active or inactive depending on their substrate. When activated by upstream regulating proteins, they propagate the signal downstream through downstream effectors. Ras GTPases include the small GTPases evoked earlier, but also G proteins, and more especially their G_α subunits which are regulated by active GPCRs. This brings us to the second family of signal transduction proteins: GPCRs. GPCRs are trans-membrane receptors involved in a large number of signal transduction processes. Upon association with an agonist ligand, the input signal, they activate G proteins, which can then propagate the signal downstream. A G protein has two subunits. One is G_α, a Ras GTPase as we evoked earlier [7]. The other is the $G_{\beta\gamma}$ dimer, which is not a GTPase. It acts both as a regulator of G_α [39], and as a signal transducer with its own downstream effectors [11]. A minimal GPCR signal transduction system must contain the GPCR and its ligand, as well as the G protein.

In order to study how non-equilibrium affects GPCR signaling, we need a sufficiently detailed and thermodynamically consistent model. The fundamental reaction scheme for GTPase is well known [16] [65]. Reaction networks for the regulation of GTPases have been the object of many previous studies [50] [41], which highlighted the non-equilibrium effects associated with GTPase systems and other similar reaction networks [41] [10]. However, if one wishes to model GPCR-based signaling pathways more thoroughly, one needs to derive more complex models. In such models, it is interesting to consider the full dissociation of the G_α and $G_{\beta\gamma}$ subunits [61], which is often not accounted for in kinetic models [51].

In the second chapter, we study GPCR and GTPase-based signal transduction systems from a theoretical and computational perspective. In particular, we introduce a kinetic model of GPCR signaling which accounts for the dissociation of the G_α and $G_{\beta\gamma}$ subunits and effector association. Furthermore, we introduce an original modular approach to enumerate the detailed balance constraints. This method, which builds on previous works [8], allows one to recast large reaction networks as the composition of more fundamental patterns.

Our original method, applied to signaling networks of growing complexity, leads us to rationalize the effects of the main molecular regulators of GTPase. Furthermore, we introduce the appropriate, biologically consistent measures of GTPase-mediated signals and investigate the associated signal transduction profiles. Moreover, we address the question of the energy cost associated with a specific signal, showing that natural cytoplasmic conditions appear to provide an optimal environment for signal transduction in a noisy, non-equilibrium environment that relies on chemostatting phosphate-associated nucleotides.

Chapter 2

Modeling biochemical reaction-diffusion systems with STReNGTHS

2.1 Introduction

The behavior of living systems results from the activity of complex biochemical reaction-diffusion (RD) networks. These networks involve all kinds of species, including ions, nucleotides and enzymes, diffusing within and across different compartments, generally separated by lipid bilayers (i.e. the plasma membrane). The biophysical study of these systems involves the construction of physical models whose mathematical analysis is able to provide a quantitative description of their behavior. A reaction-diffusion network may be modeled with a set of coupled stoichiometric equations describing chemical transformations operating on a set of chemical species. Such a network, coupled with diffusion, may then describe a reaction-diffusion system. The evolution in time of such a system can be described by the corresponding set of coupled partial differential equations, whose integration describe the time evolution of the system once the appropriate initial and boundary conditions have been specified. Stochastic methods, such as the Gillespie algorithm [22], can also be used to simulate the evolution in time of reaction-diffusion systems. As opposed to the deterministic approach, they allow one to account for the fluctuations associated with the finite number of particles in the system [22]. This allows to explore multistable regimes inaccessible at the mean-field level [5]. As the size and complexity of the reaction-diffusion networks increase, with numerous chemical species diffusing across multiple compartments, general-purpose tools designed for simple systems reach their limits. In this case, specifically designed dedicated computational tools become a necessity.

STReNGTHS (**S**imulation and modeling **T**ool for **Re**action-diffusion **N**etworks in **G**raphs and **T**ridimensional **H**eterogeneous **S**ystems) is an open-source software for modeling reaction-diffusion systems and simulating their trajectories, interfaced as a Python package, that was developed during this PhD. It allows one to model reaction-diffusion systems with different chemical species and transformations, and handle discrete diffusion spaces made of interconnected reaction volumes - or cells, which can

be assigned different properties to simulate different biological compartments.

While developing STReNGTHS, one important question emerged concerning the type of interface we should use. There are multiple ways to interface software for users. Nowadays, most will be familiar with Graphical User Interfaces (GUI), relying on responsive visual elements in windows (buttons, text area, canvas, etc.). However, not all programs work with GUI. With Command Line Interface (CLI) programs for instance, which should be especially familiar to Linux users, the program gets instruction through text inputs in a console. Eventually, we chose to develop STReNGTHS as a Python package. Such a kind of software relies on a third type of interface called an Application Programming Interface (API), where one interacts with the software using a programming language, through function calls and programming objects. This approach is the most versatile, as users can easily build their own input and output processing routines and combine the software with other tools. This kind of approach seemed the most suitable for STReNGTHS, as it allows to programmatically define reaction-diffusion networks and facilitates the analysis of simulation data using the vast panel of scientific packages available for Python. Many other similar software also made a similar choice [64] [29].

There exist other similar tools for modeling and simulating biological systems, designed around different architectures, each with their stretch of flexibility, pros and cons [18]. These include BioNetGen [26], ReaDDy [29], STEPS [64] and MesoRD [27]. Despite many of the available computational tools available to simulate RD systems are rather sophisticated, it appeared to us that the typical learning curve associated with more and more complex situations grew rapidly steep. As a consequence, confronted with the need of a simple tool, yet flexible enough so that more complex scenarios could be handled as effortlessly as possible, we found ourselves developing our own. This has now grown to a mature, open-source tool that has been released to the community [18]. We note that the architecture of our software has been explicitly designed so that it is easy for users to extend our tool and enrich it with new features.

In this chapter, we will present the STReNGTHS package [18], along with the theoretical aspects of reaction-diffusion systems kinetic modeling that are involved. For the sake of clarity, the API of the package will be presented in a non-exhaustive manner here. However, a full specification can be found in the API Reference section of the official documentation (`https://strengths.readthedocs.io/en/latest/apiref.html`). The first section quickly explains how physical quantities are handled with STReNGTHS. In the second section, we introduce the basic theoretical concepts related to pure reaction networks and systems, and explain how these concepts are implemented in STReNGTHS. In particular, we introduce some of the main modeling classes of the package. We extend the presentation in the third section with aspects realated with space and diffusion, involving modeling the system space as a network of cells. The fourth section is about simulation. The different deterministic and stochastic methods implemented in STReNGTHS are described in detail. Concerning the package, we then present how simulations are handled with an abstract interface, and how the simulated trajectories are structured and can be visualized. The fifth section is dedicated to two key modeling features of the software, reaction-diffusion environments and chemostats, while in the sixth section, we present the space coarse-graining feature which can help optimize some simulations. The capacities of the software are illustrated with more complex examples in the seventh section. Finally, before concluding, we briefly discuss matters related to the source code of the project.

2.2 Notations and use of symbols across the chapter

Across this chapter, and especially in the more theoretical parts, we use many different symbols and precise notations and writing conventions. While these are of course introduced gradually in the main text, it is useful to summarize them at the start for the reader's use.

- n, m, ω and λ: Number of elements in sets. Specifically, for a given reaction-diffusion system, n is the number of chemical species, m, the number of reversible reactions (except diffusive jumps), and ω, the number of spatial cells. As for λ, it is the number of samples in a discrete trajectory.
- i and j: Indices used for chemical species.
- r: Index used for reversible chemical reactions.
- α and β: Indices used for spatial cells.
- u: Index used for samples in discrete trajectories. By extension, it is also used as the iteration index in simulation algorithms.
- τ: Time step.
- s and p: Reaction stoichiometric coefficients for substrates and products of a reaction, respectively.
  - s_{ri} and p_{ri}: Substrate and product stoichiometric coefficients of a species X_i for a reversible reaction r, in the direction in which the reaction is written.
  - s_i and p_i: Substrate and product stoichiometric coefficients of a species X_i, when there is only one reversible reaction ($m = 1$).
- σ: Contact surface between cells.
  - $\sigma_{\alpha\beta}$: Contact surface between cells α and β.
- d: Distance between two cells.
  - $d_{\alpha\beta}$: Distance between cells α and β.
- V: Total volume of a system.
- v: Volume of a cell.
  - v_α: Volume of the cell α.
- X: Chemical species.
  - X: Full set of all the chemical species in a reaction-diffusion network.
  - X_i: Refers specifically to the species X_i
  - $X_{i\alpha}$: Species X_i in the cell α.
- x: State of the system, quantity of all chemical species.
  - x: Whole system state matrix.
  - $x_{i\alpha}$: Quantity of the species X_i in the cell α.

  - $x_i = x_{i,0}$ when there is only one cell ($\omega = 1$).
  - $x(t)$: Whole system state (matrix) at a given time t.
  - $\overrightarrow{x}$: Sequence of successively sampled system states in a discrete trajectory.
- $[X]$: Concentration of chemical species.
  - $[X]$: Matrix of the concentrations of every chemical species in every cells of a reaction-diffusion system. It is the system state expressed in terms of concentrations.
  - $[X_{i\alpha}]$: Concentration of the species X_i in the cell α.
  - $[X_i] = [X_{i,0}]$ when $\omega = 1$.
- D: Diffusion coefficient.
  - D_i: Diffusion coefficient of the species X_i.
  - $D_{i\alpha}$: Diffusion coefficient of the species X_i in the cell α.
- k: Reaction rate constant.
  - k_r^+ and k_r^-: Forward and reverse rate constants for the chemical reaction r.
  - $k_{r\alpha}^+$ and $k_{r\alpha}^-$: Forward and reverse rate constants for the chemical reaction r in the cell α.
  - $k_{i\alpha\beta}$ and $k_{i\beta\alpha}$: Forward and reverse rate constants for the jump process that takes a species X_i from the cell α to its neighboring cell β [4].
- Φ: Reaction current.
  - Φ_r^+ and Φ_r^-: Forward and reverse currents for the chemical reaction r.
  - $\Phi_{r\alpha}^+$ and $\Phi_{r\alpha}^-$: Forward and reverse currents for the chemical reaction r in the cell α.
- e, μ, ν: Event, in the context of the Gillespie algorithm [22].
- a: Reaction propensity in the Gillespie algorithm [22]. It has the same indexing system as the rates k.
  - a_e: Propensity (Gillespie algorithm [22]) for a given event e.
  - a_0: Sum of all the propensities at a given time, in the Gillespie algorithm [22].
- q: Event quantities for the τ-leap algorithm [21]. It has the same indexing system as a and k.
  - q_e: Quantity (τ-leap algorithm [21]) for a given event e.
- $\mathcal{N}(\alpha)$: Defines the set of neighbor cells of a given cell α.
- $\mathcal{S}$: Stoichiometric matrix.
- $\mathcal{L}(i)$: Modified discrete Laplacian for a given species X_i.

In this chapter, we use zero-based indexing. As an example, a set X of n species would be noted $\{X_0, \ldots, X_{n-1}\}$ rather than $\{X_1, \ldots, X_n\}$.

2.3 Classes for manipulating physical quantities

As STReNGTHS is focused on biochemical reaction-diffusion systems, it commonly involves manipulating physical quantities such as concentrations, reaction rate constants, diffusion coefficients, distances, volumes or times for instance. STReNGTHS implements the `UnitValue` class to represent physical quantities. It has the following API:

- `UnitValue(value, units)`: Constructor. Each argument sets the corresponding attribute.
- `value`: Numeric value (float) of the quantity.
- `units`: Units of the quantity (a `Units` object).

Below are a few examples of construction of UnitValue objects in Python:

1 μM	`UnitValue("1 µM")`
1 m^3/s	`UnitValue("1 m3/s")`
1 molecule/μm^3/s	`UnitValue("1 molecule/µm3/s")`

The `Units` objects themselves, which represent the units associated with the quantity, have two components, one - the units system - for the units associated with each fundamental dimension and one for the exponent associated with each fundamental dimension - the unit dimensions. A RD system is completely specified by setting units for:

- Space (L): nm, µm, mm, m, ...
- Time (T): ms, s, h, ...
- Quantity of matter (N): molecules, mol, ...

Units of fundamental dimensions can be combined with different exponents to form composite units:

units	units system	units dimensions
mol/m^3/min	{m, min, mol}	{-3, -1, 1}
nM^{-1} s^{-1}	{dm, s, nmol}	{3, -1, -1}

The main purpose of this class is to automatically handle unit conversions and operations involving physical quantities. For example:

```
v1 = UnitValue("1 µm2/s")
v2 = v1.convert("m2/h")
```

The class also overloads different operators to implement convenient operations between physical quantities, for instance:

```
v = UnitValue("1 µm2/s") + UnitValue("1 µm2/s") # v = 2 µm2/s
v = UnitValue("1 mol/m3") * 3 # v = 3 mol/m3
v = UnitValue("1 m")**2 # v = 1 m2
v = UnitValue("1 s") + UnitValue("1 min") # v = 61 s
```

Vector quantities can be conveniently dealt with through the `UnitArray` class. For example

```
#declaration
a = UnitArray([1,2,3,4], "mol/s")
#conversion
b = a.convert("mmol/s") # b = [1e3, 2e3, 3e3, 4e3] mmol/s
#operations
c = a * 2 # c = [2,4,6,8] mol/s
d = UnitArray([1,2], "m") / UnitValue("1 h") # d = [1,2] m/h
```

`UnitArray` objects internally store values as Numpy arrays [25].

The `UnitArray` and `UnitValue` classes can also be cast as strings. This also means that they can be handled by Python's built-in print function. As an example,

```
print(UnitValue("1 m"))
print(UnitArray([1,2,3], "m"))
```

would output

```
1.0 m
[1. 2. 3.] m
```

2.4 Modeling reaction systems

2.4.1 Fundamental concepts

This section reviews the key concepts of the mathematical modeling of chemical reaction networks [56].

2.4.1.1 Chemical species

Biochemical systems usually consist of molecules of variable size, or molecular complexes, diffusing in aqueous solutions and/or within or across phospholipid bilayers. These molecules can engage in reactions/chemical transformations. A system usually contains multiple copies of the same kind of molecule or complex. Let us refer to such a type of molecule/complex as a chemical species (or in short, species). As an example, adenosine triphosphate (ATP), a molecule present in high quantities in the cytoplasm of living cells, is a chemical species. The complex between a receptor and its ligand, while not covalent, forms a functional unit, and diffuses as one unique particle, and thus can be designed as a species, as would be the free receptor and ligand. We denote with X the set of the n chemical species that are comprised in a given RD network,

$$X = \{X_0, \ldots, X_{n-1}\} \tag{2.1}$$

2.4.1.2 Reactions

A chemical transformation, or reaction, is a discrete event in time and space during which a particular combination, in nature and quantity, of particles interacts in such a way that their natures

and/or quantities are partially or completely modified. It is important to understand that biochemical reactions do not necessarily involve the formation or breaking of covalent bonds. The formation of non-covalent molecular complexes or significant changes in the conformation of large molecules are both considered reactions. In other words, a given combination of input molecules, the substrates, interact to form another combination of (output) molecules, the products. The quantities of particles of a given species required as substrate and produced by a reaction are respectively referred to as substrate and product stoichiometric coefficients. The reaction is also characterized by a rate constant, whose units depend on the substrate stoichiometry. A reaction is usually written down as a stoichiometric equation [56], as

$$\sum_{i=0}^{n-1} s_i X_i \xrightarrow{k} \sum_{i=0}^{n-1} p_i X_i \tag{2.2}$$

where k is the reaction rate constant and s_i and p_i the substrate and product stoichiometric coefficients associated with the species X_i, while the arrow shows the direction of the reaction associated with k, from substrates to products. The dimension of the reaction rate constant k is defined for a given reaction as

$$T^{-3}(NL^{-3})^{1-o} \tag{2.3}$$

where o is the reaction order, which corresponds to the number of substrate particles required for the reaction, simply defined as

$$o = \sum_{i=0}^{n-1} s_i \tag{2.4}$$

When studying thermodynamically consistent systems, all reactions are reversible. This means that a given forward reaction is paired with a reverse reaction with reverted stoichiometry. Rather than being written separately, pairs of reverse reactions can be conveniently written as

$$\sum_{i=0}^{n-1} s_i X_i \underset{k^-}{\overset{k^+}{\rightleftharpoons}} \sum_{i=0}^{n-1} p_i X_i \tag{2.5}$$

where k^+ and k^- are respectively the rate constants of the forward and reverse reactions.

2.4.1.3 Reaction network

A reaction network describes the different species that populate a given system along with the reactions that operate on those species. A reaction network associates a set of n species $X = \{X_0, \ldots, X_{n-1}\}$ with a system of m coupled reversible reactions (stoichiometric equations) operating on this set of species

$$\sum_{i=0}^{n-1} s_{ri} X_i \underset{k_r^-}{\overset{k_r^+}{\rightleftharpoons}} \sum_{i=0}^{n-1} p_{ri} X_i \tag{2.6}$$

with $r = 0, \ldots, m-1$.

A reaction system is a system in the thermodynamic sense, characterized by a finite volume containing a finite quantity of particles of different species, whose interactions are defined by a reaction network. It is thus defined by three essential components:

- Its space.
- Its reaction network, with n species and m reactions.
- Its state x, which is the total number of particles of each of its n species inside the system.

The state x of a reaction system is defined as the vector of the species copy numbers

$$x = (x_0, \ldots, x_{n-1}) \tag{2.7}$$

The state can also be expressed in terms of concentrations, as

$$[X] = ([X_0], \ldots, [X_{n-1}]) = \frac{x}{V} = \left(\frac{x_0}{V}, \ldots, \frac{x_{n-1}}{V}\right) \tag{2.8}$$

where V is the total volume of the system.

2.4.2 With STReNGTHS

One of the main purposes of STReNGTHS is to provide a simple and intuitive interface to manipulate all the concepts previously mentioned (as well as those related to diffusion that we will introduce later). It does so using an object-oriented approach, where the concepts of species, reactions, reaction-diffusion networks and reaction-diffusion systems are represented by objects defined by different classes. In this subsection, we explain how to define a pure reaction system (i.e. with no diffusion) with STReNGTHS and present the API associated with the previously introduced concepts. Reaction(-diffusion) systems, reaction(-diffusion) networks, reactions and species are represented by objects of the classes `RDSystem`, `RDNetwork`, `Reaction` and `Species`, respectively.

A `Species` object represents a single chemical species, with a label attribute holding the name of the species. The API for this class goes as:

- `Species(label, D, density, chstt, ...)`: The constructor. Each argument sets the corresponding class attribute.
- `label`: The name (character string) of the species.
- `D`: Diffusion coefficient of the species, in dimensions of L^2/T.
- `density`: Default density of the species to be applied when creating the system state. It accepts various types of inputs, but densities are ultimately stored as `UnitValue` objects, with a dimension N/L^3.
- `chstt`: Attribute to control whether the species must be chemostatted (i.e. maintained at a fixed concentration) during the simulations. We will explain this feature in more detail later.

As an example,

```
A = Species("A", density=UnitValue("40 nM"))
```

creates a species "A" with a default initial density of 40 nM.

A `Reaction` object represents a single reversible reaction. It holds information about its stoichiometry, associating species labels to stoichiometric coefficients for both the substrate and the

product, as well as about its forward and backward reaction rate constants. An optional label attribute may be used to identify the reaction within a reaction network. The API for this class is

- `Reaction(stoechiometry, kf, kr, label, ...)` Constructor. The stoichiometry argument sets the stoichiometry and the direction of the reaction. It can be entered as a string with an intuitive syntax. For instance,

  ```
  "A + 2 C -> 3 D + E"
  ```

  describes a reaction having as substrates 1 molecule of the species A and 2 of C, and producing 3 molecules of D and one of E. The other arguments shown here set the corresponding class attributes.

- `label`: String attribute. It is the string identifier of the reaction. As opposed to the `Species` class, a label is optional.

- `kf` and `kr` correspond respectively the forward and reverse reaction rate constant of the reaction, k^+ and k^-. Their units depend on the stoichiometry.

As an example,

```
r = Reaction ("A + B -> C", "1 nM-1s-1", "2 s-1")
```

defines the reaction

$$A + B \underset{k^-}{\overset{k^+}{\rightleftharpoons}} C \tag{2.9}$$

with $k^+ = 1$ nM^{-1}s^{-1} and $k^- = 2$ s^{-1}

At this point, `Species` and `Reaction` objects are not aware of each other. `Species` and `Reaction` are associated through a reaction(-diffusion) network, represented by the class `RDNetwork`. This contains a list of `Species` objects with unique labels, and a list of `Reaction` objects operating on those species. The API for this class is

- `RDNetwork(species, reactions, environments, ...)`: The constructor. Arguments set the corresponding attributes.
- `species`: The list (tuple) of the species in the reaction network. Each species has a different label.
- `reactions`: The list (tuple) of the reactions in the reaction network.
- `environments`: The list (tuple) of reaction-diffusion environments for the network. We shall come back on this optional argument in the following.

As an example, the reaction network for the reaction of the previous example would be declared as:

```
species = [
    Species("A", density="10 molecule/µm3"),
    Species("B", density="5  molecule/µm3"),
    Species("C")]

reactions = [
    Reaction("A + B -> C", "1 nM-1s-1", "2 s-1")
    ]

network = RDNetwork(species, reactions)
```

Reaction networks are only one component of a reaction(-diffusion) system. A Reaction-diffusion system couples a reaction network to some space/reaction volume and is characterized by a state x. Those are represented by the `RDSystem` class. `RDSystem` objects hold a `network` attribute referring to a `RDNetwork` object, a `space` attribute describing the spatial configuration of the system, and a `state` attribute, holding the system state x. By default, the system space consists of a single well-mixed volume (a single cell), and there is no diffusion. System spaces and diffusion are an important topic that we will present later. The API for the `RDSystem` class is:

- `RDSystem(network, space, state, chemostats)`: The constructor, where each argument sets the corresponding attribute. Aside from `network`, all are optional and have default values.
- `network`: The reaction-network (`RDNetwork`) of the system.
- `space`: The space of the system. By default, it is a simple 1 μm^3 volume.
- `state`: The state of the system (`UnitArray` of dimension N). Its shape depends on the system space. By default, the quantities of species depend on the volume of the system space and the density attributed to the `Species` objects.
- `chemostats`: Chemostat assigned to each part of the system state. We will present this feature in more detail detail later.

As an example, a simple system with the reaction network from the previous example and a reaction volume of 1 μm^3 could be defined as:

```
species = [
    Species("A", density="10 molecule/µm3"),
    Species("B", density="5  molecule/µm3"),
    Species("C")]

reactions = [
    Reaction("A + B -> C", "1 nM-1s-1", "2 s-1")
    ]

space = RDGridSpace(cell_vol = "1 µm3")
```

```
network = RDNetwork(species, reactions)
system = RDSystem(system, space)
```

`RDGridSpace` is a class representing a reaction-diffusion space, which we will present in the next section that deals with diffusion. The only thing to understand here is that we use it to create a simple reaction volume of 1 μm^3. Using a simple 1 μm^3 reaction volume is the default behavior, so one could have left the space argument unspecified here.

We presented how to declare species, reactions and reaction(-diffusion) networks and systems using object construction in Python. However, initializing these objects by hand in such a way is not necessary, as they can be more conveniently created from a Python dictionary or a JSON file. Here are two examples using JSON and dictionary inputs to create a `RDSystem` identical to the one from the previous example. First, using a python dictionary and the `rdsystem_from_dict` function, which creates a `RDSystem` from a Python dictionary

```
system = rdsystem_from_dict({
    "network": {
        "species": [
            {"label": "A", "density": "10 molecule/µm3"},
            {"label": "B", "density": "5 molecule/µm3"},
            {"label": "C"}
            ],
        "reactions": [
            {"eq": "A + B -> C", "k+": "1 nM-1.s-1", "k-": "2 s-1"}
            ]
        },
    "space": {"cell_vol": "1 µm3"}
    })
```

The second approach consists of using a dictionary defined in a separate JSON file and loading it with the `load_rdsystem` function which creates a `RDSystem` from a JSON file, taking its path as argument:
system.json:

```
{
"network": {
    "species": [
        {"label": "A", "density": "10 molecule/µm3"},
        {"label": "B", "density": "5 molecule/µm3"},
        {"label": "C"}
        ],
    "reactions": [
        {"eq": "A + B -> C", "k+": "1 nM-1.s-1", "k-": "2 s-1"}
        ]
    },
```

```
"space": {"cell_vol": "1 µm3"}
}
```

```
system = load_rdsystem("system.json")
```

These two approaches may be more convenient than the object approach in most cases. The syntax used in a JSON file and for dictionaries is strictly the same. The same kind of approach holds for `RDNetwork` and other classes, using for example `rdnetwork_from_dict` and `load_rdnetwork`.

Let us get back to the `RDSystem` object we just created. Its state is the sequence of the quantities of each species inside the reaction volume, as defined in equation 2.7, and is arranged in the same order that the species list of the reaction network:

$$system.state = (\text{quantity of } A, \text{quantity of } B, \text{quantity of } C) \tag{2.10}$$

If a state is not explicitly defined at initialization, a default one is created using the species densities and the space volume. In this case, it would be $x = (10, 5, 0)$ molecules.

2.5 Modeling reaction-diffusion systems

2.5.1 The reaction-diffusion space

In the previous section, we considered the global evolution of the total quantity of the different species inside a volume, where molecules are considered as evenly distributed. To account for diffusion, without resorting to a particle-based approach, it is possible to break down the system space as a discrete network comprising a finite number ω of sub-volumes, or cells. Species are considered evenly distributed within each cell, but not across cells. Neighbor cells can exchange molecules by diffusion. The total volume V of the system, is distributed among its ω cells

$$V = \sum_{\alpha=0}^{\omega-1} v_\alpha \tag{2.11}$$

where v_α is the volume of the $\alpha-$th cell. STReNGTHS supports two different approaches for decomposing the reaction space into ω cells: 3D grids of cells, **grid spaces**, and graphs of cells, **graph spaces** (figure 2.1).

2.5.1.1 Grid spaces

Grid spaces are the main and the default way to describe the reaction-diffusion space (figure 2.1, left). A grid space is a 3D grid of $w \times h \times d$ cells of uniform volume v. The total number of cells is thus $\omega = whd$ and the total volume $V = whdv = \omega v$. The number of neighbors of a given cell depends on its position in the grid. A cell has 6 neighbors, minus one for each face of the grid it belongs to. A cell is primarily referred to by its index $\alpha \in [0, \omega - 1]$. However, in the case of grid spaces, a cell can also be identified by its 3D coordinates vector (x, y, z), in cells, with

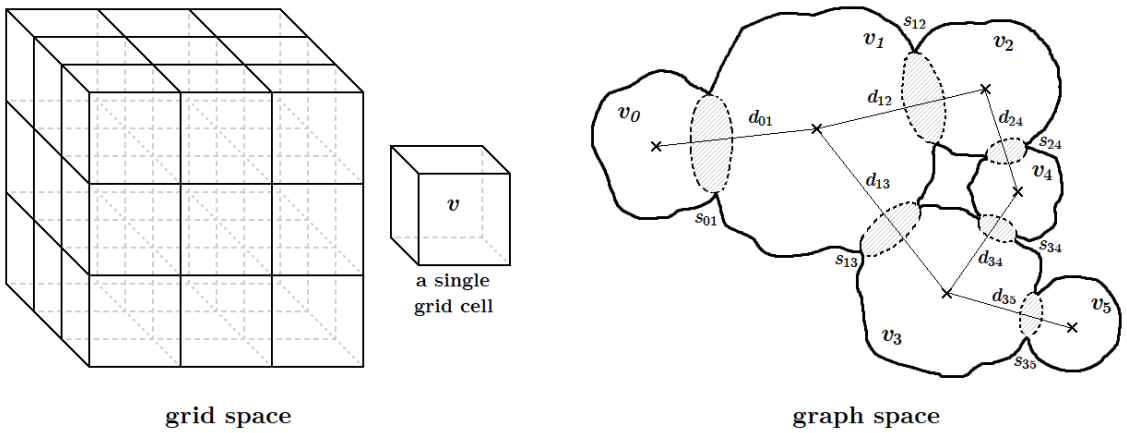


Figure 2.1: **Illustration of the two discrete space types supported by STREnGTHS.** Left: Schematic representation of a grid space. Each cell has the same volume v and the same distance from its neighbors. Right: Schematic representation of a graph space, with 6 nodes/cells. Each cell α has its volume v_α. Each pair of connected cells (α, β) is characterized by a contact surface $\sigma_{\alpha\beta}$ and a center-to-center distance $d_{\alpha\beta}$.

$x \in [0, w-1], y \in [0, h-1]$ and $z \in [0, d-1]$. The correspondence between the index α of a cell and its coordinates is defined as

$$\alpha = zwh + yw + x \tag{2.12}$$

2.5.1.2 Graph spaces

Graphs are another way to represent the diffusion space (figure 2.1, right). They are more complex than regular cell grids, but allow for more diverse and realistic configurations, as cells may have different volumes and any number of neighbors. Such undirected graphs have ultimately two components: cells, which are the nodes, and edges, which are connections between two nodes. As with grid spaces, each of the ω cells of the graph is referred to by an index $\alpha \in [0, \omega - 1]$. Edges are referred to by the indices of the two cells they link (α, β), with $\alpha, \beta \in [0, \omega - 1]$. Each node α has volume v_α, and each edge (α, β) is characterized by the contact surface between its two nodes, $\sigma_{\alpha\beta}$ and by the distance between their centers $d_{\alpha\beta}$. The total reaction volume V is given by the sum of the individual volumes of the cells:

$$V = \sum_{\alpha=0}^{\omega-1} v_\alpha \tag{2.13}$$

Graphs are the most general type of space, as all grid spaces can be described as graph spaces while the opposite is not true. As a consequence, graph spaces can be used for coarse-graining grid spaces, as we shall explain later.

2.5.2 Redefining the system state

In the previous section, we defined the state of a reaction system composed of a single reaction volume (one cell). Now that we have introduced the concept of subdivision of the reaction space, it is necessary to redefine the organization of the system state accordingly. To do so, let us discriminate

species by cell. Now, we refer to the species X_i in the cell α as $X_{i\alpha}$. The system state corresponding to the resulting set of species can be represented as a $n \times \omega$ matrix

$$x = \begin{pmatrix} x_{0,0} & \cdots & x_{0,\omega-1} \\ \vdots & \ddots & \vdots \\ x_{n-1,0} & \cdots & x_{n-1,\omega-1} \end{pmatrix} \tag{2.14}$$

where $x_{i\alpha}$ is the quantity of the species X_i in the cell α. We must also redefine the concentration state $[X]$ accordingly. To express the state in terms on concentrations, we have to divide each of its elements by the corresponding volume

$$[X] = \begin{pmatrix} [X_{0,0}] & \cdots & [X_{0,\omega-1}] \\ \vdots & \ddots & \vdots \\ [X_{n-1,0}] & \cdots & [X_{n-1,\omega-1}] \end{pmatrix} \tag{2.15}$$

with $[X_{i\alpha}]$, the concentration of the species X_i in the cell α, defined as

$$[X_{i\alpha}] = \frac{x_{i\alpha}}{v_\alpha} \tag{2.16}$$

2.5.3 Modeling diffusion as a reaction

A standard approach consists of modeling the diffusion of a given species X_i from a given cell α to a neighbor cell β as a first-order reaction with a rate constant $k_{i\alpha\beta}$ [4]

$$X_{i\alpha} \underset{k_{i\beta\alpha}}{\overset{k_{i\alpha\beta}}{\rightleftharpoons}} X_{i\beta} \tag{2.17}$$

where, using the notation defined previously, $X_{i\alpha}$ and $X_{i\beta}$ are the species X_i in the cells α and β respectively [4]. The diffusive reaction rate $k_{i\alpha\beta}$ is determined from the species X_i's inter-cell diffusion coefficient $D_{i\alpha\beta}$, between the cells α and β. For a grid of regular cubic cells, the inter-cell diffusion coefficient is defined as

$$D_{i\alpha\beta} = \frac{\ell_\alpha + \ell_\beta}{\frac{\ell_\alpha}{D_{i\alpha}} + \frac{\ell_\beta}{D_{i\beta}}} \tag{2.18}$$

where ℓ_α and ℓ_β are the sides of the cubic cells α and β, respectively, and $D_{i\alpha}$ and $D_{i\beta}$ the respective diffusion coefficients of species X_i [4]. For both space types, the cell width is defined more generally as the cube root of the cell volume. In the case of a grid space, $k_{i\alpha\beta}$ is then defined as

$$k_{i\alpha\beta} = D_{i\alpha\beta}\frac{1}{\ell^2} \tag{2.19}$$

where ℓ is the width of a individual cell [4]. In the case of graph spaces, however, cells are connected through contact surfaces and distances that can change from edge to edge. In order to deal with the general case, we extended equation 2.19 to account for the contribution of $\sigma_{\alpha\beta}$ and $d_{\alpha\beta}$, for any edge (α, β). Elementary dimensional reasoning suggests to consider

$$k_{i\alpha\beta} = D_{i\alpha\beta}\frac{\sigma_{\alpha\beta}}{v_\alpha d_{\alpha\beta}} \tag{2.20}$$

which guarantees that the diffusion rate is proportional to the cell's contact surface, inversely proportional to the cell volume and center distance, and simplifies to equation 2.19 when the graph corresponds to a regular grid space.

2.5.4 Reaction-diffusion network and system

We now turn to extending the concept of reaction networks and systems (presented section 2.4.1.3) to include diffusion and space subdivision. We consider a system with n species and m chemical reactions (i.e. except those that account for diffusive jumps), in a volume V comprising ω cells. As explained in section 2.5.2, we now make a distinction between the subpopulation of each of the n species in the ω cells of the system, writing the full set of species

$$X = \{X_{i\alpha} | i \in [0, n-1], \alpha \in [0, \omega - 1]\} \tag{2.21}$$

The same distinction is applied to the system state, defined in equation 2.14 as a $n \times \omega$ matrix

$$x = \begin{pmatrix} x_{0,0} & \dots & x_{0,\omega-1} \\ \vdots & \ddots & \vdots \\ x_{n-1,0} & \dots & x_{n-1,\omega-1} \end{pmatrix} \tag{2.22}$$

The full reaction-diffusion network for the system is defined as the set of all the diffusive and non-diffusive reactions that can happen in the system

$$\begin{aligned} &\left\{ \sum_{i=0}^{n-1} s_{ri} X_{i\alpha} \underset{k^-_{r\alpha}}{\overset{k^+_{r\alpha}}{\rightleftharpoons}} \sum_{i=0}^{n-1} p_{ri} X_{i\alpha} \,\middle|\, \alpha \in [0, \omega - 1], r \in [0, m-1] \right\} \\ \cup \\ &\left\{ X_{i\alpha} \underset{k_{i\beta\alpha}}{\overset{k_{i\alpha\beta}}{\rightleftharpoons}} X_{i\beta} \,\middle|\, i \in [0, n-1], \text{ for all edges } (\alpha, \beta) \right\} \end{aligned} \tag{2.23}$$

where $k^+_{r\alpha}$ and $k^-_{r\alpha}$ are the forward and reverse rate constants of the reaction r in the cell α, as we consider that the rate constants of a given reaction may change depending on the cell.

2.5.5 With Strengths

In the previous section, we presented 5 of the main classes of STReNGTHS: `Species`, `Reactions`, `RDNetwork`, `RDSystem` and briefly `RDGridSpace`. After reading this section, one should now have an idea of what the `RDGridSpace` represents. We will now extend the presentation of the classes presented in the last section with the new diffusion-related concepts introduced in this section.

As we have seen previously, the `RDSystem` class has a space attribute representing the reaction-diffusion space, which accepts a `RDGridSpace` object. Here, we will see that the system space also accepts another class: `RDGraphSpace`. Grid and graph spaces are respectively implemented in the `RDGridSpace` and `RDGraphSpace` classes. On the one hand, `RDGridSpace` objects have `w`, `h` and `d` attributes, corresponding to the dimensions of the grid in cells. The `cell_vol` attribute defines the volume of an individual cell of the grid. The API for this class is

- `RDGridSpace(w, h, d, cell_env, cell_vol)`: The constructor, where each argument sets the corresponding attribute.
- `w`, `h` and `d` are the width, height and depth, respectively, of the grid in cells, set to 1 by default.
- `cell_env`: An attribute that we will introduce later.
- `cell_vol`: The volume of an individual cell, 1 μm^3 by default.

On the other hand, `RDGraphSpace` objects possess a `nodes` and a `edge` attributes, giving access to the list (`tuple`) of nodes/cells and edges of the graph, respectively. Each node is itself an object with a `volume` attribute defining its volume. Each edge is also an object, with attributes `i` and `j`, corresponding to the indices α and β of the two linked cells, as well as a `surface` and `distance` attributes, corresponding respectively to $\sigma_{\alpha\beta}$ and $d_{\alpha\beta}$. The APIs for those classes are:

- `RDGraphSpace(nodes, edges)`: The constructor, where each argument sets the corresponding attribute.
- `nodes`: The list of the graph's nodes (cells). Nodes are objects of the `RDGraphSpaceNode` class, whose API is
  - `RDGraphSpaceNode(volume, environment)`: The constructor, where each argument sets the corresponding attribute.
  - `volume`: Volume of the node/cell.
  - `environement`: Attribute homologous to `RDGridSpace`'s `cell_env` attribute. We will give more details about it later and put it aside for now.
- `edges`: The list of the graph's edges (cells). Edges are objects of the `RDGraphSpaceEdge` class, whose API is
  - `RDGraphSpaceEdge(i, j, surface, distance)`: The constructor, where each argument sets the corresponding attribute.
  - `i` and `j` are the indices α and β of the nodes/cells connected by the edge.
  - `surface`: Surface of contact $\sigma_{\alpha\beta}$ between the two linked nodes.
  - `distance`: Distance $d_{\alpha\beta}$ between the center of the two linked nodes.

To illustrate the use of these classes, let us get back to the example system from the last section. This time we want the space to be a 2×2 cubic arrangement of cells with individual volumes of 1 μm^3. Also, let us add diffusion coefficients for A, B and C of 1, 2 and 3 μm^2/s, respectively. The JSON/dictionary that defines such a system using a grid space, reads

```
{
"network": {
    "species": [
        {"label": "A", "D": "1 µm2/s", "density": "10 molecule/µm3"},
```

```
        {"label": "B", "D": "2 µm2/s", "density": "5 molecule/µm3"},
        {"label": "C", "D": "3 µm2/s"}
        ],
    "reactions": [
        {"eq": "A + B -> C", "k+": "1 nM-1.s-1", "k-": "2 s-1"}
        ]
    },
"space": {
    "w": 2,
    "h": 2,
    "cell_vol": "1 µm3"
    }
}
```

The previous example should be self-explanatory. This time, however, it is more clear that `"cell_vol"` sets the volume of an individual cell and not that of the whole system. Now, using a graph space, the same system could be built as

```
{
"network": {
    "species": [
        {"label": "A", "D": "1 µm2/s", "density": "10 molecule/µm3"},
        {"label": "B", "D": "2 µm2/s", "density": "5 molecule/µm3"},
        {"label": "C", "D": "3 µm2/s"}
        ],
    "reactions": [
        {"eq": "A + B -> C", "k+": "1 nM-1.s-1", "k-": "2 s-1"}
        ]
    },
"space": {
    "type": "graph",
    "nodes": [
        {"volume": "1 µm3"},{"volume": "1 µm3"},
        {"volume": "1 µm3"},{"volume": "1 µm3"}
        ]
    "edges": [
        {"i": 0, "j": 1, "distance": "1 µm", "surface": "1 µm2"},
        {"i": 1, "j": 3, "distance": "1 µm", "surface": "1 µm2"},
        {"i": 2, "j": 3, "distance": "1 µm", "surface": "1 µm2"},
        {"i": 0, "j": 2, "distance": "1 µm", "surface": "1 µm2"}
        ]
    }
}
```

By default, the `"space"` key is expected to describe a grid space. The `"type"` key must be set

to `"graph"` so that the `"space"` dictionary is interpreted as describing a graph space.

The corresponding systems are loaded as described in the last section, using `rdsystem_from_dict` or `load_rdsystem`. Since the system space counts multiple cells, the state is not the same as before. This time, there is a distinction between the species in each cell and the system is a matrix as in equation 2.14. The quantity of a given species X_i in a cell α is referred by a global index I, such that

$$\begin{aligned} system.state[I] &= x_{i\alpha} \\ I &= i\omega + \alpha \end{aligned} \tag{2.24}$$

The density specified at the species level is just one of the options available to create a default system state. It is also possible to define the system state as a whole. This can be done at initialization, either by specifying the `state` of the `RDSystem` constructor or by setting the `"state"` key in the system dictionary/JSON. If the state is defined explicitly, it takes priority over the densities specified for the species. If we take the case of the last example, we can specify the state as:

```
{
"network": {
    "species": [
        {"label": "A", "D": "1 µm2/s"},
        {"label": "B", "D": "2 µm2/s"},
        {"label": "C", "D": "3 µm2/s"}
        ],
    "reactions": [
        {"eq": "A + B -> C", "k+": "1 nM-1.s-1", "k-": "2 s-1"}
        ]
    },
"space": {
    "w": 2,
    "h": 2,
    "cell_vol": "1 µm3"
    },
"state": {"value": [
 #----#
  5,5,#
  5,5,#  A
 #----#
  3,3,#
  3,3,#  B
 #----#
  2,2,#
  2,2,#  C
 #----#
], "units": "molecules"}
}
```

Here, we have set the the quantity of A, B and C to 5, 3 and 2 molecules in all cells, respectively. The system state can also be modified after initialization/loading, using the `get_state` and `set_state` methods of `RDSystem`:

```
a0 = system.get_state("A", (1,1,0)) # = 5 molecule
a0 = system.get_state("A", 3)       # Equivalent.

v = UnitValue("10 molecule")
system.set_state("A", 3, v)  # Setting the state for A at pos. 3 to v.
system.get_state("A", 3)     # = 10 molecule
```

2.6 Simulating reaction-diffusion system trajectories

One of the main purposes of STReNGTHS is to deal with the simulation of trajectories of reaction-diffusion systems. These trajectories define the system state x as a function of time t, $x(t)$. There are two kinds of approaches supported by STReNGTHS: Deterministic ones, relying on the integration of the system's rate equations, and stochastic ones, relying on kinetic Monte Carlo algorithms [22] [21]. Simulated trajectories are sequences of states sampled at successive times during a simulation of the time-evolution of the system. We refer to a discrete trajectory, composed of λ samples, as a the vector $\overrightarrow{x}$

$$\overrightarrow{x} = (x(t_0), \ldots, x(t_{\lambda-1})) \tag{2.25}$$

2.6.1 Deterministic approaches

The rate equations of a system are the set of Ordinary Differential Equations (ODE) that describe how the system state evolves in time in a well-stirred volume. The deterministic approach corresponds to the mean-field approximation of the full stochastic equation. In the two next sections, we will see how to define the rate equations, first for the simple case of a pure reaction system, and then for a full reaction-diffusion system. We note that, treating diffusion as a special kind of reaction (involving transfer of molecules from a cell to one of its neighbors), we effectively describe a full RD system as an enlarged set of rate equations.

2.6.1.1 Rate equations for a reaction system

Before addressing the case of reaction-diffusion systems, let us start with the simple case of a pure reaction system, i.e. without diffusion. We consider a simple system of volume V with a reaction network as defined in section 2.4.1.3, with n species and m reactions. For a given reversible reaction r, the forward and reverse reaction currents, Φ_r^+ and Φ_r^-, which correspond to the number of occurrences of the forward and reverse reactions, respectively, per unit of time in the system, are

defined from the system state as

$$\begin{aligned}\Phi_r^+ &= Vk_r^+ \prod_{i=0}^{n-1} \left(\frac{x_i}{V}\right)^{s_{ri}} \\ \Phi_r^- &= Vk_r^- \prod_{i=0}^{n-1} \left(\frac{x_i}{V}\right)^{p_{ri}}\end{aligned} \tag{2.26}$$

The net reaction current is the difference between the forward and reverse currents, defined as

$$\Phi_r = \Phi_r^+ - \Phi_r^- \tag{2.27}$$

The rate equations can then be written as

$$\begin{pmatrix} \frac{dx_0}{dt} \\ \vdots \\ \frac{dx_{n-1}}{dt} \end{pmatrix} = \mathcal{S} \begin{pmatrix} \Phi_0 \\ \vdots \\ \Phi_{r-1} \end{pmatrix} \tag{2.28}$$

where $\mathcal{S}$ is the $n \times m$ stoichiometric matrix for the reaction network, defined as

$$\mathcal{S} = \begin{pmatrix} p_{0,0} - s_{0,0} & \cdots & p_{r-1,0} - s_{r-1,0} \\ \vdots & \ddots & \vdots \\ p_{0,n-1} - s_{0,n-1} & \cdots & p_{r-1,n-1} - s_{r-1,n-1} \end{pmatrix} \tag{2.29}$$

2.6.1.2 Rate equations for a reaction-diffusion system

In the case of a reaction-diffusion system comprising n species, m reactions and ω cells, as defined in equation 2.23, one needs to consider species and reactions by cell and to account for diffusive reactions. The currents for a chemical reaction r in a cell α, are defined as in equations 2.26 and 2.27, except that the volume must be that of the cell, and we use the specific rate constants for the cell α

$$\begin{aligned}\Phi_{r\alpha}^+ &= v_\alpha k_{r\alpha}^+ \prod_{i=0}^{n-1} \left(\frac{x_{i\alpha}}{v_\alpha}\right)^{s_{ri}} \\ \Phi_{r\alpha}^- &= v_\alpha k_{r\alpha}^- \prod_{i=0}^{n-1} \left(\frac{x_{i\alpha}}{v_\alpha}\right)^{p_{ri}} \\ \Phi_{r\alpha} &= \Phi_{r\alpha}^+ - \Phi_{r\alpha}^-\end{aligned} \tag{2.30}$$

For the diffusion of a given species X_i, we can define the modified discrete Laplacian $\mathcal{L}_i$ as

$$\mathcal{L}_{i\alpha\beta} = \begin{cases} -\sum_{\gamma \in \mathcal{N}(\alpha)} k_{i\alpha\gamma} & \text{if } \beta = \alpha \\ k_{i\beta\alpha} & \text{if } \beta \in \mathcal{N}(\alpha) \\ 0 & \text{otherwise} \end{cases} \tag{2.31}$$

In components, the rate equation reads

$$\frac{dx_{i\alpha}}{dt} = \sum_{r=0}^{m-1} \Phi_{r\alpha}(p_{ri} - s_{ri}) - \sum_{\beta=0}^{\omega-1} \mathcal{L}_{i\alpha\beta} x_{i\alpha\beta} \tag{2.32}$$

Alternatively, they can be written with the stoichiometric matrix defined in equation 2.29, as

$$\frac{dx_{i\alpha}}{dt} = (\mathcal{S}\Phi)_{i\alpha} - \sum_{\beta=0}^{\omega-1} \mathcal{L}_{i\alpha\beta} x_{i\alpha\beta} \tag{2.33}$$

2.6.2 Time evolution of the system

STReNGTHS leaves the user the option to choose between a deterministic evolution and a stochastic approach. In the first case, the rate equations are solved numerically with a simple Euler algorithm. For systems with only one cell, STReNGTHS can be interfaced with the SciPy Python package [59], which provides more robust solvers. However, simulating the trajectory of a system by integrating its rate equations is not always appropriate, as it does not account for the discrete quantity of molecules/particles in the system. This may not be a problem with a large number of molecules/particles, but this is usually not the case for biological systems where some proteins may be present in low quantities within a cell [40]. Finite quantities induce fluctuations, and some systems may exhibit multistable steady states that cannot be accounted for at the mean-field level [5]. In such cases one should solve the full master equation (ME) of a system through a kinetic Monte Carlo approach.

The Gillespie algorithm, introduced in 1977 by Daniel T. Gillespie [22], provides a simple and elegant way to obtain an exact numerical solution to the ME. For a given state x at a time t, a random event and a random time step are drawn to specify the next system state at time $t+\tau$. By construction, this algorithm operates on system states consisting of discrete quantities of particles for each species. In this section, we will recapitulate in detail how the Gillespie algorithm works, along with its key concepts such as propensities, in the context of reaction-diffusion systems.

Let us consider the case of a system with n species, m reactions and ω cells. We call an event any specific diffusive or non-diffusive reaction in a given direction. The propensity of a given event is a measure of its probability to happen [21]. When the event is a reversible reaction r in a cell α, the forward and reverse propensities, $a^+_{r\alpha}$ and $a^-_{r\alpha}$ are defined as

$$\begin{aligned} a^+_{r\alpha} &= v_\alpha k^+_{r\alpha} \prod_{i=0}^{n-1} \prod_{l=0}^{s_{ri}} \frac{x_{i\alpha} - l}{v_\alpha} \\ a^-_{r\alpha} &= v_\alpha k^-_{r\alpha} \prod_{i=0}^{n-1} \prod_{l=0}^{p_{ri}} \frac{x_{i\alpha} - l}{v_\alpha} \end{aligned} \tag{2.34}$$

For the diffusive reaction of a species X_i, from a cell α to a neighbor cell β, the forward and reverse propensities are defined as [4]

$$\begin{aligned} a_{i\alpha\beta} &= k_{i\alpha\beta} x_{i\alpha} \\ a_{i\beta\alpha} &= k_{i\beta\alpha} x_{i\beta} \end{aligned} \tag{2.35}$$

The propensities are functions of time, $a_\mu = a_\mu(t)$. The system evolves by drawing two independent random variables, i.e. the time τ that elapses until the next event will occur, and the probability that the event that occurs between time $t + \tau$ and time $t + \tau + dt$ (where dt is the time step) will be the event μ. It can be shown that

$$P(\tau) = a_0 e^{-a_0 \tau} \tag{2.36}$$

where $a_0 = \sum_\nu a_\nu$, and

$$P(\mu) = \frac{a_\mu}{a_0} \tag{2.37}$$

[22]. Once the time τ has been drawn from the PDF 2.36 and the next event from the PDF 2.37, the time is incremented by τ and the system state updated, $x \to x + \Delta x$. The state variation Δx is given by

$$\Delta x_{i,\alpha} = \begin{cases} 1 & \text{if } \mu \text{ refers to the diffusion of the species } X_i \text{ into the cell } \alpha \\ -1 & \text{if } \mu \text{ refers to the diffusion of the species } X_i \text{ out of the cell } \alpha \\ p_{ri} - s_{ri} & \text{if } \mu \text{ refers to the foward reaction } r \text{ in the cell } \alpha \\ s_{ri} - p_{ri} & \text{if } \mu \text{ refers to the reverse reaction } r \text{ in the cell } \alpha \\ 0 & \text{otherwise} \end{cases} \tag{2.38}$$

The Gillespie algorithm iterates the two steps described above until the specified end condition is met. It is possible in certain systems that at some point no event is possible. In such cases, $a_0 = 0$ and the simulation is halted.

2.6.2.1 The τ-leap algorithm

The τ-leap algorithm is an approximation of the Gillespie numerical procedure [22] introduced by David T. Gillespie himself in 2001 [21]. Instead of executing one event at a time, the τ-leap approach draws, based on the propensities, random numbers of events that should occur over some fixed time step (for each event).

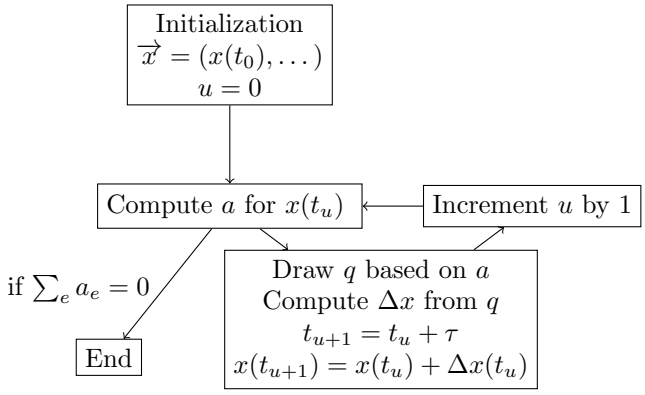
We consider again the case of a reaction-diffusion system with n species, m reactions and ω cells with a state x at some time t, as well as a fixed time step τ. The propensities a are computed for all events at time t, as described in equations 2.34 and 2.35. Then, we draw the number q_e of any possible event e, occuring over the time-step τ, from a Poisson distribution [21]

$$q_e \sim \text{Poisson}(a_e \tau) \tag{2.39}$$

The state variation matrix, Δx, follows then as

$$\Delta x_{i\alpha} = \sum_{r=0}^{m-1} (q^+_{r\alpha} - q^-_{r\alpha})(p_{ri} - s_{ri}) + \sum_{\beta \in \mathcal{N}(\alpha)} (q_{i\alpha\beta} - q_{i\beta\alpha}) \tag{2.40}$$

where $q^+_{r\alpha}$ and $q^-_{r\alpha}$ are the number of occurrences of the reaction r in the cell α respectively in the forward and reverse directions, and $q_{i\alpha\beta}$ and $q_{i\beta\alpha}$ are the number of diffusive reactions, for the species X_i, from the cell α to the cell β and vice-versa respectively. The workflow for the τ-leap [21] algorithm can be summarized as



which is analogous to the Gillespie algorithm. A key advantage of the τ-leap algorithm is that it requires fewer iterations. Also, in the context of dicrete reaction-diffusion systems, it offers better perspectives of computational parallelization, as events happen simultaneously in all cells, which could lead to faster simulations.

2.6.3 With STReNGTHS

STReNGTHS uses a common abstract interface for reaction-diffusion simulation algorithms, through object called engines. This allows the user to set up simulations independently of the method used to compute the time evolution of the system, whether deterministic or stochastic.

STReNGTHS has 3 built-in engines, implementing a simple Euler integration of the rate equations, The Gillespie algorithm [22] and the tau-leap algorithm [21], all handling diffusion through additional jump reactions as previously explained [4]. However, more engines should be developed in the future, hopefully, as the engine abstraction is also designed for STReNGTHS to be extensible with new methods and/or implementations.

Engines can be classes derived from the `RDEngineBase` base class, which defines the interface. The common API for simulation engines is

- There is no standard interface for the constructor of the engine. However, the `RDEngineBase` class takes two arguments: `option` and `description` setting the corresponding properties.
- Attributes:
  - `option`: String whose purpose is implementation-dependent. It can be used to input additional parameters to the simulation method implemented by the engine.
  - `description`: String describing the simulation engine, which will be stored in simulation outputs for context.
- `setup(script)`: Initialize the engine with a simulation `script` (A `RDScript` object).
- Iteration functions:
  - `iterate()`: Executes one iteration of the simulation algorithm
  - `iterate_n(n)`: Executes `n` iterations of the simulation algorithm

  - `run(dt)`: Executes iterations of the simulation algorithm for a duration `dt`.

- `sample()`: Samples the current system in the trajectory. This is useful if one wants to deal with sampling manually, However, this is not required, as some sampling policy can be defined in the simulation script.
- `is_complete()`: Tells if the simulation is complete (returns `True` if this is the case).
- `get_output()`: Returns the simulated system trajectory (A `RDTrajectory` object, whose class will be presented later).

The setup method excepts a `RDScript` object. The `RDScript` class wraps simulation parameters. Its API reads:

- `RDScript(system, t_sample, ...)`: The constructor, where, as usual, each argument sets the corresponding attribute.
- `system`: The reaction-diffusion system whose trajectory shall be simulated.
- `t_sample`: The time points at which the system state should be sampled during the simulation to build a trajectory.

There is more to `RDScript` than just the `system` and `t_sample` attributes, though we will not go into such details here. Continuing with the system from the last example, let us set up a trajectory, using STReNGTHS's built-in deterministic engine. Here, let us suppose that the system is described in a JSON file named "system.json" in the same directory as the Python script:

```
system = load_rdsystem("system.json")
t_sample = UnitArray(np.linspace(0, 100, 1000), "s")
script = RDScript(system, t_sample)
engine = euler_engine()
engine.setup(script)
while not engine.is_complete():
    engine.iterate()
trajectory = engine.get_output()
```

While engines are powerful objects, using them manually as above is not necessary in most cases. The `simulate` function is the preferred option to launch simulations, as it wraps the manipulation of the engine. `simulate` takes the same arguments as `RDScript`, plus a few more, including in particular the simulation engine to be used for the simulation. The same example using `simulate` reads

```
system = load_rdsystem("system.json")
t_sample = UnitArray(np.linspace(0, 100, 1000), "s")
engine = euler_engine()
trajectory = simulate(system, t_sample, engine=engine)
```

The simulate function returns a `RDTrajectory` object, the one returned by the `get_output` method of its engine at the end of the simulation. As the name suggests, an object of the `RDTrajectory` class represents the trajectory of a reaction-diffusion system. The API for this class is:

- `RDTrajectory(data, t_sample, system, script, ...)` : The constructor, where each argument sets the corresponding attribute, except for `t_sample` which sets the attribute `t`, but this should change in future versions of the package. The `script` argument is optional. `RDTrajectory` objects are typically generated by simulation engines, so that users should not have to call this constructor.
- Attributes/Properties:
  - `data` : The sequence of sampled system states of the trajectory, $\overrightarrow{x}$.
  - `t` : The sequence of time points associated with the sampled system states, $\overrightarrow{t}$.
  - `system` : The system whose trajectory is represented by the instance.
  - `script` : The simulation script associated with the trajectory. It is optional and only serves to give context to the trajectory.
- Methods:
  - `get_trajectory(species, position, merge)` : Returns the trajectory of a given `species` at a given `position`. The `merge` argument is optional and set to `False` by default. if set to `True`, the global trajectory of the `species` in the whole system is returned.
  - `get_state(species, sample)` : Returns the system state for a given `species` for a given `sample` of the trajectory. If `species` is set to `None`, the whole state is returned for the given `sample`.

As for `RDNetwork` or `RDSystem` object, `RDTrajectory` objects can be saved/loaded using the `load_rdtrajectory` and `save_rdtrajectory` functions. Once we have obtained a system trajectory out of a simulation, the next step is to analyze it. The first and most basic thing one can do is to plot the global trajectory for the different species as well as the distribution of the species at different times. This can be done using the `Matplotlib` package [30]. The `plot` submodule of STReNGTHS, which wraps `Matplotlib` calls, can also be used to facilitate some plots. Here it is an example showcasing a system with a single species A diffusing with a coefficient of 1 μm^2/s in a 10×10 grid of 1 μm^3 cells, while only half of the system is initially filled with molecules (1000 molecules per cell):

```
import numpy as np
from strengths import *
from strengths.plot import *

# defining the system
```

```
system = rdsystem_from_dict({
    "network": {
        "species": [
            {"label": "A", "D": "1 µm2/s"}
            ],
        "reactions": []
        },
    "space": {
        "cell_vol": "1 µm3",
        "w": 10,
        "h": 10
        }
    })

# filling the lower half of the system, while leaving the rest empty

for y in range(5):
    for x in range(10):
            system.set_state("A", (x,y,0), "1000 molecule")

# simulation

trajectory = simulate(
    system,
    UnitArray(np.linspace(0, 100, 1000), "s"),
    engine=tauleap_engine()
    )

# plot of the global trajectory of A

plot_trajectory(trajectory, "A")

# plot of the global trajectory of in the first cell

plot_trajectory(trajectory, "A", position=0)

# plot of the distribution of A in the system for each sample

for t in ["0 s", "5 s", "100 s"]:
    plot_sample_state_2D(
        trajectory,
        "A",
        trajectory.get_sample_index(t),
        xmin=min(trajectory.data.value),
        xmax=max(trajectory.data.value)
```

```
        )
```

The simulation is performed with the `simulate` function using the `tauleap_engine()` implementing the τ-leap algorithm [21]. Plots are realized using the `plot_trajectory` and `plot_sample_state_2D` from the `plot` submodule of STReNGTHS, which internally relies on the `Matplotlib` package. The resulting plots can be seen in figure 2.2.

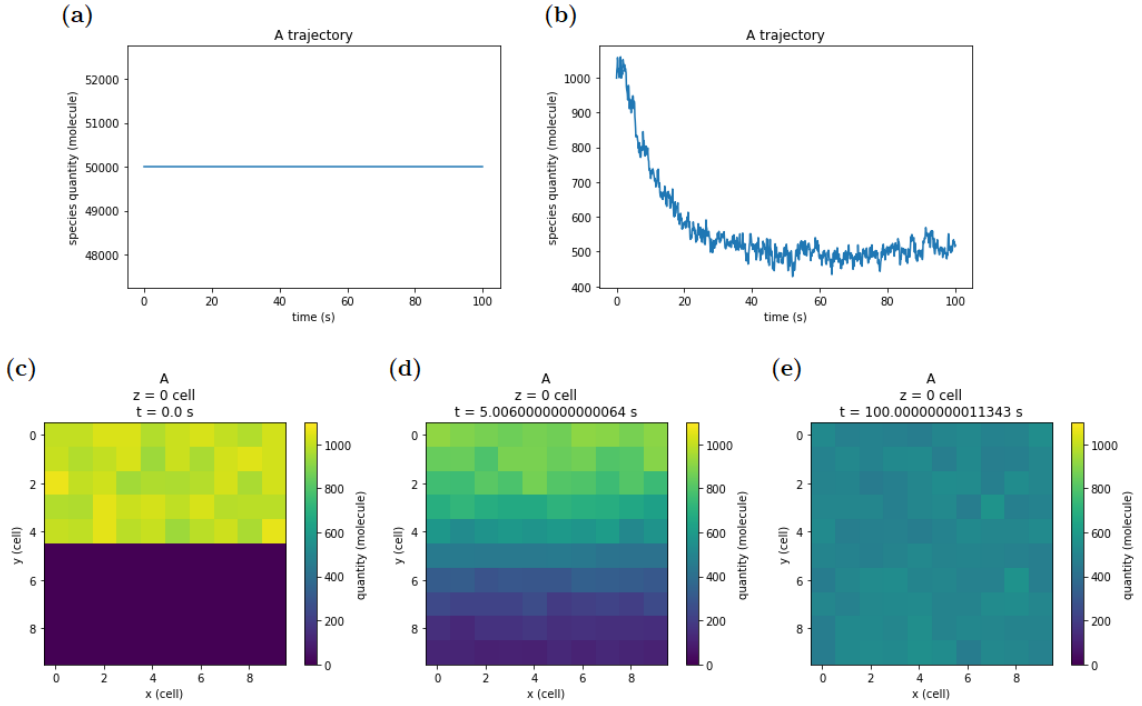


Figure 2.2: **Trajectory of a single-species diffusion system simulated with the τ-leap algorithm [21].** The system counts only one species A, with a diffusion coefficient of 1 μm^2/s. The system space is a 10×10 grid and each cell has a volume of 1 μm^3. In the initial conditions, half of the system is filled with A at 1000 molecules per cell. The rest of the system is empty. (a) Trajectory of the quantity of A molecules in the whole system. (b) Trajectory of the quantity of A in the top left most (or (0,0)) cell of the system. (c, d, e) Distribution of the quantity of A in the system at various times.

2.7 Important features for modeling biochemical systems

In the previous sections, we have seen how, with STReNGTHS, one can define reaction-diffusion systems and simulate their trajectories. With what we have seen so far, one can already model a wide variety of interesting systems. However, biochemical reaction-diffusion systems usually involve species diffusion in different compartments, typically including phospholipidic membranes. More-

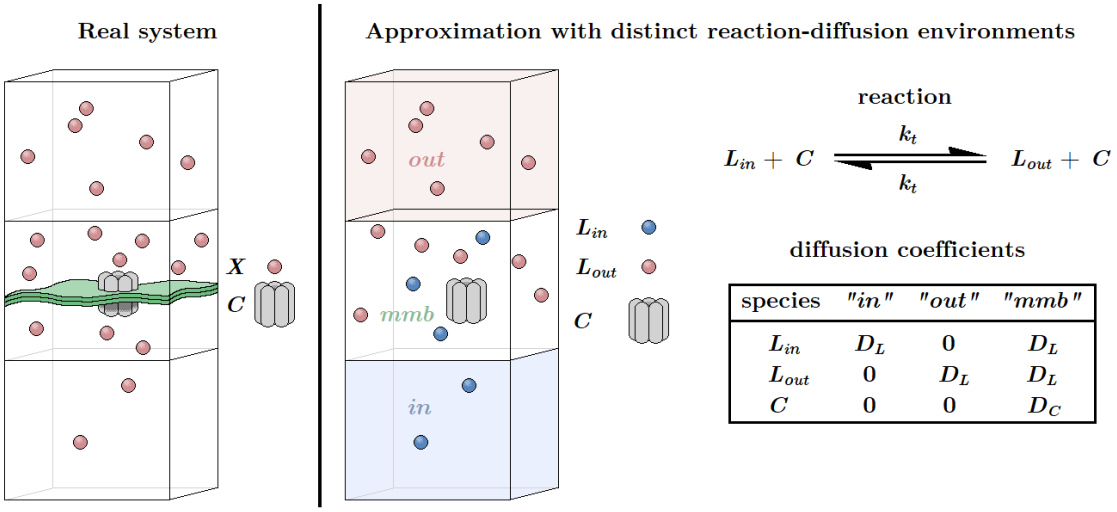


Figure 2.3: **Illustration of the concept of reaction-diffusion environment, through the example of a system with two compartments separated by a membrane.** The left scheme shows the ideal system we want to model A first species L diffuses on each side of the membrane, and a second one, C, inside the membrane, acts as a transporter for the first species. The right scheme shows how we can model this system using reaction-diffusion environments, by distinguishing L based on which side of the membrane it is. The reaction and diffusion coefficients are those of the corresponding reaction-diffusion network.

over, living systems work far from equilibrium, usually due to some species having their densities maintained at fixed levels in some part (or all) of the systems. As a consequence, STReNGTHS implements two features, which allow to account for such phenomena: **reaction-diffusion environments**, allowing for the definition of different cell types with different properties, and **chemostats**, which allow to impose fixed quantities for some species in some or all parts of the system space.

2.7.1 Reaction-diffusion environments

A wide variety of different systems can be modeled with homogeneous system spaces, however, biological systems generally feature different compartments with different chemical properties. To account for such a spatial inhomogeneity, STReNGTHS relies on the concept of **reaction-diffusion environments**. Reaction-diffusion environments can be seen as cell types, for which we can define specific reactive and diffusive properties. For a given reaction-diffusion network, a set of different environments is defined, and then each cell of the system is assigned one of those environments. Then some species properties, such as the density or the chemostat flag, can be defined environment-wise, which can help in setting up some specific initial state or landscape. The species diffusion coefficient can also be defined environment-wise, which can for example serve to restrict diffusion of certain species to certain areas of the system. Reactions can also be confined to certain environments.

An example is illustrated in figure 2.3: Say we want to create a three-cells long system featuring two compartments separated by some membrane in the middle. The system should contain two molecular species. A first species L, which should diffuse freely on both sides of the membrane, but

not through it, and a second one, a passive transporter species C inside the membrane allowing the first species to cross the membrane. To model such a system with reaction-diffusion environments, 3 environments would be required: one per compartment, and one for the space close to the membrane. Membranes often separate an "outside" relative to an "inside", so we label our environments *in*, *out* and *mmb* ("membrane"). The transporter species must only exist inside the membrane compartment. The transported species would exist in two versions, L_{in} and L_{out}, (treated as two different species), the inside one and the outside one, and both would diffuse only in their respective compartment as well as in the membrane one. The L_{in} species would be transformed by C to produce L_{out} and vice-versa.

For a given system, the list of the different reaction-diffusion environments must be specified in the reaction-diffusion network (`RDNetwork`). This is where the `environment` attribute of `RDNetwork` comes into play:

- `environments` list of labels of the different reaction-diffusion environments for the reaction-diffusion network.

In the case of the previous example, considering L_0 the initial density of L on each side of the membrane and C_0 the initial density of C at the membrane, with D_L and D_C the respective diffusion coefficients of L on both side of the membrane and C in the membrane, and k_t the passive transport rates of L across the membrane through C, we could define the network as

```
species = [
    Species("Lin",  D={"mmb, in" : D_L }, density={"mmb, in" : L_0 }),
    Species("Lout", D={"mmb, out": D_L }, density={"mmb, out": L_0 }),
    Species("C",    D={"mmb"     : D_C }, density={"mmb"     : C_0 })
    ]

reactions = [
    Reaction("Lin + C -> Lout + C", kf= k_t , kr= k_t )
    ]

environments = ["out", "mmb", "in"]

network = RDNetwork(species, reactions, environments)
```

or with a dictionary/JSON file, as

```
{
"species": [
    {"label": "Lin",  "D": {"mmb, in" : D_L }, "density": {"mmb, in" : L_0 }},
    {"label": "Lout", "D": {"mmb, out": D_L }, "density": {"mmb, out": L_0 }},
    {"label": "C",    "D": {"mmb"     : D_C }, "density": {"mmb"     : C_0 }}
    ],
"reactions": [
    {"eq": "Lin + C -> Lout + C", "k+": k_t , "k-": k_t }
```

```
    ],
"environments": ["out", "mmb", "in"]
}
```

where D_L, D_C, L_0 and C_0 should be replaced by their actual value. Here, we can see a new way to define certain properties. Species densities and diffusion coefficients may be defined environment-wise using dictionaries. In the case of L_{in} for example,

```
"density": {"mmb, in" : L_0}
```

sets the density of L_{in} to L_0 in the *mmb* and *in* environments, and 0 in the others (the *out* environment).

We still have to define which cell is associated with which environment. This is to be done in the reaction-diffusion space. Let us consider the 3 meshes layout of the example. With a grid space, the layout could be specified as:

```
"space": {
    "w": 3,
    "cell_vol" : "1 µm3",
    "cell_env" : [0, 1, 2]
    }
```

where `"cell_env"` maps each cell to a given environment. For efficiency reasons, in this case, environments are not referred to by their labels, but by their indices in the environment list of the corresponding `RDNetwork`. As a consequence, in this case, the first cell is assigned *out*, the second *mmb* and the third *in*. With a graph space, it would be:

```
"space": {
    "type": "graph",
    "nodes" : [
        {"volume": "1 µm3", "env" : 0},
        {"volume": "1 µm3", "env" : 1},
        {"volume": "1 µm3", "env" : 2}
        ],
    "edges" : [
        {"i": 0, "j": 1, "surface": "1 µm2", "distance": "1 µm"},
        {"i": 1, "j": 2, "surface": "1 µm2", "distance": "1 µm"}
        ]
    }
```

It works the same, except the environments are set with the `"env"` key of each node.

2.7.2 Chemostats

In biological reaction-diffusion systems, some species may have their concentrations maintained at a fixed concentration, whether due to metabolic activity or to the system being open and connected to large, well-mixed reservoirs. STReNGTHS implements a chemostat feature, which works like the species densities: Chemostats may be set at the species level, either globally or environment-wise, i.e.

```
Species("A", chstt=True)                # A will be chemostated.
Species("B", chstt=1)                   # Equivalent.
Species("C", chstt={"some_env": True}) # A will be chemostated only in the
                                        # "some_env" environment.
```

or in the dictionary/JSON form:

```
{"label": "A", "chstt": true},
{"label": "B", "chstt": 1},
{"label": "C", "chstt": {"some_env": true}}
```

When a `RDSystem` is created, the `chstt` attribute of species is used to create a chemostats map, the `chemostats` attribute of `RDSystem` which have the same shape as the system `state`, and define whether a given species at a given position is chemostatted or not. It is thus possible to define chemostats directly at the `RDSystem` level, as we would do for the system state, without specifying them inside the `Species` objects. Here is an example, where we only want to chemostat the center of some 3×3 cells system:

```
{
"network": {
    "species": [
        {"label": "A", "density": 100}
        ],
    "reaction":[
        {"eq": "A ->", "k+":1}
        ]
    }
"space": {
    "w": 3,
    "h": 3
    },
"chemostats": [
    0,0,0,
    0,1,0,
    0,0,0,
    ]
}
```

As for the system state, the chemostat map can also be manipulated after the `RDSystem` object is created, using the `set_chemostat` and `get_chemostat` method:

```
system.set_chemostat("A", (1,1), True)
c = system.get_chemostat("A", (1,1)) # c is True
```

2.8 Optimizations

2.8.1 Coarse-graining

Let us consider a system featuring a membrane compartment where most of the events happen. In this compartment, diffusion is slow, and a high spatial resolution is required. Outside of this compartment, however, only fast diffusion occurs, and a lower spatial resolution would be acceptable. The problem, with a grid space, is that since the membrane requires a high spatial resolution, this high resolution must also be applied to the rest of the system. As a consequence, an exponentially high number of cells is required and the simulation performance is proportionally impacted. One way to solve this problem is to coarse-grain the system grid space as a graph, where the spatial resolution would be adapted depending on the system kinetics. As we have already discussed, graph spaces are more complicated to define than grid spaces. Creating by hand a coarse-grained version of a grid would thus be tedious. To face this problem, STReNGTHS implements a system of grid coarse-graining relying on a *coarse-graining map*, which describes how the grid should be coarse-grained. The coarse-graining map associates to each cell of the initial grid a cell index of the coarse-grained graph space, as illustrated in figure 2.4. The grid cells are appropriately merged and proper edges are generated in the process. It is important to consider, however, that the coarse-graining must be performed among cells with the same environments, as merging cells with different environments would be ambiguous.

The simplest way to coarse-grain a grid space for a simulation is by providing the `cgmap` argument of the `simulate` function with a coarse-graining map. Here is an example corresponding to the case of figure 2.4(b) where a grid of 10×10 cells with 3 environments *out*(blue), *mmb*(green) and *in*(grey) is coarse-grained to a system comprising 13 cells of variable volumes:

```
system = rdsystem_from_dict({
  "network": {
    "species": [{"label": "A"}],
    "reactions": [];
    "environments": ["out", "mmb", "in"]
    },
  "space": {
    "w": 4,
    "h": 4,
    "env": [
      0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,
      0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,
```

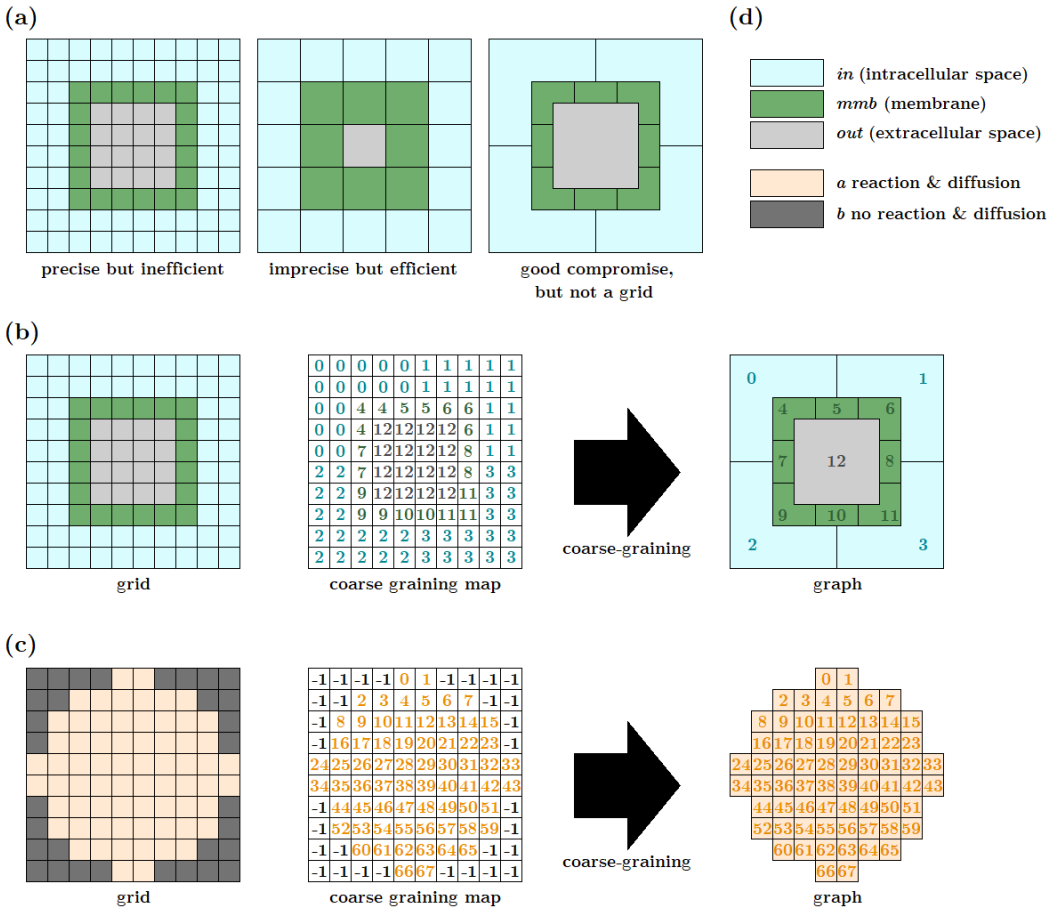


Figure 2.4: **Coarse-graining grid spaces with STReNGTHS.** (a) Illustration of how, with regular mesh grids, a high spatial resolution required in some parts must also be applied everywhere else, impacting the performance of simulations. (b) Illustration of the concept of coarse-graining map, mapping together the cell indices of the (input) grid space to those of the (output) graph space. (c) Another example of coarse-graining that consists in the removal of unused cells. (d) Cell environments legend.

```
      0 ,0 ,1 ,1 ,1 ,1 ,1 ,1 ,0 ,0 ,
      0 ,0 ,1 ,2 ,2 ,2 ,2 ,1 ,0 ,0 ,
      0 ,0 ,1 ,2 ,2 ,2 ,2 ,1 ,0 ,0 ,
      0 ,0 ,1 ,2 ,2 ,2 ,2 ,1 ,0 ,0 ,
      0 ,0 ,1 ,2 ,2 ,2 ,2 ,1 ,0 ,0 ,
      0 ,0 ,1 ,1 ,1 ,1 ,1 ,1 ,0 ,0 ,
      0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,
      0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
     ]
   }
 })

cgmap = [
  0 ,0 ,0 ,0 ,0 ,1 ,1 ,1 ,1 ,1 ,
  0 ,0 ,0 ,0 ,0 ,1 ,1 ,1 ,1 ,1 ,
  0 ,0 ,4 ,4 ,5 ,5 ,6 ,6 ,1 ,1 ,
  0 ,0 ,4 ,12,12,12,12,6 ,1 ,1 ,
  0 ,0 ,7 ,12,12,12,12,8 ,1 ,1 ,
  2 ,2 ,7 ,12,12,12,12,8 ,3 ,3 ,
  2 ,2 ,9 ,12,12,12,12,11,3 ,3 ,
  2 ,2 ,9 ,9 ,10,10,11,11,3 ,3 ,
  2 ,2 ,2 ,2 ,2 ,3 ,3 ,3 ,3 ,3 ,
  2 ,2 ,2 ,2 ,2 ,3 ,3 ,3 ,3 ,3
  ]

trajectory = simulate(system, [1,2,3], cgmap=cgmap)
```

By using this approach, the coarse-graining step is "hidden", as the coarse-graining of the system and the un-coarse-graining of the trajectory are done inside the `simulate` function call.

Another use of coarse-graining is to optimize simulations so as to prune inert cells, as illustrated in figure 2.4(c). Cells to be removed must simply be given an index of -1 in the coarse-graining map.

2.9 Illustrative examples

This section illustrates more complex examples that still make use of the features described in this chapter [18].

For the first example, let us consider a simple model of signal transduction, where some extracellular chemical signal is sensed by a cell, which triggers the production of a second messenger, as well as the scavenging of the signal species. The network is designed as follows: The extracellular ligand L can bind to a plasma membrane receptor R to form a complex C. This species catalyzes the

conversion of the inactive second messenger X to its active form, Y. The complex C is directly converted back into R, which accounts for the internalization of the complex, the degradation of the ligand and full recycling of the receptor. No degradation of the receptors (either through the proteasomes or lysosomes) is assumed for simplicity, although it would be straightforward to add additional reactions to implement such reaction channels (see Fig. 2.5 a,c). The model features 3 reaction-diffusion environments, *ext*, *cyt* and *mmb*, accounting, respectively, for the extracellular space, the cytoplasm and the interface between these two compartments containing the plasma membrane. We use two different system spaces. The first one is a 26×26 mesh grid consisting of 1 μm^3 cubic cells, while the second one is a coarse-grained version of the first that contains only 85 nodes/cells (for 676 cells in the grid) (Fig. 2.5 b). The trajectory of the system state is simulated, both for the fully detailed grid and for its coarse-grained version, using the τ-leap algorithm [21] and the Euler method, for a total duration of 1 hour using a time step of 1 ms. The global trajectory of Y as well as its distribution at $t = 0, 100, 1500$ s are plotted in Fig. 2.5 (f, g).

For the second example, we consider a pattern-forming reaction-diffusion network based on the one used in the documentation of the package [17], similar to the Gray-Scott model and related reaction-diffusion schemes [38] [45], which features two competitive auto-catalytic species A and B mutually converting into each other (Fig. 2.6 (a)). We first simulate the evolution of the system in 1 dimension using the τ-leap algorithm [21]. The corresponding spatio-temporal evolution of A concentration over time is reported in Fig. 2.6 (c) as a 2D heat map. It can be observed how the system, starting from a homogeneous state, progressively builds up spatial reaction-diffusion patterns. We also simulate a square 100×100 mesh system and plot the final distribution of A, so that the shape of the pattern can be appreciated directly (Fig. 2.6 (d)). Next, we apply this model to two systems with higher complexity, where the synthesis rates of A and B vary depending on the region (Fig. 2.6 (b), (e), (g)). The first one (Fig. 2.6 (e)) represents pattern formation at the surface of a sphere, while the second one (Fig. 2.6 (g)) illustrates a similar phenomenon of pattern formation in a domain that takes the shape of an animal. Panels (f) and (h) in Fig. 2.6 demonstrate the evolution of both systems (levels of the A species) in time, highlighting the progressive evolution of the spatial patterns that emerge as a result of the subtle combination of the underlying auto-catalytic process with the inhomogeneous reaction landscape. Figure 2.6 (i) additionally illustrates how, due to the stochastic nature of the process, different patterns may arise from the same homogeneous initial state.

the simulation scripts for those two examples can be found in the repository of the software: `https://github.com/ThibaultFillion/strengths/tree/main` which is also archived on Zenodo [14] - as a version of record associated with the publication - at: `https://zenodo.org/records/11261599`

2.10 Source code, distribution and documentation

STReNGTHS is written in Python and C++ and relies on the standard libraries of each language as well as on the NumPy [25] and Matplotlib [30] Python packages. The API, that we partially presented in the previous sections, is fully in Python. As briefly explained in the introduction, we chose Python as it comes with a wide range of powerful tools for scientific computing as well as data analysis and visualization, such as NumPy, SciPy or Matplotlib. The three built-in engines rely on a compiled shared library written in C++, which produces faster code for this computationally intensive part. The package, including compiled binaries, is distributed with the Python Package Index (`https://pypi.org/project/strengths/`) and can be installed using pip, the package

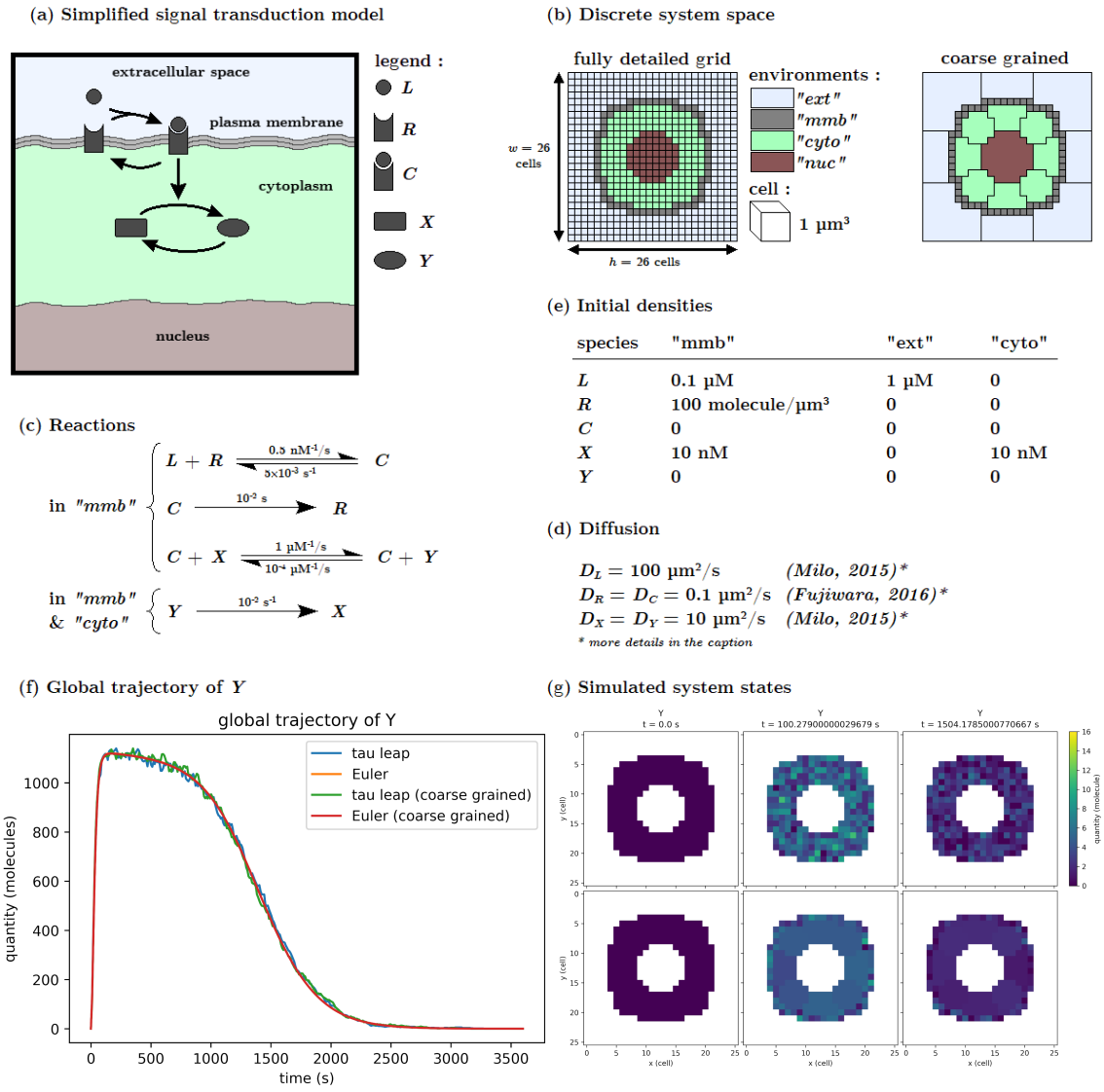


Figure 2.5: **Example of simulation with STReNGTHS implementing a simple model of signal transduction by a single cell.** (a) Schematic representation of the system. (b) Layout of the two different system spaces used, the 26×26 mesh grid (left) and its coarse-grained graph version (right). (c) Set of coupled stoichiometric equations that compose the reaction network. (d) Diffusion coefficients of individual species in the different environments. (e) Initial densities of each species in the different environments. (f) Time course of the transduced signal: Global trajectory of Y obtained from the simulation using the τ-leap algorithm [21] and the Euler method using both system spaces (see b). (g) Distribution of the Y species at different times from the τ-leap simulations. The rates used were: $L+R \to C$: $k_1 = 0.5$ nM^{-1}s^{-1}. $C \to L+R$: $k_{-1} = 0.5 \times 10^{-3}$ s^{-1}. $C \to R$: $k_2 = 10^{-2}$ s^{-1}. $C+X \to C+Y$: $k_3 = 1$ μM^{-1}s^{-1}. $C+Y \to C+X$: $k_{-3} = 10^{-4}$ μM^{-1}s^{-1}. $Y \to X$, $k_4 = 10^{-2}$ s^{-1}. The diffusion coefficients for the different species were: $D_L = 100$ μm^{2}s^{-1} (order of magnitude for the diffusion coefficient of a protein around 30 kDa in water ([40], p. 213)), $D_R = D_C = 0.1$ μm^{2}s^{-1} (transmembrane protein in a compartmented plasma membrane [20]), $D_X = D_Y = 10$ μm^{2}s^{-1} (protein around 30 kDa in the cytosol ([40], p. 213)).
Figure reproduced from reference [18].

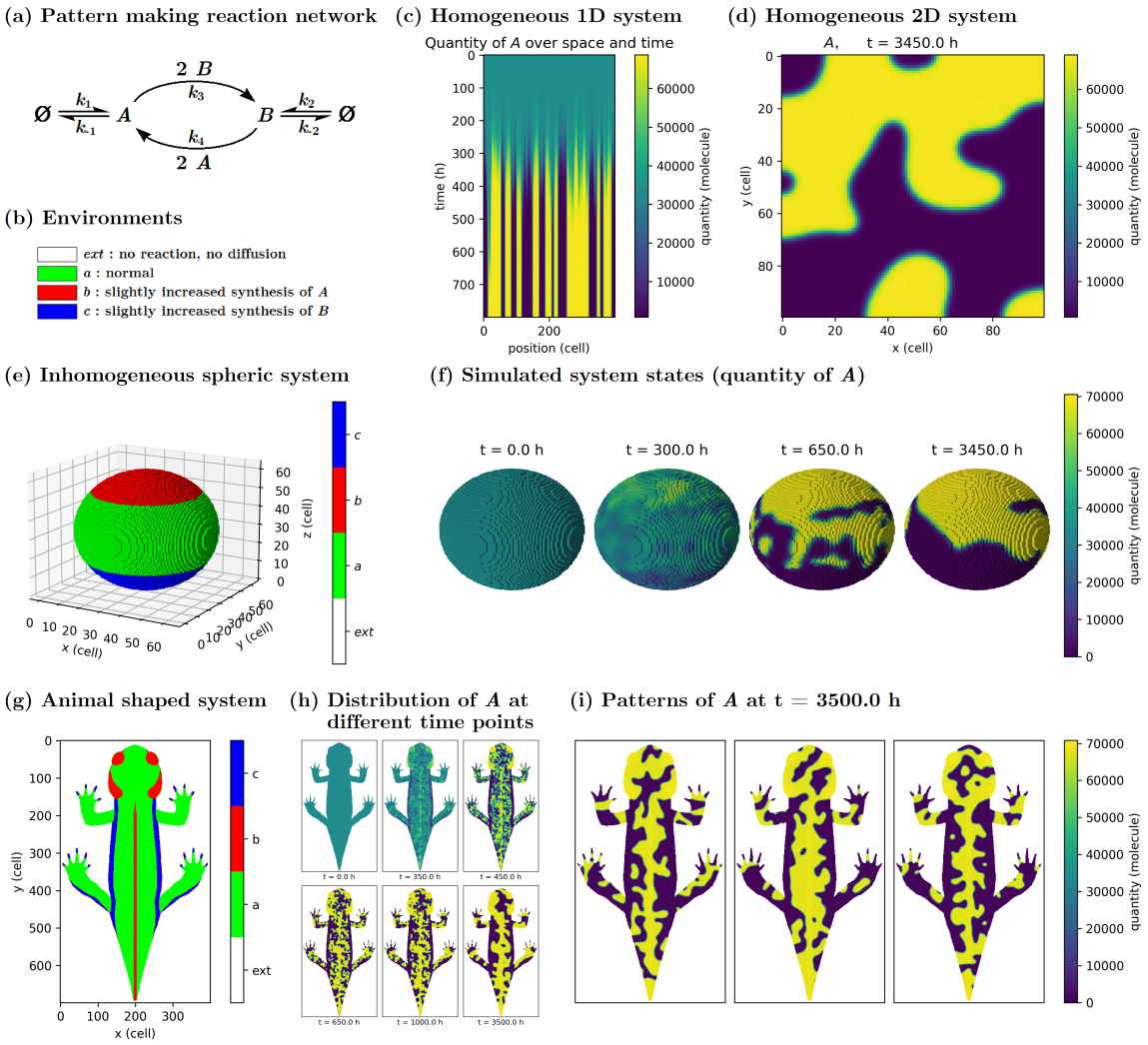


Figure 2.6: **Example of simulations of different pattern-forming reaction-diffusion systems at increasing level of environmental complexity.** All simulations are performed with the τ-leap algorithm [21]. (a) Description of the chemical reactions and associated rates. This reaction-diffusion network is similar to the Gray-Scott model [38] and the one studied by Ruijgrok and Ruijgrok [45]. Other examples using a similar reaction-diffusion network can be found in the documentation of the package [17]. (c) Evolution of a 1D system in time and space. (d) Pseudo-stationary state of a 2D system. (e) reaction-diffusion environments for the two inhomogeneous systems. (b) Layout of a 3D system where the patterns are forming at the surface of a sphere. (f) States of the 3D system described at different time points along a stochastic trajectory (concentration of the A species). (g) Layout of a 2D system mimicking the shape of an animal. (h) States of the system (distribution of the A species) at different time points from one simulation. (i) Patterns formed at t = 3500.0 h resulting from 3 different stochastic simulations. The rates used were: $\emptyset \rightarrow A$: $k_1 = 10^{-4}$ molecules$\times\mu$m$^{-3}\times$h^{-1} in a and c and 1.05×10^{-4} molecules$\times\mu$m$^{-3}\times$h^{-1} in b. $A \rightarrow \emptyset$: $k_{-1} = 10^{-3}$ h^{-1}. $\emptyset \rightarrow B$: $k_2 = 10^{-4}$ molecules$\times\mu$m$^{-3}\times$h^{-1} in a and b and 1.05×10^{-4} molecules$\times\mu$m$^{-3}\times$h^{-1} in c. $B \rightarrow \emptyset$: $k_{-2} = 0.001$ h^{-1}. $A + 2B \rightarrow 3B$: $k_3 = 1$ molecules$^{-2} \times \mu$m$^6\times$h^{-1}. $B + 2A \rightarrow 3A$: $k_4 = 1$ molecules$^{-2} \times \mu$m$^6\times$h^{-1}. The diffusion coefficient for the different species were: $D_A = D_B = 80$ μm^{2}h^{-1}. Reaction and diffusion rates constants are all 0 in compartment ext. Figure reproduced from reference [18].

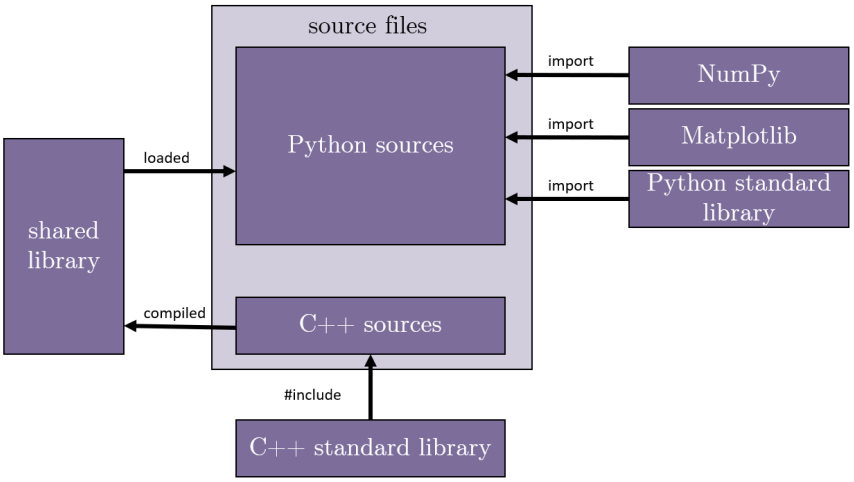


Figure 2.7: **Structure of STReNGTHS's source code and its interaction with external dependencies.**

manager of Python, with:

```
pip install STReNGTHS
```

STReNGTHS also has an online HTML documentation (`https://strengths.readthedocs.io/en/latest/`), generated using Sphinx (`https://www.sphinx-doc.org/en/master/`), and distributed using the ReadTheDocs (`https://about.readthedocs.com/?ref=readthedocs.org`). The API is documented directly within the package source files using docstrings and the corresponding web documentation is generated automatically with Sphinx.

The project is open-source, licensed under the terms of the MIT license, which allows anyone to freely fork the project and/or submit contributions provided the conditions of the aforementioned license are met. The source code of the package and its documentation, the full list of the contributors, as well as most other related information and documents can be found in the project's GitHub repository (`https://github.com/ThibaultFillion/strengths/tree/main`). The repository also contains various unit test scripts, separated from the rest of the source code, allowing to monitor the different functionalities of the software during development. Unit testing is handled with the pytest package [33]. We published STReNGTHS in The Journal of Open Source Software [18]. A version of record of the repository associated with the article is published on Zenodo [14] (`https://zenodo.org/records/11261599`).

2.11 Conclusion

In this chapter, we presented the STReNGTHS package, as well as the related modeling and simulation concepts and techniques. STReNGTHS provides an interface to model reaction networks as sets of coupled stoichiometric equations. Systems are discrete compositions of homogeneous reaction volumes, or cells, across which chemical species can diffuse. Diffusion is modeled as a jump reaction [4]. To account for the heterogeneity of biochemical networks, which often involve multiple

compartments separated by membranes, it is possible to define different types of cells that we term reaction-diffusion environments. Species can also be chemostatted to account for non-equilibrium driving. Simulations are handled by classes called simulation engines, relying on an abstract interface common to most simulation algorithms. Three simulation engines are built in STReNGTHS, implementing respectively a simple Euler scheme, integrating the rate equations of the system, and the stochastic Gillespie [22] and tau-leap [21] algorithms.

The package is written in Python, which is a popular programming language in sciences and features a rich panel of useful packages, and in C++, which is used internally to bring more performance to the simulation routines. It is an open-source project with a repository on GitHub (`https://github.com/ThibaultFillion/strengths/tree/main`).

STReNGTHS is still a fairly recent project and should hopefully grow and implement new interesting features in the future. Future development perspectives for the package would especially involve adding a massively parallel GPU implementation of the different algorithms as well as dynamic time step adaptation strategies.

Chapter 3

Modeling GTPase Signal Transduction Reaction Networks

3.1 Introduction

Living cells are complex organisms confronted with a changing environment. They are capable of sensing those changes and quickly adapt by producing an appropriate response. A striking example would be chemotaxis, where cells adapt the direction of their migration according to a chemical signal. The sensing of the chemical signal and the subsequent production of a response usually relies on highly complex cascades of chemical reactions, sometimes referred to as signaling pathways. The biochemical reaction networks involved are capable of propagating a change in the quantity of some upstream signal downstream of the signaling cascade. This process is referred to as signal transduction. Signal transduction systems are defined by their ability to produce, following some stimulus/stimuli called the **input signal**, a response called **output signal**. The output signal may differ in shape, dynamics, intensity and nature from the input signal. When studying the behavior of a signal transduction system, it is fundamental to identify its input and output signals.

GTPases of the RAS superfamily are Guanosine triphosphate (GTP) hydrolases [42] acting as protein regulators and signal transducers in various biochemical processes [7]. Those include G_α protein [7] subunits which are involved in G protein-coupled receptor (GPCR) signaling, and small GTPases, involved for example in the regulation of actin polymerization [53]. GTPases catalyze the hydrolysis of GTP into guanosine diphosphate (GDP) and inorganic phosphate (P_i) [7]. They thus cycle between 3 states depending on their interaction with their substrates: nucleotide-free, GDP-bound and GTP-bound. As GTPases have a high affinity for their substrates [65], they are mostly found in the GTP and GDP-bound states, while the nucleotide-free state is rare and transient. GTPases are active in the GTP-bound state and inactive in the GDP-bound state [49], as by switching to a GTP-bound state they undergo a conformation change which increases their affinity for downstream effectors [44].

In a typical GTPase signal transduction system, GTPases are inactive in the basal/rest state, due to the interplay between the slow dissociation of nucleotides and their hydrolytic activity [7], and

activated - following an upstream signal - by Guanine nucleotide Exchange Factors (GEFs), which increase the dissociation of their substrate nucleotides [58], allowing the GDP to be replaced by a GTP, in large excess in the cytoplasm. The active GTPases then associate with downstream effectors [44], allowing the input signal to be propagated in the signaling cascade. As the upstream signal ceases, the system returns to its basal state.

The transition between the three GTPase states results from the interplay between the association/dissociation of GTP and GDP at the active site and the catalyzed hydrolysis of the bound GTP. Those three reactions form a cycle, which we will refer to as the GTPase cycle. This cycle is shared by many other enzymes, such as G-actin [23] and Hsp70 [41], which makes it a fundamental biochemical pattern. In the cytoplasmic conditions where those reactions happen, GDP, GTP and P_i levels are maintained by the metabolism. As a consequence, this GTPase cycle works far from thermodynamic equilibrium. As we will discuss in this chapter, this is very important as the regulation of GTPase activity relies on this far-from-equilibrium context.

Aside from its substrates/products guanosine nucleotides, GTPases interact with different types of proteins, including GEFs and effectors that we previously mentioned. Those partners can be grouped into 4 categories

- The effectors. Those are the partners regulated by the active GTPase, responsible for the downstream propagation of the signal [58] [44].
- The GEFs. Those may be the most important category. They are regulating partners responsible for the activation of GTPases, accelerating the nucleotide exchange [58], and especially their dissociation, in particular for GDP [65]. GPCRs acquire this function upon association with an agonist ligand [11].
- The GTPase-Activating Proteins (GAPs). Those are inactivators of GTPases, acting as regulating partners increasing the catalytic activity of GTPases [58].
- The Guanine nucleotide Dissociation Inhibitors (GDIs). As their name implies, they prevent the dissociation of the GTPases nucleotidic substrates [43] and are also known for their sequestering role [58] [43].

Those four categories of partners are the typical components of most GTPase signal-transduction systems.

In this chapter, we will more specifically consider the case of GPCR signal transduction systems. GPCRs constitute a well-known superfamily of signaling transmembrane receptors involved in the signal transduction for a great number of biological processes [32], such as chemotaxis [37] or olfaction and vision [1] (page 832). GPCRs, as their name implies, are coupled with trimeric proteins called G proteins. G proteins work as two dissociable subunits: The G_α subunit, which is a GTPase of the Ras superfamily [7], and the $G_{\beta\gamma}$ dimer (G_β and G_γ subunits) acting as a partner for G_α specifically, binding to its inactive form [52]. The signal transduction scheme for GPCRs is the same as the one given earlier for GTPase systems. At rest, G_α subunits are inactive and bound to $G_{\beta\gamma}$ subunits. As GPCRs are activated by their agonist ligands - the input signal -, they acquire their GEF function and activate G_α subunits [11]. These dissociate from $G_{\beta\gamma}$ subunits, and both can then associate with their respective downstream effectors [11]. This is where one of the main specificities of the GPCR signal transduction system stands: it generates two parallel output signals.

In this chapter, we are going to investigate the properties of GTPase reaction networks through mathematical modeling and numerical simulations, in particular in the context of GPCR signaling. Starting with the simple case of the elementary GTPase cycle, we will gradually consider more complex reaction networks and finally focus on the specific case of the GPCR signal-transduction system.

The first section is dedicated to the presentation of the different reaction networks studied in this chapter. We begin with the fundamental GTPase cycle, then introduce networks with one or more partners, and eventually present the reaction network for a whole GPCR signal transduction network. In this section, we will specifically define the notations that we use for the reaction networks.

As evoked earlier, biochemical systems usually operate far from equilibrium. As a consequence, it is necessary to properly account for the sources of non-equilibrium driving to properly understand how they work. This driving usually results from the presence of chemostats. This means that when those are turned off, the system should relax to a state of equilibrium, where detailed balance holds. For detailed balance to hold in a reaction network, its equilibrium constants are required to be subject to specific constraints [8]. Thus, in the second section, we address the formulation of those constraints for the different reaction networks.

Signal transduction networks are interesting in their own right, but to fully understand their properties in a biological context, it is necessary to study them with relevant values of their parameters. In the third section, we will define the parameter values - associated more specifically with the reaction kinetic rate constants and species quantities - for the different systems, from the scientific literature.

The fourth section is shorter and will deal with the quantification of the signal in the different systems, defining the notation and especially the signal transduction function $\mathcal{S}$.

In the fifth section, using the thermodynamically consistent models we defined with their sets of biologically relevant parameters, we will investigate their signaling properties through numeric simulations, using both deterministic and stochastic methods. We will in particular identify and characterize the different kinetic parameter regimes of the different systems.

3.2 Reaction networks

3.2.1 Naming conventions

The reaction networks featured in this chapter are entirely built out of reversible association-dissociation reactions. Thus we consider two types of species: elementary ones, which cannot be divided into subspecies, and complexes, which are compositions of elementary species. In general, species names are written as the sequence of their subspecies names, separated by a hyphen ("-") and without spaces. As an example:

- The name A-B would refer to the complex formed by two particles of species A and B,
- A-B-C stands for the species formed by three particles of species A, B and C, no matter in which order they associate.

However, there are exceptions to those naming conventions for complexes that already have well-established names. As an example,

- The complex of GDP and P_i is GTP, not GDP-P_i.
- A receptor R and its ligand L form the complex C, not R-L.

Reactions are labeled after the species involved. Each species is assigned a one-lowercase letter label derived from its full name. For instance, species named R, C, E, X, Y, GTP, GDP and P_i would be labeled r, c, e, x, y, t, d and p. From those species letters, we generally define the reaction label from the perspective of the GTPase complex: the reaction label begins with the letter of the species being added to the complex and is followed by the letters of the species that are already part of it, except for the GTPase itself. As an example, the reaction

$$X + G \rightleftharpoons X\text{-}G \tag{3.1}$$

would simply be labeled x, while the reaction

$$X + G\text{-}GTP \rightleftharpoons X\text{-}G\text{-}GTP \tag{3.2}$$

would have the label xt as X is being added to the GTPase complex and GTP is already part of it. Only the position of the first species matters. For example, the reaction

$$X + Y\text{-}Z\text{-}G\text{-}GTP \rightleftharpoons X\text{-}Y\text{-}Z\text{-}G\text{-}GTP \tag{3.3}$$

could be labeled $xyzt$, $xtyz$ or $xzyt$ for instance. However, for the sake of clarity, a consistent notation is preferred.

3.2.2 The elementary GTPase network

The reaction network for a GTPase is well-known and contains 3 reactions ([65], [16]): the reversible association of the nucleotide-free GTPase with GDP and GTP and the hydrolysis of the bound GTP nucleotide, releasing an inorganic phosphate. The network contains 6 species: the nucleotide-free GTPase G, its GTP and GDP-bound forms, G-GTP and G-GDP, the di and triphosphate guanosine nucleotides, GTP and GDP, and the inorganic phosphate P_i. Let us consider all the reactions as being reversible, as it is necessary for microscopic reversibility. The non-catalyzed hydrolysis of free GTP is also added to the network. This reaction network is referred to as the elementary GTPase network or module (figure 3.1). The full set of stoichiometric equations describing the reaction network reads:

$$\begin{aligned} G + GTP &\underset{k_{t-}}{\overset{k_{t+}}{\rightleftharpoons}} G\text{-}GTP \qquad & G + GDP &\underset{k_{d-}}{\overset{k_{d+}}{\rightleftharpoons}} G\text{-}GDP \\ G\text{-}GDP + P_i &\underset{k_{a-}}{\overset{k_{a+}}{\rightleftharpoons}} G\text{-}GTP \qquad & GDP + P_i &\underset{k_{h-}}{\overset{k_{h+}}{\rightleftharpoons}} GTP \end{aligned} \tag{3.4}$$

The naming conventions presented previously are applied for reactions. In the reaction labels, t refers to GTP and d to GDP. The labels a (for "activity") and h (for "hydrolysis") refer to the association of P_i with GTPase-bound and free GDP, respectively. The reaction h is present in all the reaction networks of this chapter, designing the reversible hydrolysis synthesis reaction.

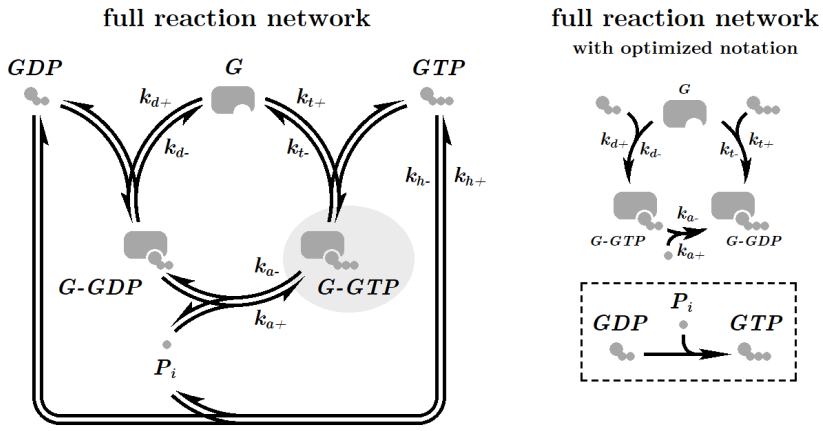


Figure 3.1: **Cartoon description of the GTPase module reaction network.** The left scheme fully describes the reaction network with stoichiometric equations. The right scheme describes the same network using a simplified notation, using simple arrows for reversible reactions, directed forward, and separating the hydrolysis to bring the focus to the GTPase cycle.

3.2.3 A GTPase with partners

GTPase partners are proteins and enzymes capable of associating with GTPases. They can be regulatory partners, such as GAPs, GEFs and GDIs, but also effectors [58]. In this subsection, we present different networks extending the elementary GTPase network with one or more partner species. We consider two types of partners: **Generic partners**, which can associate with the GTPase in any of its states, and **strict state-specific partners**, which can only bind it in one specific state (active or GDP-bound inactive).

3.2.3.1 A GTPase with one generic partner

Let us consider a system with one GTPase and a generic partner X. As a generic partner, X can bind to any form of the GTPase: active, inactive, or nucleotide-free. The partner-bound form of the GTPase will undergo its own elementary GTPase cycle, distinct from the partner-free one as it possesses its own set of kinetic parameters (figure 3.2). The reaction network consists of the

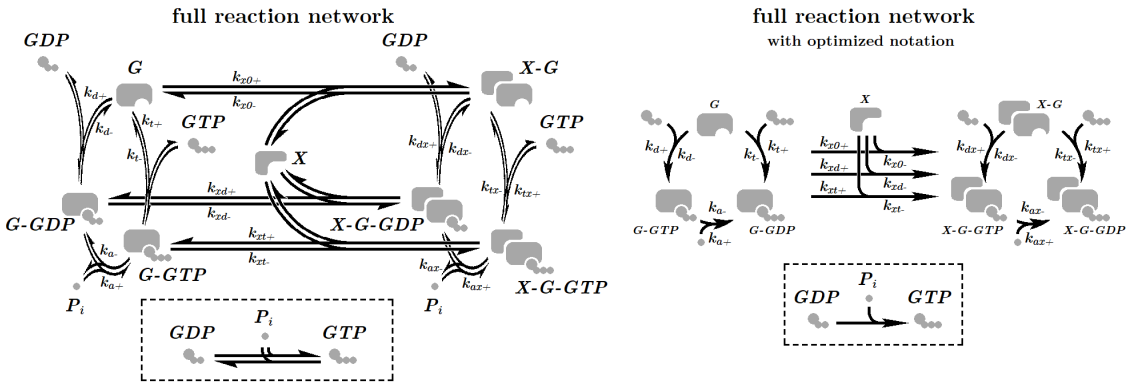


Figure 3.2: **Cartoon description of the reaction network for a GTPase with one generic partner.** This reaction network extends the elementary GTPase network with a generic partner species X binding the GTPase in any state. Both schemes represent the same network. The left one uses a more explicit notation with reversible double-harpoons, while the right one uses the simplified notation detailed in figure 3.2: Reversible reactions are noted with arrows going in the direction of the forward (+) rate. Moreover, in the simplified notation, the association reactions referring to the same partner are grouped, linking the GTPase cycles.

following set of stoichiometric equations:

$$
\begin{array}{ll}
G + GTP \underset{k_{t-}}{\overset{k_{t+}}{\rightleftharpoons}} G\text{-}GTP & G + GDP \underset{k_{d-}}{\overset{k_{d+}}{\rightleftharpoons}} G\text{-}GDP \\
G\text{-}GDP + P_i \underset{k_{a-}}{\overset{k_{a+}}{\rightleftharpoons}} G\text{-}GTP & X\text{-}G + GTP \underset{k_{tx-}}{\overset{k_{tx+}}{\rightleftharpoons}} X\text{-}G\text{-}GTP \\
X\text{-}G + GDP \underset{k_{dx-}}{\overset{k_{dx+}}{\rightleftharpoons}} X\text{-}G\text{-}GDP & X\text{-}G\text{-}GDP + P_i \underset{k_{ax-}}{\overset{k_{ax+}}{\rightleftharpoons}} X\text{-}G\text{-}GTP \\
X + G \underset{k_{x0-}}{\overset{k_{x0+}}{\rightleftharpoons}} X\text{-}G & X + G\text{-}GTP \underset{k_{xt-}}{\overset{k_{xt+}}{\rightleftharpoons}} X\text{-}G\text{-}GTP \\
X + G\text{-}GDP \underset{k_{xd-}}{\overset{k_{xd+}}{\rightleftharpoons}} X\text{-}G\text{-}GDP & GDP + P_i \underset{k_{h-}}{\overset{k_{h+}}{\rightleftharpoons}} GTP
\end{array}
\tag{3.5}
$$

The uncatalyzed hydrolysis h and the partner-free GTPase cycle reactions t, d and h are inherited from the previously presented elementary GTPase network. The reactions tx, dx and ax are the homologous of t, d and a for the partner-bound GTPase cycle. Those two cycles are linked through the partner-association reactions xt, xd and $x0$, the last one referring the the association of X with the nucleotide-free GTPase.

3.2.3.2 Strict state-specific partner

Some partners such as effectors [44] or GDIs [54] are known to show a preference for certain GTPase states. When a partner has a strong specificity for one particular state of the GTPase, it is possible to use a reaction network where the association/dissociation of the partner is restricted to this

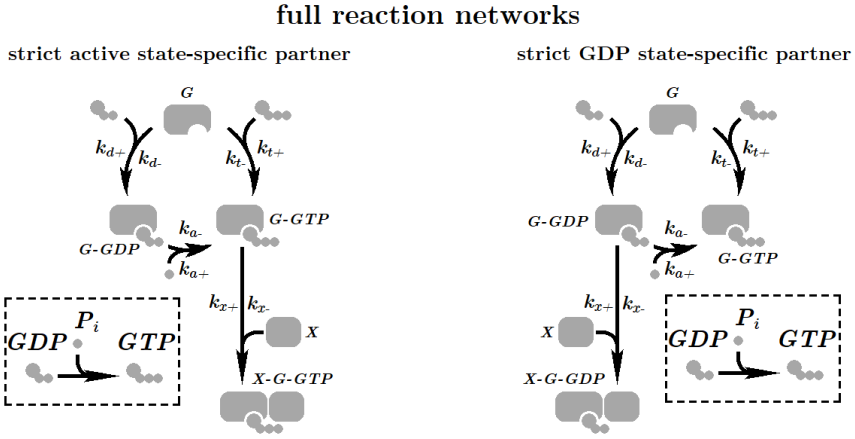


Figure 3.3: **Cartoon description of reaction networks for a GTPase with one strict state-specific partner.** Both schemes describe a reaction network of a GTPase with a strict state-specific partner using the simplified notation. In the one on the left, the specificity is toward the active state, while on the other the specificity is toward the GDP-bound state.

specific state (figure 3.3). As a consequence, the partner-bound GTPase does have its own GTPase cycle. This reaction network simply extends the elementary GTPase network with one reaction:

$$G\text{-}GTP + X \underset{k_{x-}}{\overset{k_{x+}}{\rightleftharpoons}} X\text{-}G\text{-}GTP \tag{3.6}$$

or

$$G\text{-}GDP + X \underset{k_{x-}}{\overset{k_{x+}}{\rightleftharpoons}} X\text{-}G\text{-}GDP \tag{3.7}$$

depending on the state preference of the partner. In particular, we will use this reaction network to model the GTP-bound state-specific interaction between the GTPase and the effector.

3.2.3.3 Multiple partners

Signal-transduction systems may contain multiple parameter species. For instance, a GEF and an effector. Here, let us consider the case of a system with n generic partner species $X_1, X_2, \ldots, X_n$ and m strict state-specific partners $U_1, U_2, \ldots, U_m$. The corresponding reaction network is described by

the following set of stoichiometric equations:

$$\left.\begin{array}{c}
X_i\text{-}G + GTP \underset{k_{tx_i-}}{\overset{k_{tx_i+}}{\rightleftharpoons}} X_i\text{-}G\text{-}GTP \\
X_i\text{-}G + GDP \underset{k_{dx_i-}}{\overset{k_{dx_i+}}{\rightleftharpoons}} X_i\text{-}G\text{-}GDP \\
X_i\text{-}G\text{-}GDP + P_i \underset{k_{ax_i-}}{\overset{k_{ax_i+}}{\rightleftharpoons}} X_i\text{-}G\text{-}GTP \\
X_i + G \underset{k_{x_i0-}}{\overset{k_{x_i0+}}{\rightleftharpoons}} X_i\text{-}G \\
X_i + G\text{-}GDP \underset{k_{x_id-}}{\overset{k_{x_id+}}{\rightleftharpoons}} X_i\text{-}G\text{-}GDP \\
X_i + G\text{-}GTP \underset{k_{x_it-}}{\overset{k_{x_it+}}{\rightleftharpoons}} X_i\text{-}G\text{-}GTP
\end{array}\right\} \; i \in \{1, \ldots, n\}$$

$$\left.\begin{array}{c}
U_i + G\text{-}GDP \underset{k_{u_id-}}{\overset{k_{u_id+}}{\rightleftharpoons}} U_i\text{-}G\text{-}GDP \\
U_i + G\text{-}GTP \underset{k_{u_it-}}{\overset{k_{u_it+}}{\rightleftharpoons}} U_i\text{-}G\text{-}GTP
\end{array}\right\} \; i \in \{1, \ldots, m\} \tag{3.8}$$

$$G + GTP \underset{k_{t-}}{\overset{k_{t+}}{\rightleftharpoons}} G\text{-}GTP \qquad G + GDP \underset{k_{d-}}{\overset{k_{d+}}{\rightleftharpoons}} G\text{-}GDP$$

$$G\text{-}GDP + P_i \underset{k_{a-}}{\overset{k_{a+}}{\rightleftharpoons}} G\text{-}GTP \qquad GDP + P_i \underset{k_{h-}}{\overset{k_{h+}}{\rightleftharpoons}} GTP$$

Such a network can be used to model more complex signal-transduction systems.

3.2.4 GPCR signal transduction network

The GPCR signal transduction system is the main focus of this chapter, and also its most complex reaction network. The network is based on the interaction (association/dissociation) of 8 components: The GPCR R, its agonist ligand L, the G_α and $G_{\beta\gamma}$ subunits of the G protein, their respective effectors, E and E', the diphosphate guanosine nucleotide GDP, and the inorganic phosphate P_i. All other species within the reaction network are complexes resulting from the association of those elementary components (figure 3.4). The G_α subunits are GTPases from the RAS superfamily [7], and as such, cycle through the three GTPase states: GTP-bound, GDP-bound and nucleotide-free. Previous studies already applied the GTPase reaction network pattern (referred to as double displacement scheme) to the modeling of GPCR and G proteins [28]. The G protein can interact with an inactive receptor to form a complex that does not activate G_α [11] [52]. In this model, it is assumed that $G_{\beta\gamma}$ helps G_α associate with the GPCR, as it is evoked by Smrcka [48]. However, direct interactions between $G_{\beta\gamma}$ and the receptor are neglected. For the sake of simplicity,

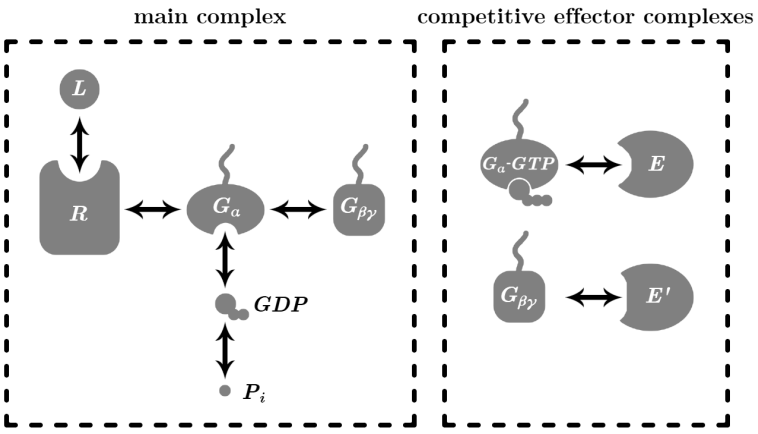


Figure 3.4: **Interactions between the elementary components of the GPCR model.** Description of the interactions within the GPCR signal transduction reaction network, indicating how elementary components interact to form complexes. The elementary components are the GPCR R, its agonist ligand L, the G_α and $G_{\beta\gamma}$ subunits and their respective effectors E and E'. Competitive effectors complexes only form in parallel with the main complex.

the receptor is considered either inactive (R) when ligand-free or active (C) when ligand-bound, though it seems in reality that GPCRs transition between multiple states during their activation [32]. It is the activation of the GPCR by an agonist ligand that allows it to acquire a GEF activity responsible for the activation of the G_α subunit [11] [52]. The $G_{\beta\gamma}$ subunit could also help stabilize G_α's nucleotide-free conformation in the active GPCR-G protein complex, hence allowing the GEF activity [39]. As a consequence, the model considers that the GEF activity of the active GPCR only exists when $G_{\beta\gamma}$ is part of the complex. The model represents a situation where the G_α and $G_{\beta\gamma}$ subunits must be free (from each other and the receptor) to interact with their respective effectors. This choice stems from the fact that at least in some cases, G_α and $G_{\beta\gamma}$ subunits seem to fully dissociate upon activation of G_α [61]. However, it is important to note that in some cases, the effector, the receptor and the G-protein can form one single complex [11]. Also, for the sake of simplicity, it is assumed that the effector E has a strict specificity toward the GTP-bound state of G_α. As a consequence, the E-G_α-GTP complex does not undergo a GTPase cycle. This choice is motivated by the fact that Ras GTPases effectors are known to have a higher affinity with the GTP-bound state [44]. The full reaction network is represented in figure 3.5.

Multiple kinetic models for GPCRs and related systems have already been proposed [28] [31] [51] [24]. In particular, Stein et Ehlert proposed a similar detailed model, also accounting for microscopic reversibility [51]. The particularity of the model presented in this chapter is that it accounts for the dissociation of the G_α and $G_{\beta\gamma}$ subunits, while keeping a detailed scheme and thermodynamic consistency. The respective downstream effectors of G_α and $G_{\beta\gamma}$ are also present in the model.

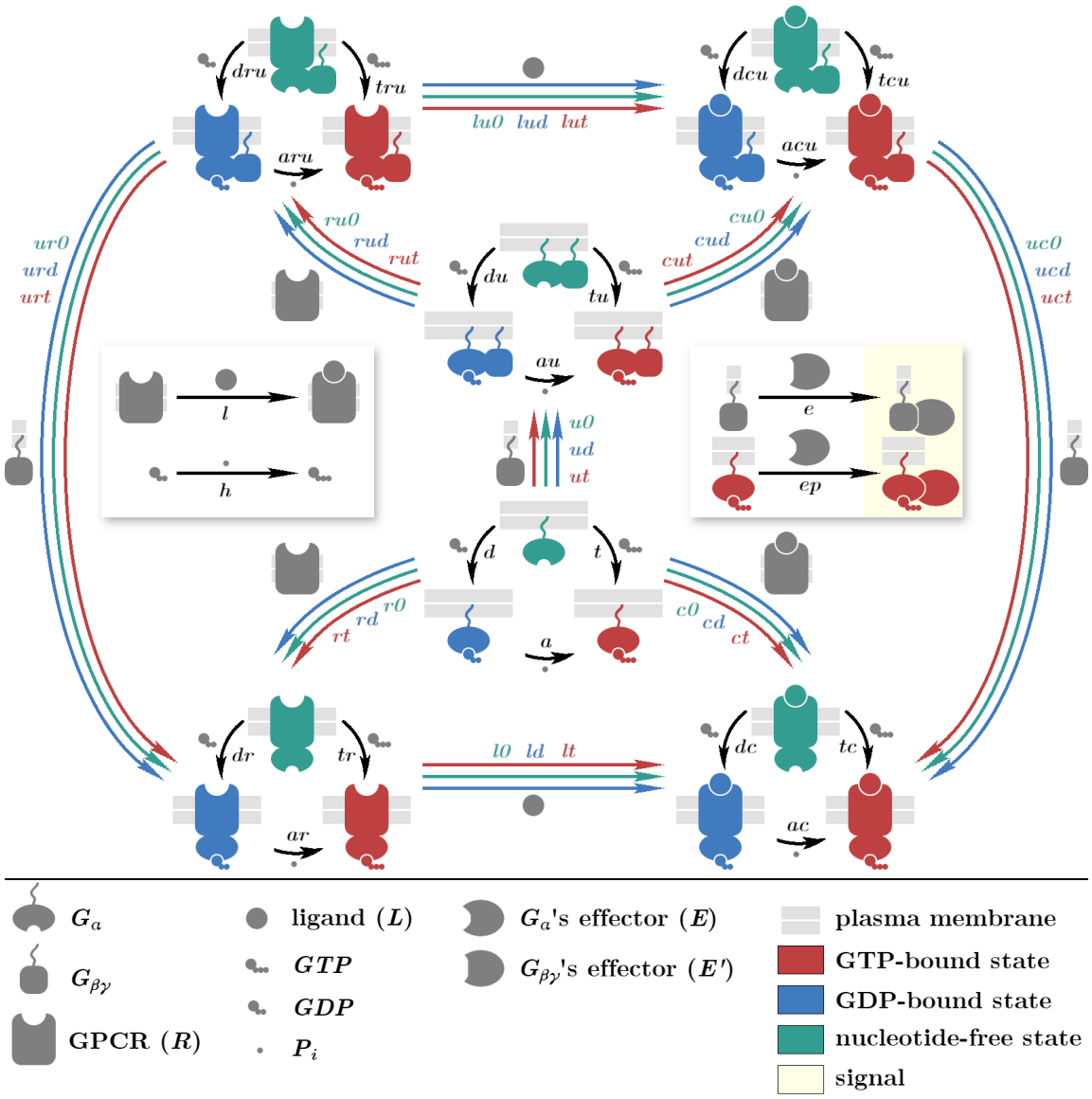


Figure 3.5: **Cartoon representation of the GPCR signal transduction reaction network.** It uses the simplified notation for GTPase cycles, partner association/dissociation and other reactions. To make the scheme easier to read, the partner association/dissociation reactions are also colored depending on the G_α state involved: red for the GTP-bound state, blue for the GDP-bound state and green for the nucleotide-free state, as written in the legend. Reaction arrows are annotated with their respective labels. As an example, the reaction $GTP + P_i \rightleftharpoons GTP$, labeled h, has forward rate constant k_{h+} and reverse rate constant k_{h-}.

3.3 Thermodynamic constraints

When dealing with biochemical reaction networks that work far from equilibrium due to the active metabolism of living organisms, it is important, from a thermodynamic perspective, to identify the sources of non-equilibrium driving. In our case, it is the chemostatted quantities of guanosine nucleotides and inorganic phosphate (GTP, GDP and P_i) that drive the systems out of equilibrium. When the sources of non-equilibrium driving - the chemostats - are turned off, the system must reach, over a certain time, its equilibrium state. This equilibrium state is a steady state where the net current of any reversible reaction vanishes. This condition is known as detailed balance (DB), often also referred to as microscopic reversibility [8]. If we fail to impose this condition, we cannot properly understand how non-equilibrium driving affects the system's properties. In this section, we explain how we managed to apply detailed balance to the different GTPase reaction networks. This entails:

1. Extracting detailed balance constraints from the reaction networks.
2. Applying these constraints to build a set of DB-abiding equilibrium constants.
3. Verifying that the network, with such a set of constraints, indeed respects DB.

3.3.1 What is a detailed balance constraint

The equilibrium constant K is the constant which, for a given reaction, links the kinetic rate constants with the equilibrium concentrations, and can be derived as follows: Let us consider a system of volume V with N species $\{X_0, \ldots, X_{N-1}\}$ and one reversible reaction

$$\sum_i s_i X_i \underset{k_-}{\overset{k_+}{\rightleftharpoons}} \sum_i p_i X_i \tag{3.9}$$

where s_i and p_i are respectively the substrate and product stoichiometric coefficients for the species X_i and k_+ and k_- the forward and reverse reaction rate constants. One can write the forward and reverse reaction currents as

$$\begin{aligned} \Phi^+ &= V k_+ \prod_{i=0}^{N-1} [X_i]^{s_i} \\ \Phi^- &= V k_- \prod_{i=0}^{N-1} [X_i]^{p_i} \end{aligned} \tag{3.10}$$

where $[X_i]$ is the concentration of X_i. Equilibrium is reached whenever detailed balance is respected, i.e. when the net reaction current vanishes

$$\Phi = \Phi^+ - \Phi^- = 0 \tag{3.11}$$

From this, one can write the association equilibrium constant as

$$K = \frac{k_+}{k_-} = \prod_i [X_i]^{(p_i - s_i)} \tag{3.12}$$

For systems where only one reversible reaction is present, as in the previous example, equilibrium is necessarily met at steady state, since their conditions are equivalent in this context:

$$\begin{gathered} \frac{d[X]}{dt} = 0 \\ \begin{pmatrix} \frac{d[X_0]}{dt} \\ \vdots \\ \frac{d[X_{N-1}]}{dt} \end{pmatrix} = S\frac{\Phi}{V} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \\ \Phi^+ = \Phi^- \end{gathered} \tag{3.13}$$

where S is the $N \times 1$ stoichiometric matrix defined as

$$S = \begin{pmatrix} p_0 - s_0 \\ \vdots \\ p_{N-1} - s_{N-1} \end{pmatrix} \tag{3.14}$$

This means that such systems are intrinsically compatible with equilibrium. However, this is generally not the case for systems with multiple reactions. Equilibrium is possible, in a given reaction network containing only reversible reactions, only if the system can reach a steady state where, for any reaction, the forward reaction current matches the reverse one. This is only possible if all the equilibrium constants are compatible with a prescribed set of thermodynamic constraints. A system that is consistent with equilibrium is thus a system where equilibrium constants are compatible with each other. The equilibrium constraints will thus be the relations among equilibrium constants that guarantee such compatibility. For a given system with N species and M reactions, there may be a set of Z constraints $\{\mathcal{C}_0, \ldots, \mathcal{C}_{Z-1}\}$ where each constraint corresponds to an equation that takes the form

$$\prod_{r=0}^{M-1} {K_r}^{n_{r,c}} = 1 \tag{3.15}$$

where $n_{r,c}$ $(c = 0, 1, \ldots, Z-1)$ are some integers.

3.3.2 How to define DB constraints for a given reaction network: The case of the elementary GTPase module

Finding all DB constraints for a given reaction network is a problem that has been addressed in the scientific literature. Colquhoun et al. introduced a method to impose detailed balance within a reaction network by imposing microscopic reversibility for every cycle of the cycle basis of the system state transition graph [8]. The formulated constraints allow them to impose the value of a given kinetic rate constant as a function of the others [8]. Along the same lines, let us define the DB constraints for the elementary GTPase reaction networks. For convenience, we will formulate constraints in terms of equilibrium constants rather than kinetic rate constants. First, we draw the graph representing the different complex states/configurations and how they are related through reactions 3.6, which allows us to visualize the reaction cycles. The GTPase module only contains one cycle (figure 3.6), which gives a unique detailed balance constraint:

$$K_a = K_h \frac{K_t}{K_d} \tag{3.16}$$

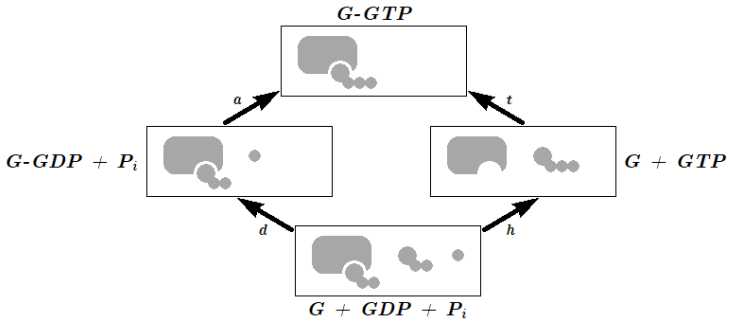


Figure 3.6: **Complex configuration transition graph for the GTPase module.** Each arrow represents the reaction responsible for the corresponding complex formation on the graph. Arrows are annotated with the corresponding reaction labels.

3.3.3 Composition of DB constraints, a modular approach

For large reaction networks, while it is always possible to build a set of constraints (i.e. with a numerical approach where a cycle basis is extracted and processed automatically), it may become difficult to make sense of the ensuing constraints. In the case of GTPase modules, and this is probably true for most biological modules, large reaction networks can be seen as compositions of smaller, simpler sub-networks, or sub-modules. These sub-networks may themselves be the composition of more fundamental networks. When enriching a reaction network with new reactions, pre-existing reaction cycles are still present, and thus, all constraints associated with the base networks still hold: Extending a reaction network can only add new constraints. A composed reaction network will thus have:

1. **Inherited constraints / submodule constraints**: All constraints associated with its sub-modules.
2. **Coupling constraints**: New constraints born from new reaction cycles appearing due to the coupling of two or more submodules that were not present in any of the submodules.

Different submodules may share the same structure/reaction pattern (i.e. the reaction network of partner-free GTPase and that of a partner-bound GTPase). By addressing the simplest/most fundamental modules first (the simple GTPase module in our case), one can progressively build complex reaction networks with their set of DB constraints through a composition approach.

In this section, using the modular/composition approach that we have outlined, we will define the constraints for the different GTPase reaction networks presented previously, proceeding by order of increasing complexity. We begin with the elementary GTPase network - though it has already been addressed. Next, we address the case of the generic and state-specific partner modules. We then consider systems with multiple partners. Finally, we present the constraints for the GPCR signal transduction module.

3.3.4 The elementary GTPase network

The GTPase module is the simplest one. We already determined its constraints in a previous subsection: The unique reaction cycle of the module (figure 3.7) prescribes that

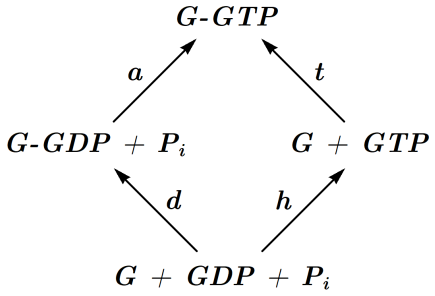


Figure 3.7: **Reaction cycle scheme for the elementary GTPase network.**

$$\frac{K_h K_t}{K_a K_d} = 1 \tag{3.17}$$

On the one hand, K_h is a well-known value, as it can be derived from the free standard Gibbs free energy change of the reaction [2], while values for K_t and K_d have been reported in multiple works [65] [16]. On the other hand, it is much more difficult to find reliable values for K_a in the literature. As a consequence, we will use the DB constraint as a way to determine K_a, as

$$K_a = \frac{K_h K_t}{K_d} \tag{3.18}$$

As the networks grow in complexity, it will become more and more difficult to apply DB constraints to a set of kinetics parameters without an efficient approach. Rather than just enumerating the constraints, we will thus introduce a "protocol" to sequentially apply all the constraints to some set of parameters. Those protocols will appear in boxes throughout this section. The constraint for this module can be applied as follows:

For given values of K_h, K_t, K_d, we determine the values of the remaining constants K_a as $K_a = K_h \frac{K_t}{K_d}$.

3.3.5 Generic partner-GTPase network

The generic partner-GTPase network, as explained earlier, extends the elementary GTPase network by adding the interaction with some partner X. The partner X can bind to every state of the GTPase (GTP-bound, GDP-bound or nucleotide-free) and the partner-bound GTPase thus displays its own GTPase cycle. The diagram representing the reaction cycles for this network (figure 3.8) has the shape of a cube. Two of the faces correspond to elementary GTPase networks, from which we get the constraints

$$K_a = \frac{K_h K_t}{K_d} \qquad K_{a_x} = \frac{K_h K_{t_x}}{K_{d_x}} \tag{3.19}$$

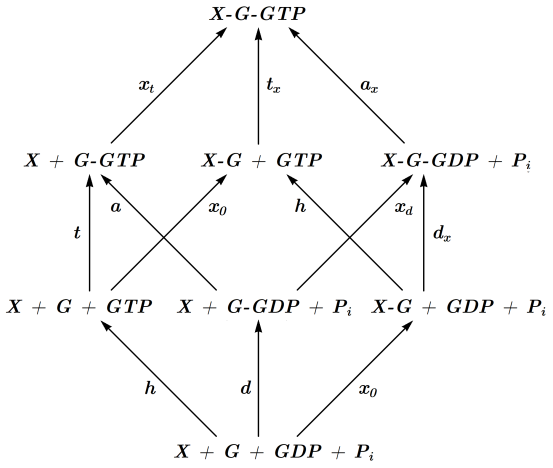


Figure 3.8: **Reaction cycles for the generic partner-GTPase reaction network.**

Then, we have four new constraints arising from the partner-binding reaction cycles

$$\begin{aligned} \frac{K_d K_{x_d}}{K_{d_x} K_x} = 1 \qquad \frac{K_t K_{x_t}}{K_{t_x} K_x} = 1 \\ \frac{K_a K_{x_t}}{K_{a_x} K_{x_d}} = 1 \qquad \frac{K_h K_x}{K_h K_x} = 1 \end{aligned} \tag{3.20}$$

among which the last one is not a constraint, and the third one is respected if the four others are. From this point on, we will be dealing with multiple reaction cycles, and thus, multiple constraint equations. As a consequence, identifying smart protocols is going to become a crucial issue. In the case of the present network, let us make a distinction between two approaches to apply the constraint: The first approach consists of imposing the nucleotides binding dynamics of the partner-bound GTPase and determining the subsequent partner binding dynamics:

For given values of K_h, K_t, K_d, K_{t_x}, K_{d_x} and K_{x_d} we determine the values of the 4 remaining constants K_a, K_{a_x}, K_{x_t} and K_x, as

1. $K_a = K_h \frac{K_t}{K_d}$
2. $K_{a_x} = K_h \frac{K_{t_x}}{K_{d_x}}$
3. $K_{x_t} = K_{x_d} \frac{K_{t_x}}{K_{d_x}} \frac{K_d}{K_t}$
4. $K_x = K_{x_t} \frac{K_t}{K_{t_x}}$

In the second approach, the partner binding dynamics is imposed and we determine the subsequent nucleotide binding dynamics for the partner-bound GTPase:

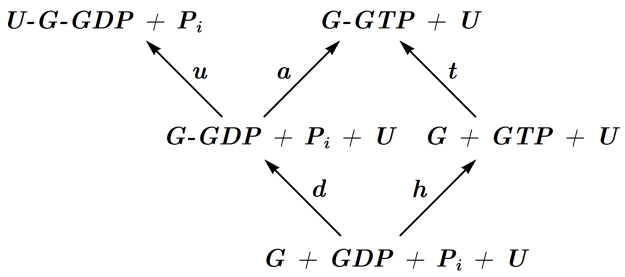


Figure 3.9: **Reaction cycles scheme for the reaction network associated with a GTPase with one strict state-specific partner.**

For given values of K_h, K_t, K_d, K_{x_t}, K_{d_x} and K_{x_d} we determine the values of the 4 remaining constants K_a, K_{a_x}, K_{t_x} and K_x, as

1. $K_a = K_h \frac{K_t}{K_d}$
2. $K_{t_x} = K_{d_x} \frac{K_{x_t}}{K_{x_d}} \frac{K_t}{K_d}$
3. $K_{a_x} = K_h \frac{K_{t_x}}{K_{d_x}}$
4. $K_x = K_{x_t} \frac{K_t}{K_{t_x}}$

In the case of strictly state-specific partners, however, there are no additional constraints, as the partner association reaction does not bring any new reaction cycles into the network (figure 3.9).

3.3.6 The case of a network with multiple competitive GTPase partners

Competitive partners are independent in terms of detailed balance constraints, which means that detailed balance can be imposed for a module with any number n of generic partners $X_1, X_2, ..., X_n$, and m of state-specific partners $U_1, U_2, ..., U_m$ by simply using the constraints detailed in the previous subsections. Since state-specific partners do not bring any constraints, only generic effector constraints are considered:

$$\begin{aligned} & K_a = \tfrac{K_h K_t}{K_d} \\ & \left.\begin{aligned} & K_{ax_i} = \frac{K_h K_{tx_i}}{K_{dx_i}} \\ & \frac{K_d K_{x_i d}}{K_{dx_i} K_{x_i}} = 1 \\ & \frac{K_t K_{x_i t}}{K_{tx_i} K_{x_i}} = 1 \end{aligned} \right\} i \in \{1, \ldots, n\} \end{aligned} \tag{3.21}$$

Those constraints are then applied using the same approach as before

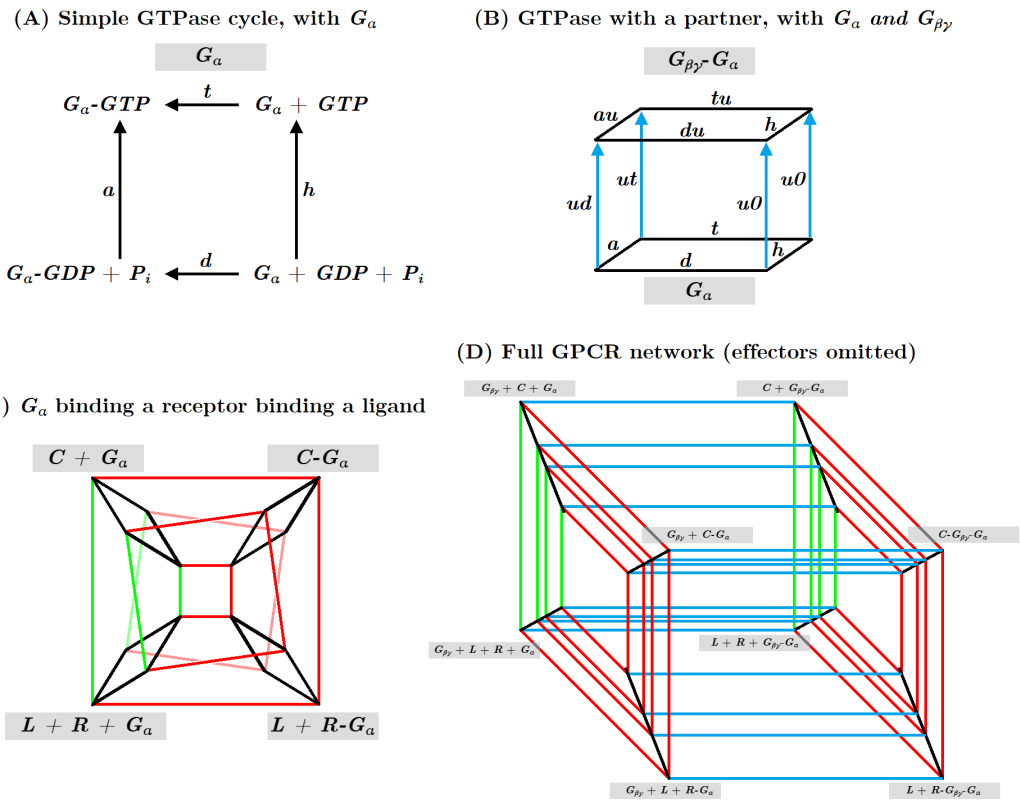


Figure 3.10: **Reaction cycle graph for the GPCR reaction network.**

For given values of K_h, K_t, K_d and K_{tx_i}, K_{dx_i}, K_{x_id} we determine the values of the 4 remaining constants K_a and K_{ax_i}, K_{x_it}, K_{x_i}, $i \in \{1, \ldots, n\}$ as

1. $K_{x_it} = K_{x_id} \frac{K_{tx_i}}{K_{dx_i}} \frac{K_d}{K_t}$
2. $K_a = K_h \frac{K_t}{K_d}$
3. $K_{ax_i} = K_h \frac{K_{tx_i}}{K_{dx_i}}$
4. $K_{x_i} = K_{x_it} \frac{K_t}{K_{tx_i}}$

3.3.7 The GPCR network

The GPCR reaction network is by far the most complex. However, its constraints can be determined using the same approach as in the previous networks. We can draw the reaction cycle graph for the GPCR network (figure 3.10). Since G_α's effectors are defined as strictly GTP-state specific and

$G_{\beta\gamma}$ does not have multiple states, the effectors association reactions will not add any constraints and thus do not need to be represented on the cycle graph. The constraints for the GPCR network include the ones from the elementary GTPase network,

$$\begin{array}{cc} K_{ar} = K_{tr}K_h{K_{dr}}^{-1} & K_{ac} = K_{tc}K_h{K_{dc}}^{-1} \\ K_{au} = K_{tu}K_h{K_{du}}^{-1} & K_{aru} = K_{tru}K_h{K_{dru}}^{-1} \\ K_{acu} = K_{tcu}K_h{K_{dcu}}^{-1} & \end{array} \tag{3.22}$$

the generic partner-GTPase network constraints

$$\begin{array}{cc} K_{rt} = K_{rd}K_{tr}{K_{dr}}^{-1}K_d{K_t}^{-1} & K_{r0} = K_{rt}K_t{K_{tr}}^{-1} \\ K_{ct} = K_{cd}K_{tc}{K_{dc}}^{-1}K_d{K_t}^{-1} & K_{c0} = K_{ct}K_t{K_{tc}}^{-1} \\ K_{ut} = K_{ud}K_{tu}{K_{du}}^{-1}K_d{K_t}^{-1} & K_{u0} = K_{ut}K_t{K_{tu}}^{-1} \\ K_{rut} = K_{rud}K_{tru}{K_{dru}}^{-1}K_{du}{K_{tu}}^{-1} & K_{ru0} = K_{rut}K_{tu}{K_{tru}}^{-1} \\ K_{urt} = K_{urd}K_{tru}{K_{dru}}^{-1}K_{dr}{K_{tr}}^{-1} & K_{ur0} = K_{urt}K_{tr}{K_{tru}}^{-1} \\ K_{cut} = K_{cud}K_{tcu}{K_{dcu}}^{-1}K_{du}{K_{tu}}^{-1} & K_{cu0} = K_{cut}K_{tu}{K_{tcu}}^{-1} \\ K_{uct} = K_{ucd}K_{tcu}{K_{dcu}}^{-1}K_{dc}{K_{tc}}^{-1} & K_{uc0} = K_{uct}K_{tc}{K_{tcu}}^{-1} \end{array} \tag{3.23}$$

as well as some new constraints

$$\begin{array}{cc} K_{rud} = K_{rd}{K_{ud}}^{-1}K_{urd} & K_{cud} = K_{cd}{K_{ud}}^{-1}K_{ucd} \\ K_{l0} = K_lK_{tr}{K_{tc}}^{-1}K_{ct}{K_{rt}}^{-1} & K_{lt} = K_{l0}K_{tc}{K_{tr}}^{-1} \\ K_{ld} = K_{l0}K_{dc}{K_{dr}}^{-1} & K_{lu0} = K_lK_{tru}{K_{tcu}}^{-1}K_{cut}{K_{rut}}^{-1} \\ K_{lut} = K_{lu0}K_{tcu}{K_{tru}}^{-1} & K_{lud} = K_{lu0}K_{dcu}{K_{dru}}^{-1} \end{array} \tag{3.24}$$

All those constraints can be applied as

For given values of the free constants K_d, K_t, K_h, K_a, K_e, K_{ep}, K_l, K_{rd}, K_{dr}, K_{tr}, K_{cd}, K_{dc}, K_{tc}, K_{ud}, K_{du}, K_{tu}, K_{urd}, K_{tru}, K_{dru}, K_{ucd}, K_{tcu} and K_{dcu} we determine the values of the constrained constants as

1. $K_{rt} = K_{rd}K_{tr}{K_{dr}}^{-1}K_d{K_t}^{-1}$
2. $K_{r0} = K_{rt}K_t{K_{tr}}^{-1}$
3. $K_{ct} = K_{cd}K_{tc}{K_{dc}}^{-1}K_d{K_t}^{-1}$
4. $K_{c0} = K_{ct}K_t{K_{tc}}^{-1}$
5. $K_{ut} = K_{ud}K_{tu}{K_{du}}^{-1}K_d{K_t}^{-1}$
6. $K_{u0} = K_{ut}K_t{K_{tu}}^{-1}$
7. $K_{rud} = K_{rd}{K_{ud}}^{-1}K_{urd}$
8. $K_{rut} = K_{rud}K_{tru}{K_{dru}}^{-1}K_{du}{K_{tu}}^{-1}$
9. $K_{ru0} = K_{rut}K_{tu}{K_{tru}}^{-1}$
10. $K_{urt} = K_{urd}K_{tru}{K_{dru}}^{-1}K_{dr}{K_{tr}}^{-1}$
11. $K_{ur0} = K_{urt}K_{tr}{K_{tru}}^{-1}$

12. $K_{cud} = K_{cd}{K_{ud}}^{-1}K_{ucd}$
13. $K_{cut} = K_{cud}K_{tcu}{K_{dcu}}^{-1}K_{du}{K_{tu}}^{-1}$
14. $K_{cu0} = K_{cut}K_{tu}{K_{tcu}}^{-1}$
15. $K_{uct} = K_{ucd}K_{tcu}{K_{dcu}}^{-1}K_{dc}{K_{tc}}^{-1}$
16. $K_{uc0} = K_{uct}K_{tc}{K_{tcu}}^{-1}$
17. $K_{l0} = K_l K_{tr}{K_{tc}}^{-1}K_{ct}{K_{rt}}^{-1}$
18. $K_{lt} = K_{l0}K_{tc}{K_{tr}}^{-1}$
19. $K_{ld} = K_{l0}K_{dc}{K_{dr}}^{-1}$
20. $K_{lu0} = K_l K_{tru}{K_{tcu}}^{-1}K_{cut}{K_{rut}}^{-1}$
21. $K_{lut} = K_{lu0}K_{tcu}{K_{tru}}^{-1}$
22. $K_{lud} = K_{lu0}K_{dcu}{K_{dru}}^{-1}$
23. $K_{ar} = K_{tr}K_h{K_{dr}}^{-1}$
24. $K_{ac} = K_{tc}K_h{K_{dc}}^{-1}$
25. $K_{au} = K_{tu}K_h{K_{du}}^{-1}$
26. $K_{aru} = K_{tru}K_h{K_{dru}}^{-1}$
27. $K_{acu} = K_{tcu}K_h{K_{dcu}}^{-1}$

3.3.8 Evaluating the constraints

Once we have defined the detailed balance constraints for a given reaction network and established a protocol to build a set of equilibrium constants respecting these constraints, we may want to verify that systems built with such a set of equilibrium constants indeed respect detailed balance in equilibrium conditions. Since we are going to deal with many different reaction networks, let us define a protocol to test the quality of the detailed balance constraints for a given reaction network. A proper set of DB constraints allows the reaction network to reach a state of equilibrium in a closed system, characterized by the absence of net reaction currents, as explained earlier. We can thus validate the constraints by checking that the steady states obtained from simulations without chemostats indeed evolve to a steady state with zero net currents when the constraints are applied. We introduce the following protocol:

1. First, we set the values of the equilibrium constants. If the constraints are applied, unconstrained equilibrium constants are assigned a random value, while constrained ones are set according to the constraints. If the constraints are not applied, all equilibrium constants are given a random value. Random values are defined as $K = 10^u$ where u is drawn from a uniform random distribution in the $(-2, 2)$ interval.
2. From the full set of equilibrium constants, we define for every reaction r the forward reaction rate constant $k_{r+} = K_r$ and the reverse reaction rate constant $k_{r-} = 1$.

3. Now that the full set of reaction rate constants is defined, we integrate the corresponding rate equations, considering initial quantities of 1 for every species, over a sufficiently long time to ensure convergence to a steady state. Alternatively, we could have determined the steady state as the null space of the transition rate matrix of the associated master equations.

4. Using the final state, we compute for each reaction r the net reaction current Φ_r as the difference between the forward and the reverse currents, as described in equation 3.11.

5. Finally, we compute the norm of the net current vector:

$$|\Phi| = \sqrt{\sum_r {\Phi_r}^2} \tag{3.25}$$

   This value should converge to 0 as the system tends toward equilibrium.

6. We repeat steps 1 to 7 a large number of times, with and without enforced constraints. We plot and compare the distribution of the net current vector norms generated with and without the constraints. If the constraints have been computed correctly, we should observe that the distribution associated with the constraints peaks around zero (to machine precision).

Using this method, we evaluated the different protocols for building sets of constraints (figure 3.11). Detailed balance is successfully applied with the constraints in the case of networks with 0 to 4 generic partners and 0 to 4 state-specific partners (figure 3.11 a, b). This covers the case of the elementary GTPase model and that of single generic (with the first protocol) and state-specific partners as well. The cases of both GTP-bound state-specific and GDP-bound state-specific partners are considered. The second protocol evoked for imposing constraints in the generic partner module is validated separately (figure 3.11 c) Finally, the protocol for building the constraints for the GPCR network is validated in figure 3.11 d.

3.4 Parameter values

Gathering an extensive and exhaustive set of parameters for reaction-diffusion models is a tedious and time-consuming task. Moreover, measurements/data for the exact processes we are interested in are rarely available, thus building a set of parameters is all about finding reasonable approximations. As a consequence, the set of parameters must be subject to discussion and the process by which values are taken or extrapolated from the literature must be explicit and transparent. In this section, we gather kinetic and quantitative parameter values for the different GTPase models.

When in the literature values are reported without a specified uncertainty, we proceed to computations (average, etc.) using the available significant figures. Computations involving uncertainties are generally carried out using the *uncertainties* python package [34]. The aim of this work is to constitute a coherent set of "exact" values to be used as simulation parameters, consistent with values reported in the scientific literature.

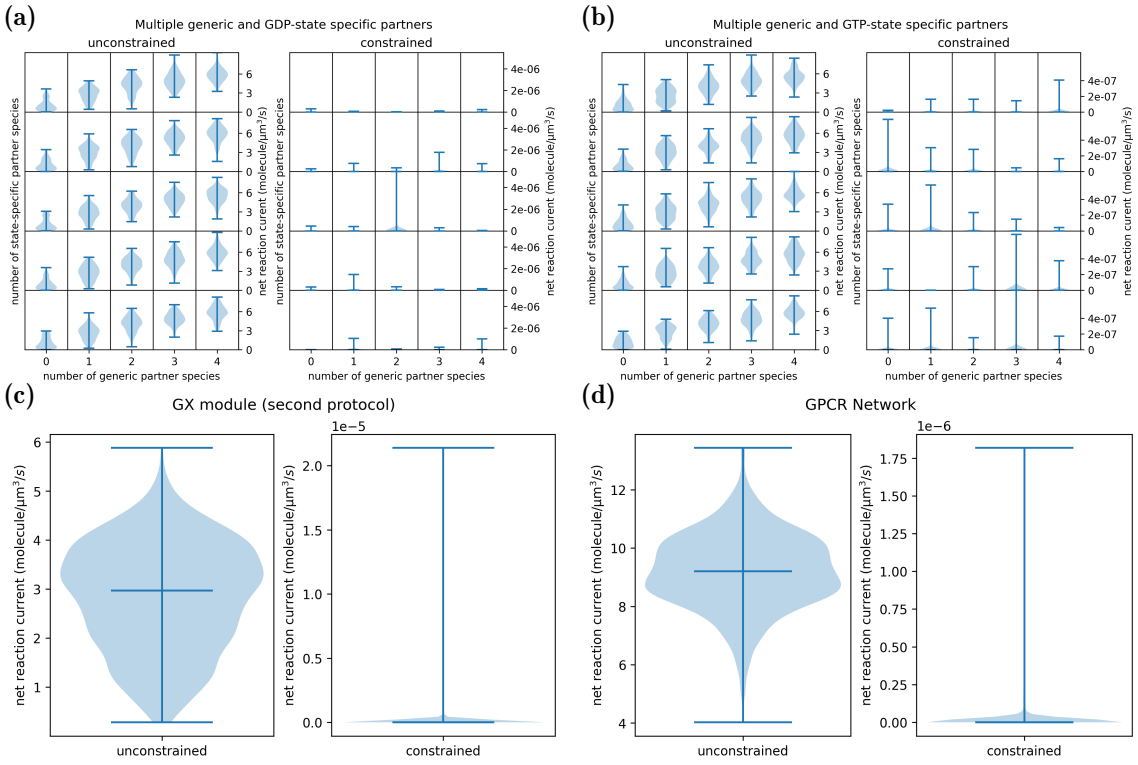


Figure 3.11: **Validating the constraints for the different reaction networks.** In each plot, the "constrained" and "unconstrained" conditions respectively correspond to situations where the constraints were and were not applied. When constraints were applied, only free equilibrium constants were randomized, while the constrained ones were set according to the constraints. When constraints were not applied, all equilibrium constants were randomized. In each situation, we plot the distribution of the total net current for 100 (a, b) or 1000 (c, d) independent realizations. For (a) and (b) The plot is realized with systems containing 0 to 4 GDP (a) or GTP-bound (b) state-specific partner species and 0 to 4 generic partner species. For (c), the plot is realized for the GTPase-generic partner module using the constraints from the second protocol presented in the corresponding section, which constrains the GTP association equilibrium constant. For (d), the plot is realized for the GPCR network.

3.4.1 Species quantities

3.4.1.1 Nucleotides and inorganic phosphate

Ref. [55] reports quantitative information regarding nucleotides and related molecules in mammals. We will use the average of the values they report in human cells for the different nucleotides and for inorganic phosphate as reference concentrations for those species

Species	Concentration	Reference
GTP	1.33×10^5 molecule/µm^3	[55]
GDP	1.5×10^4 molecule/µm^3	[55]
Pi	1.67×10^6 molecule/µm^3	[55]

3.4.1.2 GPCRs

Wang et al [60] measured the density of the CXCR4 receptor in 4 breast cancer cell lines. The average of the mean density reported for each cancer cell line is 84.03 molecule µm^{-2} (and 19.6 molecule µm^{-2} for the standard deviation) [60]. Based on this average value, we shall consider a quantity of 84 molecule µm^{-2} for our reference GPCR density.

Species	Concentration	Reference
R_0	84 molecule/µm^2	[60]

3.4.1.3 GTPases

Fujioka et al. report concentrations for the Ras GTPase in various cell types, both measured by them and from other works [19]. We shall use the average of those values as a reference concentration for a small GTPase: 2.6×10^2 molecule/μm^3 [19]. It is hard to find quantitative information about G proteins, moreover, the few we could find report different values: Pugh et al. report densities of G proteins of 3000 molecule/µm^2 (in mammals) in the context of rhodopsin signaling [12], while Falkenburger et al. suggest a quantity of 40 molecule/µm^2, as parameters for their own model, in the context of muscarinic (M1) receptor signaling [15]. Faced with such different values, we shall use the small GTPase reference density as a reference for the total density of both GTPases and G proteins, G_0. This also suggests that the quantity of G proteins could be an interesting parameter to study per se. We note that the **Guesstimation** [63] presctiption $G_0 \simeq \sqrt{3000 \times 40} \simeq \mathcal{O}(10^2)$ molecules/μm^3 is in agreement with the reference density.

Species	Concentration	Reference
G_0	2.6×10^2 molecule/µm^3 (whole cell)	[19]

3.4.1.4 GTPase partners

Milo et al. discuss the generic concentrations of signaling molecules in a cell, and propose concentration magnitudes in the 10^{-8} to 10^{-5} M range (1×10^2 molecule/µm^3 to 1×10^5 molecule/µm^3) in a HeLa cell [40]. We propose to use for this time cytoplasmic densities of GTPase partners in this order of magnitude range.

Species	Concentration	Reference
Generic partners (Effectors, GAP, GEF, GDI)	1×10^2 to 1×10^5 molecule/µm^3	[40]

3.4.2 Kinetic parameters

3.4.2.1 The GPCR and its ligand

Vega et al. report the binding kinetic parameters (k_{on}, k_{off} and K_d) of the interaction between the chemokine CXCL12 (the ligand) and CXCR4 (its receptor), measured by plasmon surface resonance, $k_{on} = 4.20 \times 10^5 \pm 0.56 \times 10^5$ M^{-1} s^{-1} and $k_{off} = 8.24 \times 10^{-3} \pm 0.11 \times 10^{-3}$ s^{-1} [57]. We shall base our reference parameter values for ligand binding kinetic rates, k_{l+} and k_{l-}, on their average k_{on} and k_{off} values :

Parameter	Value	Reference
k_{l+}	6.97×10^{-4} µm^3 molecule^{-1} s^{-1}	[57]
k_{l-}	8.24×10^{-3} s^{-1}	[57]

3.4.2.2 Uncatalyzed hydrolysis of GTP into GDP

The hydrolysis of ATP in ADP and P_i has an apparent equilibrium constant of 2.51×10^6 (that can be interpreted as a dissociation constant in M, or 1.51×10^{15} molecule/µm^3) according to its standard transformed Gibbs energy variation $\Delta G^{\circ\prime} = -36.53$ kJ/mol at 25°C, neutral pH and 0.1 M ionic strength, reported by Alberty [2]. We use the value for a 0.1 M ionic strength as it seems to be a reasonable value for a Eukaryotic ionic strength, (Liu at al. report ionic strengths, measured with various sensors in eukaryotic (HEK293) cells, above 100 mM [36]), even if the value at 0.25 M could have been a good choice too. In the hypothesis that GTP follows the same dynamics as ATP in that regard, we find it reasonable to use this value as a reference for the GTP hydrolysis reaction association constant.

According to Sigma-Aldrich's product datasheet, ATP is stable one week at 0°C and neutral pH [47]. Assuming it would take a few days (3 days) to degrade (99% of) ATP at room temperature and neutral pH, we get $k_{h-} = -ln(0.01)/259200 \simeq 1.8 \cdots \times 10^{-5} \text{s}^{-1} \simeq 10^{-5} \text{s}^{-1}$ (order of magnitude). However, this reaction will be ignored most of the times, since nucleotides and inorganic phosphate concentrations are chemostated in cytoplasmic conditions. Also, in simulation scripts where chemostats are set, the actual values assigned to k_{h+} and k_{h-} may vary, but this has no effect on the results.

Parameter	Value	Reference
K_h	6.61×10^{-16} µm^3/molecule	[2]
k_{h-}	10^{-5} s^{-1}	guessed based on [47]

3.4.2.3 GTPase cycle reactions

Zhang et al. [65] and Fiegen et al. [16] both investigate the binding kinetics of GTP and GDP analogs to multiple GTPases, including Rac1. In this thesis, we will use the data from Fiegen et al. as they report a full set of parameters (nucleotide exchange dissociation constants and dissociation rate constants and hydrolysis k_{cat}) for Rac1 and Rac1b [16]. We shall use the values for Rac1 from

Fiegen et al. [16] as references for our simulation parameters (we take the inverse of the dissociation constant, as we are interested in the association constant). Still, one should keep in mind that the values reported by Fiegen et al [16] differ greatly from those reported by Zhang et al. [65].

Parameter	Value	Reference
K_d	$59\,\mu m^3$/molecule	[16]
K_t	$25\,\mu m^3$/molecule	[16]
k_{d-}	$7.0 \times 10^{-5}\,s^{-1}$	[16]
k_{t-}	$1.0 \times 10^{-4}\,s^{-1}$	[16]
k_{a-}	$1.8 \times 10^{-3}\,s^{-1}$	[16]

Forward association rate constants are derived from the association constants and dissociation rate constants while K_a is obtained from the network's constraints (equation 3.18).

3.4.2.4 GTPase association/dissociation with effector proteins

Bergbrede et al. measure the binding kinetics of GTP-bound Rab6a (using a GTP analog that cannot be hydrolyzed) with three of its effectors at 25°C, for which the averages kinetic rate constants are $k_{on} = 4.37 \times 10^6\,M^{-1}\,s^{-1}$ and $k_{off} = 9.17\,s^{-1}$ [3]. We shall use those values as reference parameter values for the rate constants of the association/dissociation of GTP-bound GTPases with an effector.

Parameter	Value	Reference
k_{e+}	$7.26 \times 10^{-3}\,\mu m^3\,\text{molecule}^{-1}\,s^{-1}$	[3]
k_{e-}	$9.17\,s^{-1}$	[3]

Linnemann et al also report values at 26°C for Ras and Rap1A [35].

3.5 Signal transduction

GTPases are signal transducers [7]. On activation, triggered by the onset of some input signal, they associate with downstream effector partners, propagating the signal [44]. In this section, we formalize how the input and output signals are quantified depending on the system and which are the important aspects to consider when doing so.

3.5.1 The signal transduction function

Signal transduction is an input-output process. One can define for a given signal transduction system a signal transduction function $\mathcal{S}$ as

$$out = \mathcal{S}(in, t, \overrightarrow{\lambda}) \tag{3.26}$$

where *out* is the output signal, *in* is the input signal, t is the time and $\overrightarrow{\lambda}$ is a vector of system parameters. While the input and output signals are always related to the quantity of some chemical species at time t in the system, their nature depends on the system in question. The output signal is related either to the active GTPase or to the activated effector. In both cases, the signal can be related to multiple species, for example all the active-GTPase species. Since we only consider

situations where the total quantity of GTPase G_0 is conserved in the system, the output signal can be defined as the quantity of the output species normalized by G_0:

$$\mathcal{S}(in,t) = \frac{[output\ species]_t}{G_0} \tag{3.27}$$

where $[output\ species]_t$ corresponds to the total density of all the output species at time t. In the absence of an input signal, the system should converge to a basal steady state $\mathcal{S}(0,\infty)$. The output species always involve GTPases, whose total density is conserved in the system. As a consequence, the output signal has a value in the $[0,1]$ interval. Considering the basal signal level, it is possible to define the relative signal increase $\Delta\mathcal{S}$ as

$$\Delta\mathcal{S}(in,t) = \mathcal{S}(in,t) - \mathcal{S}(0,\infty) \tag{3.28}$$

This function reflects the output signal increase relative to the basal state a time t after stimulation by the onset of an input signal *in*.

3.5.2 The output signal as the quantity of activated effector

Ultimately, GTPases transduce the signal by associating with downstream effectors upon activation [44]. As a consequence, the most straightforward way to define the signal is through the quantity of activated effector. Let us consider the simplest case of a network where a GTPase can bind an effector E that has a strict specificity for the GTP-bound state. In such a system, the signal could be defined as

$$\mathcal{S}(in,t) = \frac{[E\text{-}G\text{-}GTP]_t}{G_0} \tag{3.29}$$

Now, let us consider a more complex case where the effector is not strictly specific to the GTP-bound state. This time, the way the signal is quantified depends on which complex is considered active:

$$\mathcal{S}(in,t) = \begin{cases} \frac{[E\text{-}G\text{-}GTP]_t}{G_0} & \textit{if only active GTPases activate the effector} \\ \frac{[E\text{-}G\text{-}GTP]_t + [E\text{-}G]_t}{G_0} & \textit{if nucleotide-free GTPases activate the effector} \\ \frac{[E\text{-}G\text{-}GTP]_t + [E\text{-}G\text{-}GDP]_t}{G_0} & \textit{if GDP-bound GTPases activate the effector} \\ \frac{[E\text{-}G\text{-}GTP]_t + [E\text{-}G]_t + [E\text{-}G\text{-}GDP]_t}{G_0} & \textit{if all GTPase states activate the effector} \end{cases} \tag{3.30}$$

depending on whether the effector complexes with nucleotide-free and GDP-bound GTPases are considered active or not. However, due to the preference of the effector for the GTP-bound state, there should be little difference between the approaches. For the sake of simplicity, we will generally consider effectors to be strictly GTP-state specific, though this is a consideration to keep in mind.

When the system involves partners that can bind to GTPase along with the effector E, then all complexes involving E and $G-GTP$ should be counted in the signal. For instance, if a partner X can bind the GTPase without competing with E, then the signal is defined as

$$\mathcal{S}(in,t) = \frac{[E\text{-}G\text{-}GTP]_t + [X\text{-}E\text{-}G\text{-}GTP]_t}{G_0} \tag{3.31}$$

3.5.3 The output signal as the quantity of available GTP-bound state GTPase

The quantity of active GTPase can also be used as a proxy for the signal. This is especially useful for systems without effectors, such as the elementary GTPase module for instance. In a simple system with a GTPase and no partners, the signal can be defined as

$$\mathcal{S}(in,t) = \frac{[G\text{-}GTP]_t}{G_0} \tag{3.32}$$

This approach has been previously used by Stanley et al. in their work [50]. For systems involving partners, potential competition with the effector must still be accounted for. Only the quantity of active GTPase available for interaction with potential effectors should be considered. As an example, let us consider a system with 3 partners: one effector E, and two non-effector partners X and Y. The partner X does not compete with E, but Y does. For this system, the signal should thus be defined as

$$\mathcal{S}(in,t) = \frac{[G\text{-}GTP] + [E\text{-}G\text{-}GTP]_t + [E\text{-}X\text{-}G\text{-}GTP]_t + [X\text{-}G\text{-}GTP]_t}{G_0} \tag{3.33}$$

Since effector-bound partners are of course contributing to the signal, and X-G-GTP is likely to bind an effector, as opposed to Y-G-GTP where the Y prevents its association.

3.5.4 The case of the GPCR system

In the case of the GPCR system, the input signal is the conserved total density of the agonist ligand L_0. The case of the GPCR output signal is slightly more complicated. As the active GPCR activates its coupled G protein by acting as a GEF toward the G_α subunit [11], the latter dissociates into free G_α and $G_{\beta\gamma}$ [61]. Both subunits, G_α and $G_{\beta\gamma}$, can then bind their respective downstream effectors [11], E and E'. The GPCR network thus has two parallel output signals associated respectively with E and E'. Their respective signal transduction functions, $\mathcal{S}_\alpha$ and $\mathcal{S}_{\beta\gamma}$, are defined using the density of activated effectors, as

$$\begin{aligned} \mathcal{S}_\alpha(L_0,t) &= \frac{[E\text{-}G_\alpha\text{-}GTP]_t}{G_0} \\ \mathcal{S}_{\beta\gamma}(L_0,t) &= \frac{[E\text{'-}G_{\beta\gamma}]_t}{G_0} \end{aligned} \tag{3.34}$$

where L_0 is the total density of agonist ligand and G_0 the total density of G proteins, equivalent to both the total quantity of G_α subunits, $G_{\alpha 0}$, and of $G_{\beta\gamma}$ subunits, $G_{\beta\gamma 0}$.

3.6 Methods

The different system's models were built numerically using STReNGTHs [18]. Numerical integrations of the rate equations were performed using SciPy [59], while stochastic simulations were performed using the Gillespie algorithm [22]. Only in the case of the elementary GTPase module,

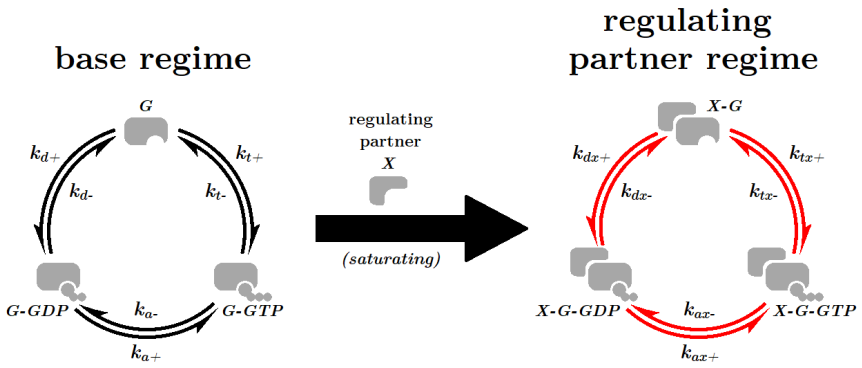


Figure 3.12: **Illustration of the concept of regime for the GTPase module.** The figure illustrates the case of a simple GTPase module, with rates k_{d+}, k_{d-}, k_{t+}, k_{t-}, k_{a+} and k_{a-}. In the absence of regulation, its dynamics is determined solely by those rates. This is the base regime. In the presence of a saturating quantity of some regulating partner X, all GTPases are bound to X, and now follow a dynamics determined by the rates associated with the partner-bound form: k_{dx+}, k_{dx-}, k_{tx+}, k_{tx-}, k_{ax+} and k_{ax-}. We refer to this new dynamic as the partner regime.

the steady state admits an elementary analytical solution, which reads

$$\begin{aligned}\mathcal{S}(in,\infty) &= \frac{A}{B}\\ A &= k^*_{a+}k^*_{d+} + k^*_{a+}k^*_{t+} + k_{d-}k^*_{t+}\\ B &= k^*_{a+}k^*_{d+} + k_{a-}k^*_{d+} + k^*_{d+}k_{t-} + k^*_{a+}k^*_{t+}\\ &\quad + k^*_{a+}k_{t-} + k_{a-}k_{d-} + k_{a-}k^*_{t+} + k_{d-}k^*_{t+} + k_{d-}k_{t-}\end{aligned} \tag{3.35}$$

where k^*_{r+}, for a given reaction r corresponds to $k_{r+}X$ where X is the chemostated density of either GTP, GDP of P_i, depending on which is the substrate of the reaction.

3.7 Results/Discussion

3.7.1 The different regimes of the elementary GTPase cycle

3.7.1.1 Introduction

The activity of GTPases is determined by the interplay between hydrolysis and nucleotide exchange reactions, which constitute the GTPase cycle. It is by changing the rates of those reactions that regulatory proteins such as GEFs, GAPs or GDIs activate/inactivate GTPases [58] [43]. We refer to those alterations in the GTPase cycle as changes of *regime*. A GTPase regime can be defined as a set of kinetic parameter values giving the GTPase cycle a certain behavior (illustrated in figure 3.12). The regime using literature values for the rates, corresponding for example to a GTPase with no partner, is referred to as the **base regime**. The presence of regulating partners can switch the dynamics from its base regime to some regulatory regime. In one of these situations, any rate constant of the cycle can be either increased, decreased or unchanged compared to the base regime. Therefore, there is a total of $3^6 = 729$ possible regimes for the GTPase cycle.

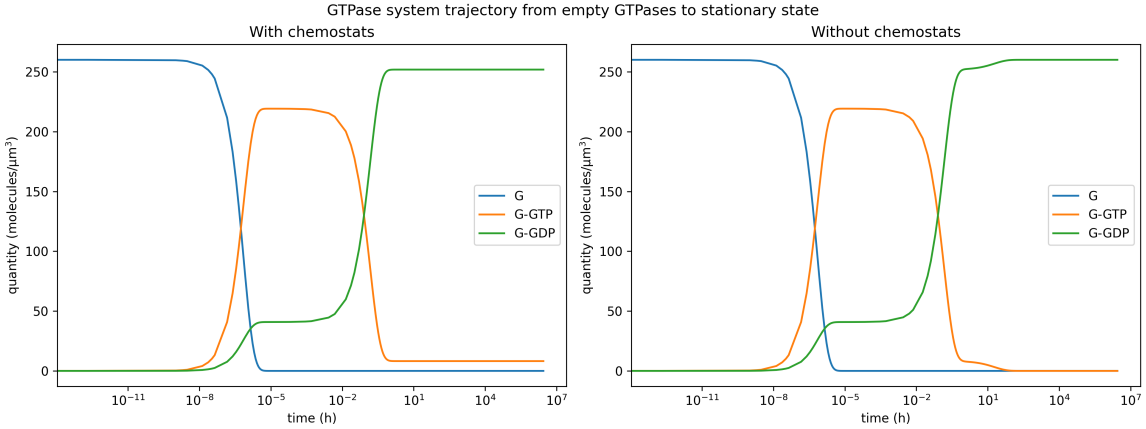


Figure 3.13: **Representative trajectories of the GTPase module.** In the initial state, all GTPases are nucleotide-free. Plots show how the concentration of all 3 GTPase species evolves over time. The left plot shows the trajectory for a system with GTP, GDP and P_i chemostatted, while the right one shows a trajectory for a system without chemostats, which therefore evolves toward thermodynamic equilibrium.

Also, it is important to keep in mind that in a biological context, quantities of nucleotides and inorganic phosphate are chemostated. We will thus mainly study the behavior of GTPase systems in this far-from-equilibrium context, and show how non-equilibrium driving is essential for regulatory regimes to function. In particular, we will explore regimes corresponding to the main regulators of GTPases: GEFs, GAPs and GDIs. This will allow us to establish the basics of GTPase activity/signal regulation. By doing so, we will also discuss the implications of the detailed balance constraints on the signal.

3.7.1.2 The base regime

Let us start with the base regime. We computed the evolution (i.e. integrated the associated rate equations) of the elementary GTPase module, with the standard set of parameters, until the system reached a steady state. In equilibrium conditions (absence of chemostats), the module follows a trajectory in which we may distinguish four phases, occurring at different time scales (Fig. 3.13). First, the GTPases, initially substrate-free, have their active sites quickly occupied with nucleotides. This happens very quickly, in less than 10^{-2} seconds. During this phase, the GTPases mostly associate with GTP due to its excess in the system. The second phase, which unfolds over a few hours, reflects the hydrolytic activity of the GTPase combined with the relatively slow rate of GDP dissociation, and is thus this time in favor of the GDP-bound state. The third and last phase, which extends over a few days, corresponds to the system reaching equilibrium, as the excess GTP gets slowly hydrolyzed. In non-equilibrium conditions (GTP, GDP and P_i chemostated), only the 2 first phases are observed. The non-equilibrium steady state corresponds to the end of the second phase.

From the far-from-equilibrium steady state, we can get the base non-equilibrium steady state signal

of the elementary GTPase system module,

$$\mathcal{S} = \frac{[G\text{-}GTP]}{G_0} = 0.031 \tag{3.36}$$

This base signal is rather small, which means that with the standard set of parameters, the GTPase is not constitutively active. It would need to be positively regulated by a GEF to produce a higher signal.

3.7.1.3 The regulatory regimes

On the one hand, GEFs are known to activate GTPases by accelerating the nucleotide exchange, while on the other hand GAPs and GDIs have inhibitory functions, respectively by accelerating their catalytic activity and decreasing nucleotide dissociation. In order to properly illustrate how those two regulators work, we investigated their respective regimes.

We determined the steady state signal for the G module under different regulatory regimes, corresponding to the supposed actions of GEF, GAP and GDI. For each regime, some rate constants of the GTPase cycle are increased or decreased by some acceleration factor, while others are not. First, we considered four scenarios related to different aspects of the possible action of a GEF:

- **Nucleotide exchange regime**: applying the acceleration factor to the nucleotide association and dissociation reactions rate constant k_{d+}, k_{d-}, k_{t+} and k_{t-}.
- **GDP exchange regime**: applying the acceleration factor only to the GDP association and dissociation rate constants k_{d+} and k_{d-}.
- **Nucleotide dissociation regime**: applying the acceleration factor only to the nucleotide dissociation rate constants k_{d-} and k_{t-}.
- **GDP dissociation regime**: applying the acceleration factor only to the GDP dissociation rate constant k_{d-}.

Then, we considered three alternative regimes, which impact on nucleotide association/dissociation in ways that differ from the GEF regime

- **Nucleotide association regime**: applying the acceleration factor only to the nucleotide association rate constants k_{d+} and k_{t+}.
- **GTP exchange regime**: applying the acceleration factor only to the GTP association and dissociation rate constants k_{t+} and k_{t-}.
- **GTP dissociation regime**: applying the acceleration factor only to the GTP dissociation rate constant k_{t-}.

Finally, we considered a regime associated with the action of a GAP

- **Catalyzed hydrolysis regime**: applying the acceleration factor to the catalyzed hydrolysis forward and reverse reaction rate constants k_{a+} and k_{a-}.

and two regimes associated with the action of a GDI (here, rates are decreased rather than increased)

- **Decreased nucleotide dissociation regime**: applying the deceleration factor to nucleotide dissociation rates k_{d-} and k_{t-}.

- **Decreased nucleotide exchange regime**: applying the deceleration factor to nucleotide association and dissociation rates $k_{d+}, k_{d-}k_{t+}$ and k_{t-}.

Figures 3.14 to 3.17 show the signal as a function of the acceleration factor, in the different regimes, both for equilibrium and non-equilibrium steady states.

As it is expected, the results clearly show that GEF and GAP/GDI regimes have antagonistic effects on the module signal: boosting the exchange increases the signal, while boosting the catalytic hydrolysis or decreasing nucleotide exchange has the opposite effect. However, we observe differences among the different GEF and non-GEF nucleotide association/dissociation regimes.

Only accelerating nucleotide dissociation, and accelerating exchange globally have the same effect for low acceleration factors ($\lesssim 10^4$), while only accelerating association does not increase the signal past its base value. This suggests that it is actually the nucleotide dissociation rates, rather than the exchange itself, that determine the signal increase: Increasing the nucleotide dissociation rate will increase the GTPase signal, while decreasing it will turn it down. Accelerating only GTP exchange globally, or only GTP dissociation, does not produce any signal above the base value. On the contrary, accelerating only GTP dissociation causes the signal to drop to zero, as the affinity between GTP and the GTPase disappears. Instead, accelerating only GDP dissociation leads to a full activation of the GTPases, which is not observed in any other regime. This suggests that the acceleration of k_{d-}, possibly relative to the other rate constants, is the key parameter for GTPase activation. The "nucleotide dissociation" regime brings an interesting aspect of the regulation that is not taken into account in the others: as the nucleotide dissociation rates increase, the affinity between the GTPase and the nucleotides decreases accordingly, which eventually causes the disappearance of the signal altogether due to the absence of association between nucleotides and GTPases. Such a phenomenon, where the GEF activity could lead to GTPase inactivation, has been pointed out before by Stanley et al [50]. It is important to note that only exchange regimes associated with possible GEF effects happen to produce signal past the base value. Other exchange regimes fail to produce a signal increase, or even cause the signal to decrease.

On the side of inhibitory regimes, accelerating hydrolysis (GAP regime), decreasing nucleotide dissociation (first GDI regime) and decreasing nucleotide exchange globally (second GDI regime) all produce the same effect in non-equilibrium conditions, namely the inhibition of the signal.

It is interesting that the activatory and inhibitory effects can be observed without changing the equilibrium constants (affinities), by only accelerating or slowing down reversible reactions. The activatory and inhibitory effects are only achieved in the non-equilibrium context where GTP, GDP and P_i are chemostated. In equilibrium conditions, for the same reaction rate variations, the signal does not increase past its base level in most cases, except in the "GDP dissociation" regime where the signal slightly increases due to the GTPase losing its affinity for GDP. The base level itself is way lower than that of its far-from-equilibrium counterpart, and falls down to 0 when the affinity between the GTPase and the nucleotides decreases too much. At equilibrium, only the equilibrium constants matter, which explains why approaches conserving them leave the signal unchanged at its base level.

For most of the regimes, the K_d/K_t ratio is conserved, either because the equilibrium constants themselves are conserved (i.e. applying acceleration factors both to forward and reverse rate constants) or because K_d and K_t are changed by the same factor. Since we are imposing $K_a = K_h K_t/K_d$ for detailed balance, its value remains unchanged so long as K_t/K_d is conserved. How-

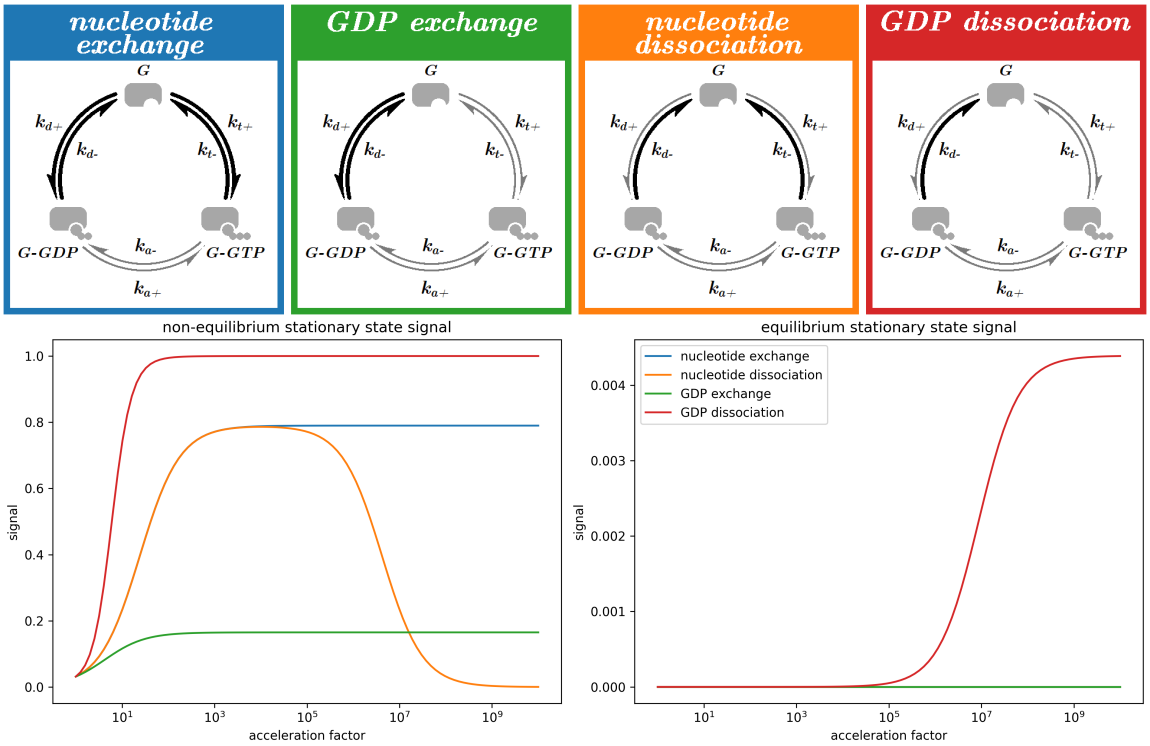


Figure 3.14: **Exploring different GEF regimes for increasing rate acceleration factors.** In each scenario, some reaction rate constants are accelerated - with respect to their standard value - by some acceleration factor. The two bottom plots show the signal at steady state as a function of the rate constant(s) acceleration factor. Left: system driven far from equilibrium - by chemostating GTP, GDP and P_i. Right: system that relaxes to equilibrium (no chemostats).
Four different rate acceleration scenarios corresponding to the regimes of GTPase bound to a GEF are considered (top panels). Those scenarios differ in which rates are accelerated and which are not:
1. **Nucleotide exchange**: association and dissociation rate constant for GTP and GDP (k_{d+}, k_{d-}, k_{t+} and k_{t-}) are accelerated by the same factor. The equilibrium constants are left unchanged.
2. **GDP exchange**: association and dissociation rate constants for GDP (k_{d+} and k_{d-}) are accelerated by the same factor. The equilibrium constants are left unchanged.
3. **Nucleotide dissociation**: only dissociation rate constants for GTP and GDP (k_{d-} and k_{t-}) are accelerated by the same factor. The equilibrium constants K_d and K_t are decreased, but since the factor is the same, the detailed balance constraint is still respected.
4. **GDP dissociation**: only the dissociation rate constant for GDP k_{d-} is accelerated. The equilibrium constants K_d is decreased. The detailed-balance constraints require to adapt K_a, which is done by shifting both k_{a+} and k_{a-}, as described in appendix A.1.

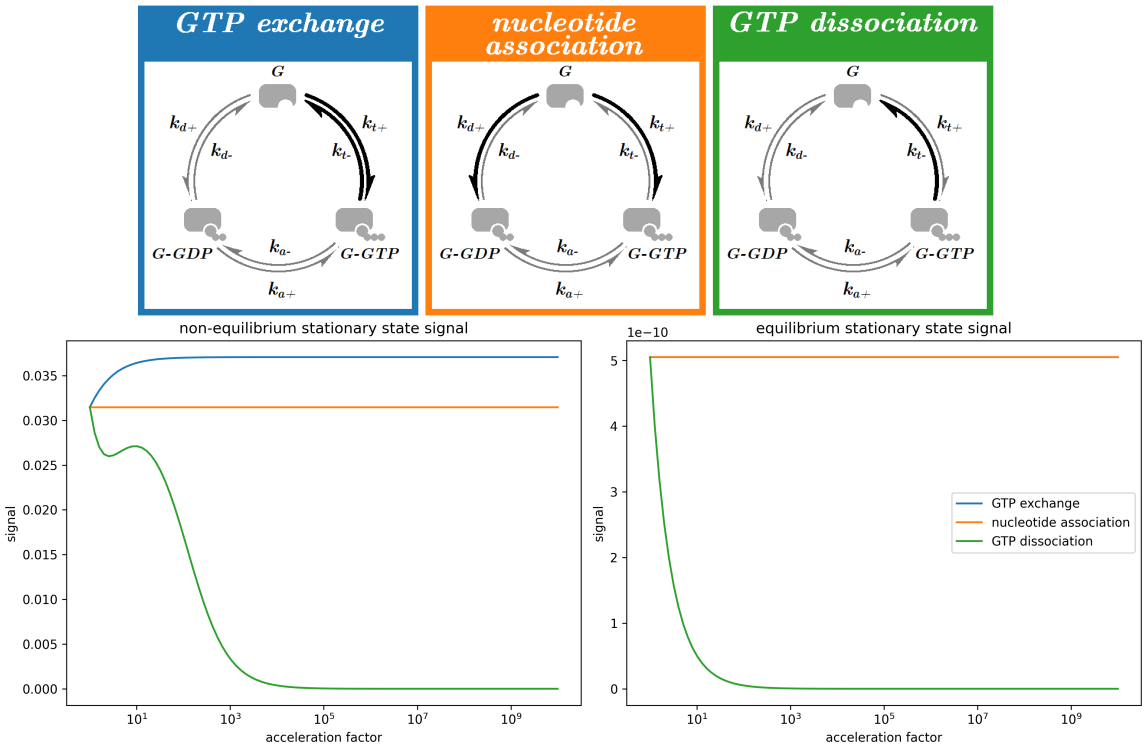


Figure 3.15: **Exploring some regimes for scenarios that are not related to GEF, GAP nor GDI for increasing rate acceleration factors.** In each scenario, some reaction rate constants are accelerated with respect to their standard value by a given factor. The two bottom plots show the signal at steady state as a function of the rate constant(s) acceleration factor. Left: system driven far from equilibrium by chemostating GTP, GDP and Pi (left plot). Right: system that relaxes to equilibrium (no chemostats). Three different rate acceleration scenarios affecting nucleotide association/dissociation are considered (top panels). Those scenarios differ by which rates are accelerated and which are not:
1) **GTP exchange**: association and dissociation rate constants for GTP (k_{t+} and k_{t-}) are accelerated by the same factor. The equilibrium constants are left unchanged.
2) **nucleotide association**: only association rate constants for GTP and GDP (k_{d+} and k_{t+}) are accelerated by the same factor. The equilibrium constants K_d and K_t are increased, but since the scaling factor is the same, the detailed balance constraint is still respected.
3) **GTP dissociation**: only the dissociation rate constant for GTP k_{t-} is accelerated. The equilibrium constant K_t is decreased. The detailed-balance constraint requires to adapt K_a, which is done by shifting both k_{a+} and k_{a-}, as described in appendix A.1.

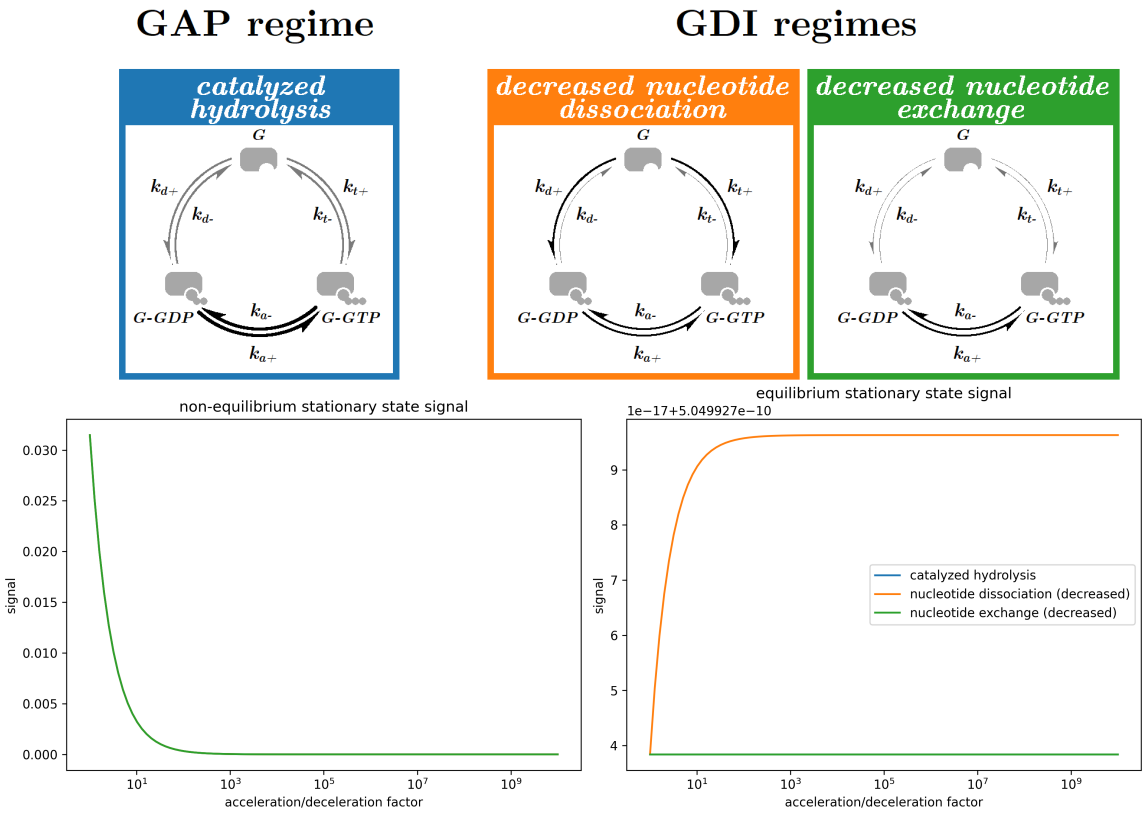


Figure 3.16: **Exploring GAP and GDI regimes, depending on the reaction rate change factor.** For each regime, we consider a rate-change factor f. Top panels (from left to right): Illustration of the GAP regime where the hydrolysis reaction is accelerated by f. Illustration of two GDI regimes, one where only nucleotide dissociation is decreased (divided by f), and the other where both association and dissociation of nucleotides are slowed down (divided by f). The plots show the signal at steady state as a function of the acceleration factor for the three regimes. Left plot: signal at the non-equilibrium steady state, with GTP, GDP and P_i chemostatted. Right plot: signal at the equilibrium steady state, without any chemostat.

ever, when only k_{off} changes for only one of the nucleotides, the K_d/K_t ratio changes, which means K_a must be changed accordingly to respect the constraint. To do so, we have multiple options: change only k_{a+}, change only k_{a-}, or change both to fit the new K_a. In figures 3.14 and 3.15, we only considered the case in which k_{a+} and k_{a-} were both changed to fit K_a. However, as we have observed, for a given equilibrium constant, far from equilibrium, different reaction rate values can have very different effects on the signal at steady state. As a consequence, we decided to look a little further into the "GDP dissociation" regime by considering the different ways to adapt catalyzed hydrolysis rates to match the new value of K_a (figure 3.17). We considered three ways to adapt the constants:

- by adapting k_{a+} as $k_{a+} = K_a k_{a-}$,
- by adapting k_{a-} as $k_{a-} = k_{a+}/K_a$ and
- by adapting both k_{a+} and k_{a-} at the same time, as described in section A.1.

We observed that these different approaches indeed induce different signal values at steady state out of equilibrium. For acceleration factors $> 10^4$, the signal approaches its maximum value ($\simeq 1$), but for lower values, adapting k_{a+} leads to lower signal values than adapting k_{a-} or both rate constants at the same time (the latter giving similar profiles). What is important to understand and remember here is the following: When we consider a GTPase module by varying some kinetic parameters of interest, other rates may also have to be modified due to the detailed balance constraints. Adapting those rates, which may not be the parameters of interest, may be done in different ways, which may have a drastic impact on how the system behaves. Thus, it is a concern that should absolutely not be overlooked. This will be especially important in the next sections, where we will be considering more complex GTPase systems featuring different partners.

3.7.1.4 Effect of the nonequilibrium driving

We have seen that GTPases signaling is bound to a context where GTP, GDP and P_i are chemostatted. We may thus wonder how the concentrations at which those species are chemostatted affect the signal, and more specifically the regulation of GTPases. To investigate this, we measured the signal for the base regime (standard GTPase parameters), the GEF regime (nucleotide dissociation boosted by 10^4) and the GAP regime (hydrolysis boosted by 10^4) as a function of the concentrations of GDP, GTP and P_i. First, we plotted the signal as a function of the proportion of non-dissociated GTP, while conserving the standard quantities of GDP and GTP (figure 3.18 left). The plots revealed that the closest the nucleotide unbalance is to 1, the stronger the signal is for all the regimes alike. The ratio from the literature is already close to 1, which means that the signal is already rather optimized with respect to this aspect. It is interesting to check how the chemostatted density of inorganic phosphate, P_i, impacts the signal. In order to do so, we next plotted the signal as a function of the chemostatted density of P_i in the system, ranging from zero to twice the literature value. We remark that the quantity of P_i in this range has almost no effect on the signal, for all 3 regimes. Even the total absence of P_i does not seem to disturb the system. An absence of inorganic phosphate makes the hydrolysis reaction completely irreversible. However, this is already almost the case, even when P_i is present in the system. This could explain why the change of density of P_i has so little impact on the signal. Overall, this suggests that it is mainly the GTP/GDP balance that affects the activity of GTPases.

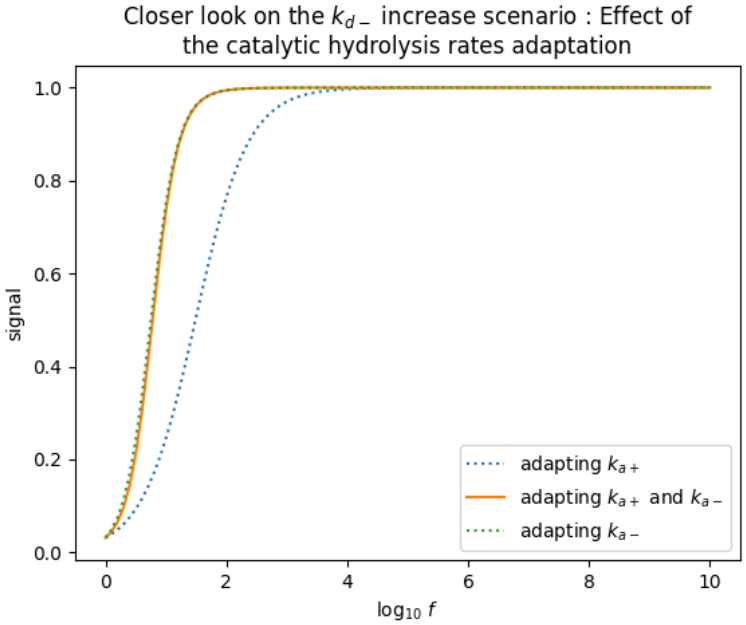


Figure 3.17: **When applying detailed balance constraints, the choice of kinetic rates has a significant impact on the system dynamics.** Here, we consider the GEF regime where only k_{d-} is increased, plotting as in the other figures, the non-equilibrium steady state signal as a function of the increase factor on k_{d-} relative to its standard value. Since changing k_{d-} impacts K_d, the value of K_a must be subsequently adapted due to the DB constraints. Three curves are plotted, each one corresponding to a different way to recalculate K_a: by adapting k_{a+} (dotted blue line), by adapting k_{a-} (dotted green line) or by adapting both k_{a+} and k_{a-} (plain orange line) using the approach described in section A.1.

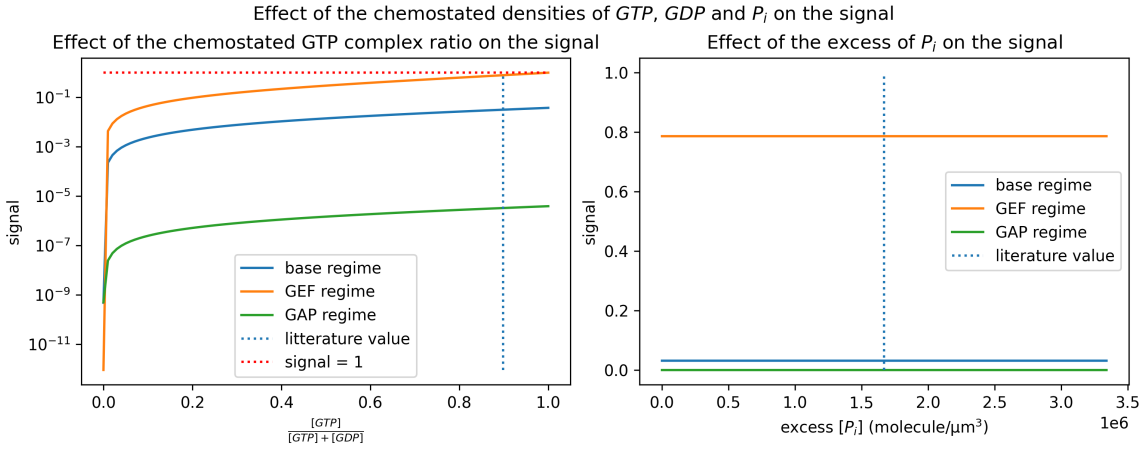


Figure 3.18: **Effect of the non-equilibrium driving on the activity of the base, GAP and GEF regimes.**

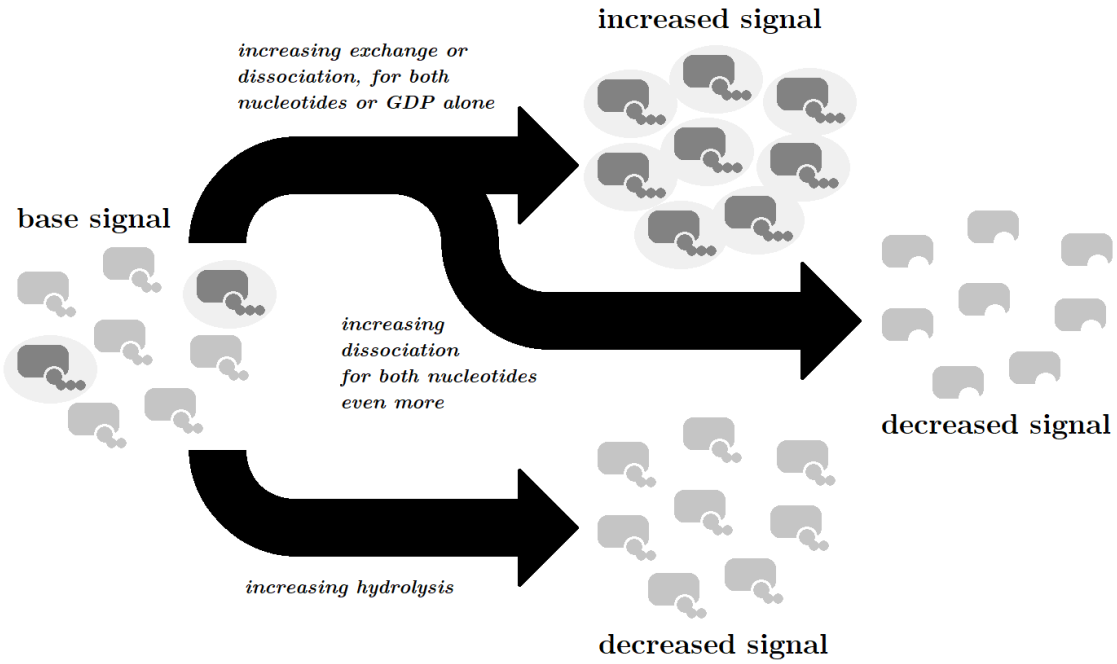


Figure 3.19: **Scheme of the GEF and GAP regimes.**

3.7.1.5 Conclusion

To conclude, all these observations clearly illustrate how the regulation of GTPases by GAP and GEF is a pure **non-equilibium effect**. A previous work by Nguyen et al [41] already highlighted the role of non-equilibrium processes such as infra-affinity in the context of GTPase networks involving GAP and GEFs. Also, we have seen here that studying the elementary GTPase module is enough to understand most of the core properties of GTPase signaling regulation by GEFs and GAPs. We are thus ready to push the investigation further by considering more complex GTPase systems involving partners.

3.7.2 Partner-GTPase module

3.7.2.1 Introduction

In the previous sections, we studied the different regimes associated with the base GTPase network, which gave us a good idea on how its dynamics is regulated. In reality, in a dynamic biological system, changes in GTPase are dealt with by GTPase partners, which change the rates of the GTPase cycle upon association. In this subsection, we investigate the dynamics of a system with a GTPase and one or more partners, as well as the subsequent regimes of the systems.

Now that we explicitly take into account the GTPase partners, it is necessary to consider the potential competition of a given partner with the downstream effector, as evoked earlier in the "Signal transduction" section. In a context of signal transduction, the quantification of the signal

should reflect the quantity of activated downstream effectors. For a system with one generic partner X, we defined two ways of measuring the signal, depending on whether or not X would compete with downstream effectors:

- The **competitive scenario**, corresponding to the case where X compete with potential effectors. In this scenario, in the absence of downstream effectors, only the partner-free GTP-bound GTPases contribute to the signal. This signal is thus measured as

$$\mathcal{S}(in, t) = \frac{[G\text{-}GTP]_t}{G_0} \tag{3.37}$$

  where in is the input signal, whose meaning depends on the system, and G_0 is the conserved total density of GTPases.

- The **non-competitive scenario**, corresponding to the case where the partners and downstream effectors can bind the GTPase at the same time. In this scenario, the partner-bound GTPase shall thus be counted in the signal, which is then, in the absence of downstream effectors, defined as

$$\mathcal{S}(in, t) = \frac{[X\text{-}G\text{-}GTP]_t + [G\text{-}GTP]_t}{G_0} \tag{3.38}$$

In this section, the choice of the scenario we consider will also be a parameter.

3.7.2.2 Generic partner

We will start with the simplest case: That of a generic partner X, associating with all states of the GTPase without preference. In such a system, the fraction of bound partner depends on the affinity between the GTPase and the partner and the total density of the partner as shown in figure 3.20. The base reference parameters used to model the interaction between the GTPase and its partner are based on those from an effector. In such kinetics, it takes a large excess of partners to saturate the GTPase. This is important to consider, as the fraction of partner-bound GTPases is a major factor of the difference between the scenarios. In the next subsubsection, we explore the GAP and GEF regimes, where GAPs and GEFs are treated as generic partners.

3.7.2.3 GAP and GEF regimes

GTPases systems featuring one GEF and one GAP are typical and have been dealt with in a lot of previous works [41] [24] [50]. To bring further the investigation of the GEF and GAP regimes that we started in the last subsection, let us redefine the GEF and GAP regimes, which are this time imposed by partners (illustrated in figure 3.21), as follows:

- **GEF regime**: acceleration of nucleotide dissociation rates k_{dx-} and k_{tx-} in the GEF-bound form by some acceleration factor f,
- **GAP regime**: acceleration of catalyzed hydrolysis rates k_{ax+} and k_{ax-} in the GAP-bound form by some acceleration factor f.

These regimes only affect the kinetic parameters of the partner-bound GTPase. This means that in the absence of a partner, the system is at its base regime, with the corresponding base signal. In the same spirit as our earlier study of the elementary GTPase module, we investigated how

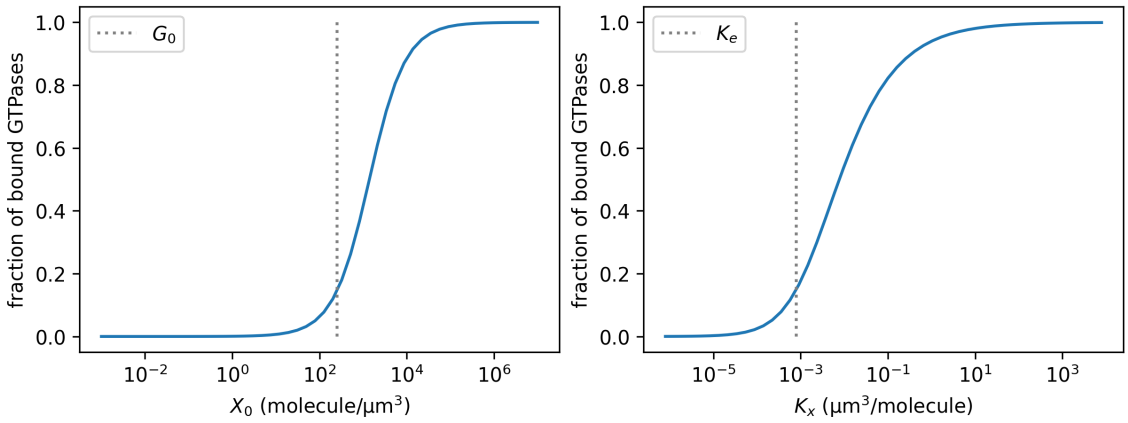


Figure 3.20: **Fraction of partner-bound GTPase as a function of the partner density and affinity** Fraction of GTPase bound to the partner X as a function of the total partner density (left) and of the binding equilibrium constant K_x (right). Here, $K_x = K_{x0} = K_{xt} = K_{xd}$. When X_0, the total density of partner, varies, $K_x = K_e$, while when K_x varies, $X_0 = G_0$.

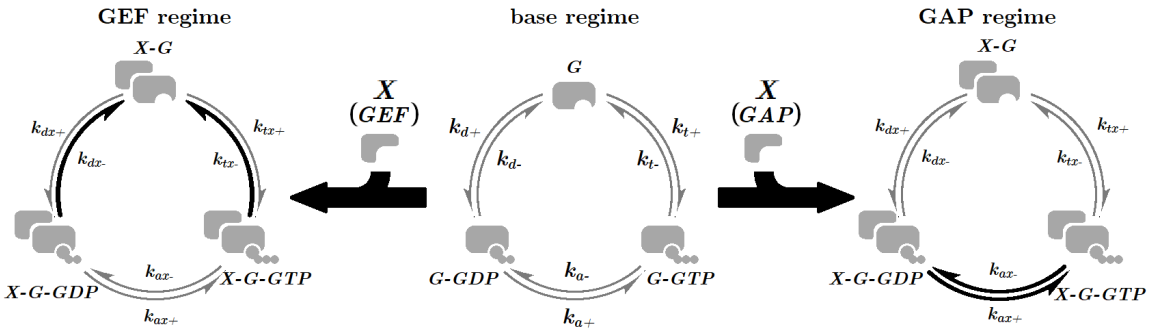


Figure 3.21: **Illustration of the GEF and GAP regimes in the partner-GTPase reaction network.** In the GEF regime, nucleotide dissociation rates k_{dx+} and k_{tx-} are accelerated by a factor f for the partner-bound form. In the GAP regime, catalyzed hydrolysis rates k_{ax+} and k_{ax-} are accelerated by a factor f for the partner-bound form.

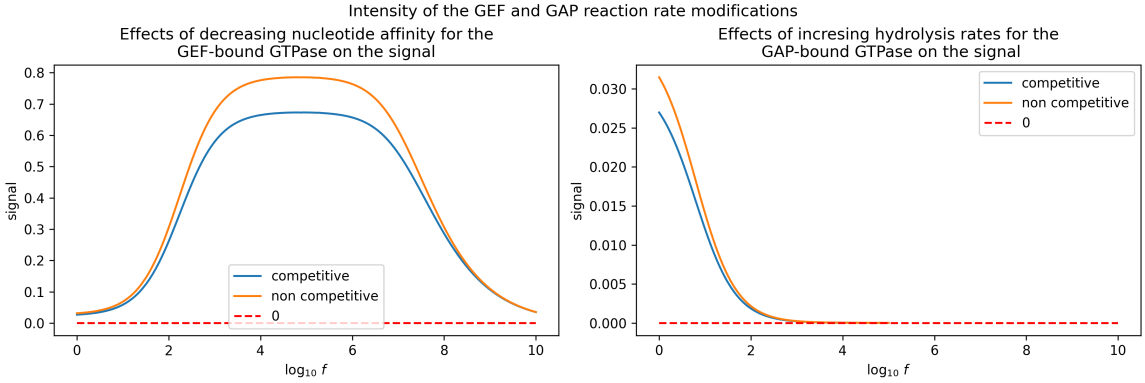


Figure 3.22: **Signal as a function of the acceleration factor gauging the strengths of the GEF (left) and GAP(right)behaviors imposed by the partner.** In the GEF regime, f accelerates GTP and GDP's dissociation, while in the GAP regime, it accelerates globally the catalyzed hydrolysis.

the signal at steady state evolved as a function of the acceleration factor in the GEF and GAP regimes (figure 3.22). Results are in good accordance with those observed for the G module with the "nucleotide dissociation" and "catalyzed hydrolysis" regimes: The more the GEF accelerates nucleotide dissociation, the faster the signal is, until a point where the affinity becomes too low, past which the signal decreases. Analogously, the more the GAP accelerates the hydrolysis, the lower the signal is. We observe little differences between the signals evaluated with the competitive and non-competitive scenarios. As one would expect, the signal is slightly higher in the non-competitive scenario, since there are more species to consider.

We next evaluated the steady state signal, for both the GAP and GEF regimes, also as a function of the quantity of partner (figure 3.23). This time, we observe a major difference between the two scenarii: In the non-competitive scenario, the signal reaches a steady value depending on the level of GEF/GAP activity. Past an optimal density of GEF, when the GEF activity is too strong, the signal starts decreasing. For lower GEF activities, the regulating effect is scaled by the density of partners, as partner-saturation means fully imposing its regime to the system. However, in the competitive scenario, the quantity of partner has an inhibitory effect on the signal, as it decreases the quantity of partner-free GTPases. Consequently, as the quantity of partner increases, the signal will always eventually drop to zero. Since GTPases have a rather low affinity for the partner, it takes a large density ($\gtrsim 10^5$ molecules/μm^3) of partners to saturate them, which is a few orders of magnitude larger than the total density of GTPase G_0 (figure 3.20).

Our results corroborate the observation made by Stanley et al. [50] who pointed out the inhibitory action of GEFs, as their excess or excessive activity may prevent the nucleotides from staying in the active site. However, here, we bring some additional nuances, as it seems that with our model, GEF excess does not decrease the signal in all circumstances: as the concentration of GEF increases, GTPases get saturated, until every single GTPase is bound to a GEF. Of course, in the competitive scenario, this means that there is no more signal, not specifically because of the GEF activity, but simply as there is no more free GTPase. However, in the non-competitive scenario, all GTPases still work under the GEF regime. Then, as long as the GEF activity is low enough to let the nucleotides associate with the GTPases, GEFs will still have an activating effect.

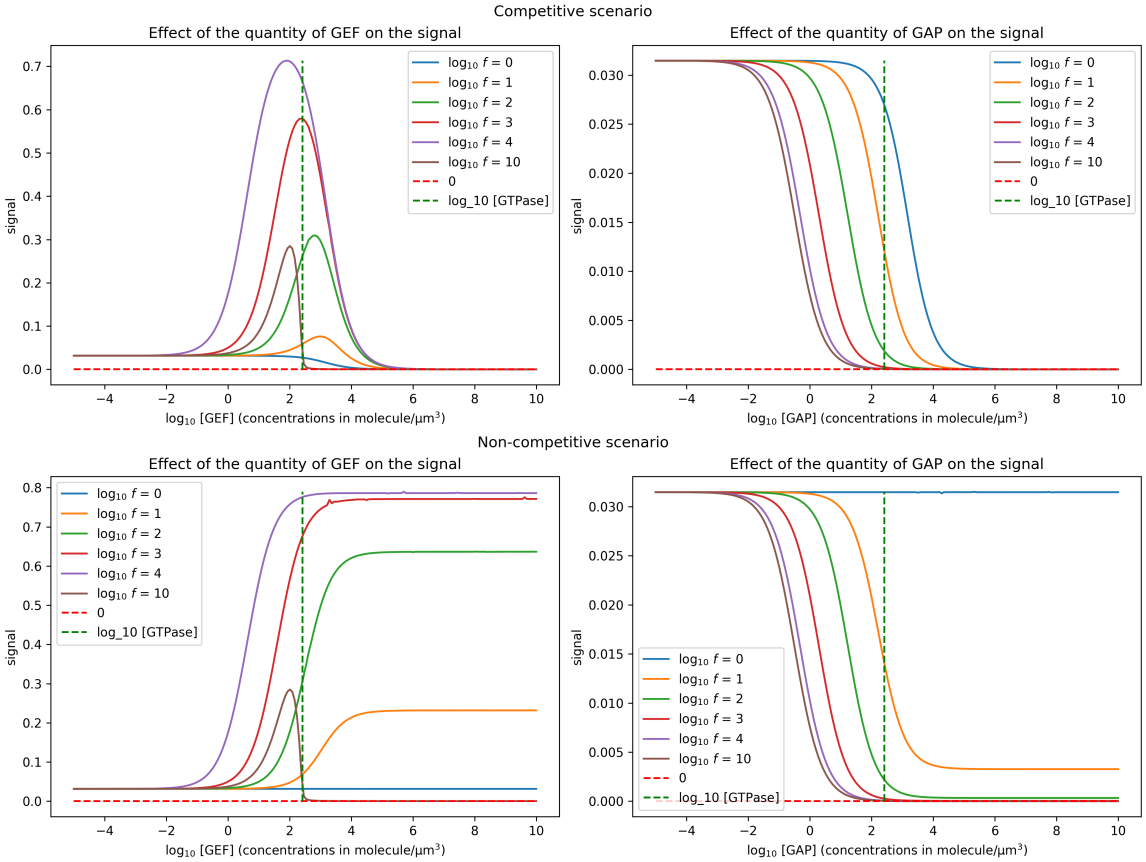


Figure 3.23: **Signal as a function of the quantity of GEF or GAP partners.** In the GEF regime, f is the acceleration factor on the nucleotide dissociation rates in the GEF-GTPase complex. In the GAP regime, f is the acceleration factor on the forward and reverse hydrolysis rates in the GAP-GTPase complex. This figure shows plots of the steady state signal as a function of the density of partner (GEF (left) or GAP (right)) in either the competitive (top) or non-competitive (bottom) scenarios, for different values of f.

3.7.2.4 Generic partner with a partial state preference

We have just examined the influence of a generic partner X with no preference for any state of the GTPase. However, some partners do have a different affinity depending on the state of the GTPase. The DB constraint imposes that

$$\frac{K_{xt}}{K_{xd}} = \frac{K_{tx}}{K_{dx}} \frac{K_d}{K_t} \tag{3.39}$$

As we are considering a partner that brings the GTPase out of its base regime, K_d and K_t are kept at their reference value. Since the K_d/K_t ratio is fixed, the ratio of the association constants of the partner with *G-GTP* and *G-GDP*, K_{xt}/K_{xd}, and the ratio of the association constants of GTP and GDP with the partner-bound GTPase K_{tx}/K_{dx} must be correlated. Increasing the affinity of the partner for the GTP-bound state means also increasing the affinity of GTP for the partner-bound GTPase. On the contrary, decreasing the affinity of the partner for the GTP-bound state means also decreasing the affinity of GTP for the partner-bound GTPase. The DB constraint from the Partner-bound GTPase cycle imposes that

$$K_{ax} = \frac{K_h K_{tx}}{K_{dx}} \tag{3.40}$$

Thus, changing the K_{xt}/K_{xd} ratio also impacts the catalyzed hydrolysis equilibrium. Increasing/decreasing the affinity of the partner with the GTP state means increasing/decreasing the catalyzed hydrolysis dissociation equilibrium constant.

3.7.2.5 Strict partner specificity in a context of signal transduction

Let us consider a GTPase system with two partner species: A GEF and a downstream effector. The GEF and the effector are competitive partners, which means they cannot bind the GTPase at the same time. We consider two ways to model such a system: In the first, the effector is a generic partner that can bind to any state of the GTPase (although not necessarily with the same affinity), while in the second one, the partner is a strict GTP-bound state-specific partner. In both cases, the GEF is a generic partner, with the same affinity for all states of the GTPase. This is a full signal transduction module, as it has an input signal, the density of GEF, and an output signal, the density of activated effectors.

Based on these two models, we considered 3 different situations regarding the interaction between the GTPase and the effector:

1. Using the first model, the effector has the same affinity for *G-GTP* and for *G-GDP*.
2. Still using the first model, the effector has an affinity for *G-GDP* decreased by an arbitrary factor of 10^{-2} compared to its affinity for *G-GTP*.
3. Using the second model, but letting the effector only associate with *G-GTP*.

For each situation, we calculated the non-equilibrium steady state signal as a function of the density of GEF ($[GEF]$) in the system (figure 3.24). The signal was measured in 3 different ways, and each plot features the three situations with a specific way of measuring the signal. In the first plot (3.24, left), the signal is measured as the sum of available active GTPases

$$\mathcal{S}_1(in, t) = \frac{[\text{G-GTP}]_t + [\text{E-G-GTP}]_t}{G_0} \tag{3.41}$$

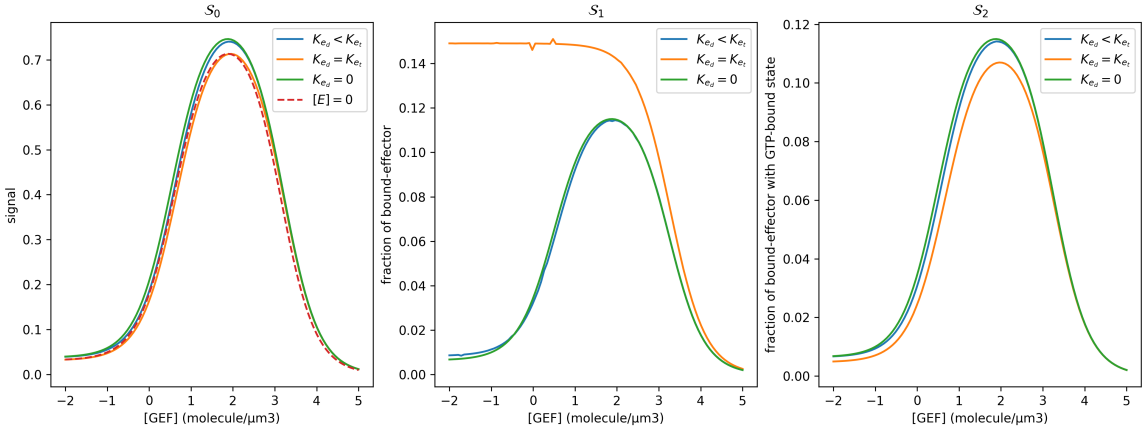


Figure 3.24: **Signal transduction function with different effector state-specificities and signal quantification methods.** Signal as a function of the concentration of GEF in a module consisting of a GTPase with two partners: a GEF and an effector, competing with each other. The GEF is not state-specific and increases the nucleotide dissociation rates by a factor of 10^4. Concerning the Effector specificity, we consider 3 situations:
1. A state-specific effector preferring *G-GTP*. Its affinity for *G-GDP* is decreased by a factor 10^{-2} ($K_{e_d} = 10^{-2}K_{e_t}$).
2. A state-independent effector ($K_{e_d} = K_{e_t}$).
3. A state-specific effector that only associates with *G-GTP* ($K_{e_d} = 0$).
The effector (when present) has an initial density of G_0.

where *in*, the input signal, corresponds to [GEF] and G_0 the (conserved) total density of GTPases. In the second plot (figure 3.24, middle), it is measured as the fraction of effector bound to a GTPase, regardless of its state

$$\mathcal{S}_2(in, t) = \frac{G_0 - [E]_t}{G_0} \tag{3.42}$$

In the third plot (figure 3.24, right), the signal is defined as the fraction of effector bound to *G-GTP*

$$\mathcal{S}_3(in, t) = \frac{[E\text{-}G\text{-}GTP]_t}{G_0} \tag{3.43}$$

In all situations, no matter how the signal is measured, as the GEF concentration increases, the signal first increases and then decreases due to the GTPase being saturated by the GEF (as we are in the competitive scenario). This is a phenomenon we already discussed previously. With the usual way to measure the signal (figure 3.24, left), there is little difference between the 3 situations. This is due to the weak affinity of the effector for the GTPase, even in the GTP state. The presence of the effector does not change the signal much. We can say the same thing when the signal is measured as the proportion of *E-G-GTP* (figure 3.24, right). However, when the signal is measured as the global fraction of bound effectors (middle plot), the situation where the effector has the same affinity for *G-GTP* and *G-GDP* stands out: The signal is already maximum at [GEF]= 0, and decreases as [GEF] increases. This is to be expected, as in this situation, the fraction of bound effectors does not depend of the GTPase state. This gives us insight on how and when the density of available GTP-bound GTPases can be used as a proxy for the output signal.

In the case where only *G-GTP* can biologically activate effectors (*G-GDP* association with the effector would not lead to any change in its activity), or where the effector has a strong preference for GTP in terms of affinity, this seems to be a good choice. However, if 1) the effector has no preference, and 2) both *G-GTP* and *G-GDP* activate the effector upon association, then using the density of available *G-GTP* as a proxy for the signal may be misleading. Also, we see that, even if the fraction of available active G protein reaches a peak value above 0.7, the actual proportion of effector bound to *G-GTP* only reaches 0.07.

The main conclusion, here, is that the simplified network where state-specific partners bind only to their preferred state is a good approximation to the complete network in the case of the effectors, and that using the density of available *G-GTP* is a good proxy for the signal (case where the effector prefers *G-GTP*).

3.7.2.6 Conclusion

GTPase partners bring regime changes to the GTPase cycle. We studied the GAP and GEF partners regimes considering two possible scenarios : Either the partner was considered to compete with downstream effectors, in which case only partner-free *G-GTP* was counted in the signal, or the partner was considered not to compete with downstream effectors, and then both *X-G-GTP* and *G-GTP* were counted for the signal. The regimes are very similar to their elementary equivalents, aside from the inhibiting effect caused by the GTPase saturation with partners in the competitive scenario. Next, we considered the state specificity. State specificity may be partial, in which case the asymmetry in the partner association kinetics reflects on the nucleotide association/dissociation kinetics of the complex. Otherwise, the state-specificity is strict and does not bring any thermodynamic constraints. The GTPase effectors are typically state-specific partners, having a preference for the GTP-state [44]. We compared partial and strict state-specific models for the effector and showed that the former provide a good approximation of the latter.

3.7.3 GPCR signal transduction model

3.7.3.1 Introduction

In this subsection, we will study the full GPCR signal transduction system. In this system, the GTPase is the G_α subunit of the G protein. It is regulated by the $G_{\beta\gamma}$ subunit and by the active GPCR, which acquires the GEF function upon ligand binding [11]. In this transduction system, the input signal is the total density of agonist ligand, L_0. One important particularity of this system is that it has two different, independent output signals, associated respectively with each subunit of the G protein, G_α and $G_{\beta\gamma}$. Each of the two subunits has its own distinct downstream effector species, termed E and E'. Since E and E' are explicitly present in the system, their quantities are used to define the two signals:

$$\begin{cases} \mathcal{S}_\alpha(in,t) = \dfrac{[E\text{-}G_\alpha\text{-}GTP]_t}{G_0} \\ \mathcal{S}_{\beta\gamma}(in,t) = \dfrac{[E'\text{-}G_{\beta\gamma}]_t}{G_0} \end{cases} \tag{3.44}$$

where *in* is the input signal, which corresponds to the total density of ligand L_0, and G_0 is the total density of G proteins.

Parameter	Biophysical effect
ϕ	Strength of the GEF effect of the activated GPCR.
δ	Preference of $G_{\beta\gamma}$ for the G_α-GDP over G_α-GTP
γ	Affinity between $G_{\beta\gamma}$ and G_α-GDP relative to the effector affinity K_e

Table 3.1: **The 3 free parameters regulating the kinetics of the GPCR signal transduction model and their effects.**

The dynamics of the GPCR model is governed by 98 kinetic rates, or 96 if we put aside the uncatalyzed hydrolysis rates. Such a large number of parameters makes investigating the regimes of the GPCR module quite difficult. Combining the thermodynamic constraints coming from DB and reasonable guesses based on the literature, it is possible to isolate three free parameters that regulate the dynamics of our model, which we denote with ϕ, δ and γ. Physically, each parameter gauges a different effect (see table 3.1).

The GEF function of the activated receptor responsible for the activation of G_α [11] requires the presence of $G_{\beta\gamma}$ within the complex [39], and corresponds to an increase in the dissociation of the nucleotides [58]. This effect is captured by the parameter ϕ, which measures how much the dissociation of GTP and GDP for G_α is increased when the trimeric G protein is bound to an active GPCR. It is defined as

$$\phi = \frac{K_{du}}{K_{dcu}} = \frac{K_{dru}}{K_{dcu}} = \frac{K_{tcu}}{K_t} \tag{3.45}$$

where K_{du}, K_{dru} and K_{dcu} are the association equilibrium constants of GDP with the trimeric G protein, alone, bound to an inactive receptor and bound to an active receptor, respectively, while K_t and K_{dcu} are the association equilibrium constants of GTP with the G_α alone and with the trimeric G protein bound to an active receptor, respectively. The parameter ϕ is an important parameter in our model as it measures the strengths of the GEF effect. Consequently, we term it the **GEF strength**. For the sake of simplicity, we assume that the association of a GPCR with a G protein does not depend on whether the latter is in the GTP- or GDP-bound state, this is why ϕ impacts both nucleotide's dissociation rate equally.

The trimeric G protein has a higher affinity for GDP than free G_α [6] which is consistent with the preference of $G_{\beta\gamma}$ for G_α-GDP over G_α-GTP, given the DB constraints. This effect is captured by the parameter δ, defined as

$$\delta = \frac{K_{du}}{K_d} = \frac{K_{dru}}{K_d} = \frac{K_{dcu}\phi}{K_d} > 1 \tag{3.46}$$

where the label du, dru and dcu correspond respectively to the reactions G_α-$GDP + G_{\beta\gamma} \rightleftharpoons G_\alpha$-$GDP$-$G_{\beta\gamma}$, R-G_α-$GDP + G_{\beta\gamma} \rightleftharpoons R$-$G_\alpha$-$GDP$-$G_{\beta\gamma}$ and C-G_α-$GDP + G_{\beta\gamma} \rightleftharpoons C$-$G_\alpha$-$GDP$-$G_{\beta\gamma}$. The parameter δ gauges $G_{\beta\gamma}$'s preference for the GDP-bound state of G_α over its GTP-bound state. The association of $G_{\beta\gamma}$ impacts both GDP's k_{on} and k_{off} [6]. For the association of GDP with G_α-GTP-$G_{\beta\gamma}$, we take

$$K_{du} = \frac{k_{d+}}{k_{d-}} f^2 \tag{3.47}$$

where f is a factor that gauges the modification of k_{d+} and k_{d-} upon association of $G_{\beta\gamma}$ with G_α-GDP. The factor f can be derived from equations 3.46 and 3.47

$$f = \sqrt{\delta} \tag{3.48}$$

and shall be applied to all the GDP association reactions involving a $G_{\beta\gamma}$ subunit as part of the complex (reactions *du*, *dru* and *dcu*), as a global acceleration factor. The factor f is a function of δ which is why it is not considered a parameter per se. Furthermore, we assume that the binding of $G_{\beta\gamma}$ does not alter GTP's binding dynamics. As a consequence, we take

$$k_{tu+} = k_{t+} \quad \text{and} \quad k_{tu-} = k_{t-} \tag{3.49}$$

The third parameter γ, measures the cohesion of the trimeric G protein. The higher it is, the stronger the G_α-*GDP* $G_{\beta\gamma}$ complex is. It is defined as

$$\gamma = \frac{K_{ud}}{K_e} = 0.1\frac{K_{urd}}{K_e} = 0.1\frac{K_{ucd}}{K_e} \tag{3.50}$$

The parameter γ is proportional to the affinity between $G_{\beta\gamma}$ and the GDP-bound states of G_α, no matter their partner. We define it relative to K_e, the affinity between G_α-*GTP* or $G_{\beta\gamma}$ and their downstream effectors. It appears reasonable to consider $\gamma > 1$ since K_e is rather low.

It also appears reasonable to assume that the association with the GPRC increases the affinity between G_α-*GDP* and $G_{\beta\gamma}$ of a factor $\mathcal{O}(10)$. As a consequence, in the following we take

$$K_{ucd} = K_{urd} = 10\gamma K_e \tag{3.51}$$

The full set of stoichiometric equations for the reaction network, as well as the description of how the kinetic rate constants are defined as a function of the parameters ϕ, δ and γ are given in table 3.2.

3.7.3.2 The model has 3 optimal signaling regimes

An efficient signal transduction module can be characterized by its ability to display an ample signal variation upon activation. For the present model, since there are two different parallel output signals, $\mathcal{S}_\alpha$ and $\mathcal{S}_{\beta\gamma}$, one may define three different kinetic parameter regimes depending on which signal variation is maximized: the α regime for $\Delta\mathcal{S}_\alpha$, the β regime for $\Delta\mathcal{S}_{\beta\gamma}$, and the $\alpha\beta$ regime when both $\Delta\mathcal{S}_\alpha$ and $\Delta\mathcal{S}_{\beta\gamma}$ are maximized. In order to identify the parameters associated with each of those three regimes, a stochastic maximization of $\Delta\mathcal{S}_\alpha(\infty,\infty)$ and/or $\Delta\mathcal{S}_{\beta\gamma}(\infty,\infty)$ as a function of ϕ, δ and γ was performed (figure 3.25). The three regimes are well separated in the parameter space, and thus distinct from each other. As expected, the α regime provides the highest $\Delta\mathcal{S}_\alpha$, while the β regime provides the highest $\Delta\mathcal{S}_{\beta\gamma}$, and the $\alpha\beta$ one provides a good compromise. However, it appears that the α regime exhibits a $\Delta\mathcal{S}_{\beta\gamma}$ that is considerably lower than for the two other regimes. Overall, an important observation here is that these three regimes do not overlap in parameter space. Thus, depending on its parameters, the model may favor either G_α or $G_{\beta\gamma}$ signaling, at the expense of the other.

Table 3.3 gives the optimal parameters and the corresponding signal values for each of the three regimes. For the different regimes, the GEF strength is high, as ϕ at maximum signal ranges between $\simeq 10^6$ and $\simeq 10^{10}$. This means that in all cases, the active GPCR greatly increases the dissociation of *GTP* and *GDP* from the trimeric G protein. However, one can notice that the GEF strength is still separated by several orders of magnitude from one regime to the other, with the highest value of ϕ corresponding to the α regime and the lowest to the $\alpha\beta$ regime. This is consistent

reaction	k_+	k_-	label
$G_\alpha + GTP \rightleftharpoons G_\alpha\text{-}GTP$	k_{t+}	k_{t-}	t
$G_\alpha + GDP \rightleftharpoons G_\alpha\text{-}GDP$	k_{d+}	k_{d-}	d
$G_\alpha\text{-}GDP + P_i \rightleftharpoons G_\alpha\text{-}GTP$	k_{a+}	k_{a-}	a
$GDP + P_i \rightleftharpoons GTP$	K_h	1	h
$R\text{-}G_\alpha + GTP \rightleftharpoons R\text{-}G_\alpha\text{-}GTP$	k_{t+}	k_{t+}/K_t	tr
$R\text{-}G_\alpha + GDP \rightleftharpoons R\text{-}G_\alpha\text{-}GDP$	k_{d+}	k_{d+}/K_d	dr
$R\text{-}G_\alpha\text{-}GDP + P_i \rightleftharpoons R\text{-}G_\alpha\text{-}GTP$	k_{a+}	k_{a+}/K_{ar}	ar
$C\text{-}G_\alpha + GTP \rightleftharpoons C\text{-}G_\alpha\text{-}GTP$	k_{t+}	k_{t+}/K_t	tc
$C\text{-}G_\alpha + GDP \rightleftharpoons C\text{-}G_\alpha\text{-}GDP$	k_{d+}	k_{d+}/K_d	dc
$C\text{-}G_\alpha\text{-}GDP + P_i \rightleftharpoons C\text{-}G_\alpha\text{-}GTP$	k_{a+}	k_{a+}/K_{ac}	ac
$G_\alpha\text{-}G_{\beta\gamma} + GTP \rightleftharpoons G_\alpha\text{-}GTP\text{-}G_{\beta\gamma}$	k_{t+}	k_{t+}/K_t	tu
$G_\alpha\text{-}G_{\beta\gamma} + GDP \rightleftharpoons G_\alpha\text{-}GDP\text{-}G_{\beta\gamma}$	$f\ k_{d+}$	$f\ \delta^{-1}\ k_{d+}/K_d$	du
$G_\alpha\text{-}GDP\text{-}G_{\beta\gamma} + P_i \rightleftharpoons G_\alpha\text{-}GTP\text{-}G_{\beta\gamma}$	k_{a+}	k_{a+}/K_{au}	au
$R\text{-}G_\alpha\text{-}G_{\beta\gamma} + GTP \rightleftharpoons R\text{-}G_\alpha\text{-}GTP\text{-}G_{\beta\gamma}$	k_{t+}	k_{t+}/K_t	tru
$R\text{-}G_\alpha\text{-}G_{\beta\gamma} + GDP \rightleftharpoons R\text{-}G_\alpha\text{-}GDP\text{-}G_{\beta\gamma}$	$f\ k_{d+}$	$f\ \delta^{-1}\ k_{d+}/K_d$	dru
$R\text{-}G_\alpha\text{-}GDP\text{-}G_{\beta\gamma} + P_i \rightleftharpoons R\text{-}G_\alpha\text{-}GTP\text{-}G_{\beta\gamma}$	k_{a+}	k_{a+}/K_{aru}	aru
$C\text{-}G_\alpha\text{-}G_{\beta\gamma} + GTP \rightleftharpoons C\text{-}G_\alpha\text{-}GTP\text{-}G_{\beta\gamma}$	k_{t+}	$\phi\ k_{t+}/K_t$	tcu
$C\text{-}G_\alpha\text{-}G_{\beta\gamma} + GDP \rightleftharpoons C\text{-}G_\alpha\text{-}GDP\text{-}G_{\beta\gamma}$	$f\ k_{d+}$	$f\ \phi\ \delta^{-1}\ k_{d+}/K_d$	dcu
$C\text{-}G_\alpha\text{-}GDP\text{-}G_{\beta\gamma} + P_i \rightleftharpoons C\text{-}G_\alpha\text{-}GTP\text{-}G_{\beta\gamma}$	k_{a+}	k_{a+}/K_{acu}	acu
$R + G_\alpha \rightleftharpoons R\text{-}G_\alpha$	k_{e+}	k_{e+}/K_{r0}	r0
$R + G_\alpha\text{-}GDP \rightleftharpoons R\text{-}G_\alpha\text{-}GDP$	k_{e+}	k_{e+}/K_e	rd
$R + G_\alpha\text{-}GTP \rightleftharpoons R\text{-}G_\alpha\text{-}GTP$	k_{e+}	k_{e+}/K_{rt}	rt
$C + G_\alpha \rightleftharpoons C\text{-}G_\alpha$	k_{e+}	k_{e+}/K_{c0}	c0
$C + G_\alpha\text{-}GDP \rightleftharpoons C\text{-}G_\alpha\text{-}GDP$	k_{e+}	k_{e+}/K_e	cd
$C + G_\alpha\text{-}GTP \rightleftharpoons C\text{-}G_\alpha\text{-}GTP$	k_{e+}	k_{e+}/K_{ct}	ct
$G_{\beta\gamma} + G_\alpha \rightleftharpoons G_\alpha\text{-}G_{\beta\gamma}$	k_{e+}	k_{e+}/K_{u0}	u0
$G_{\beta\gamma} + G_\alpha\text{-}GDP \rightleftharpoons G_\alpha\text{-}GDP\text{-}G_{\beta\gamma}$	k_{e+}	$\gamma^{-1}\ k_{e+}/K_e$	ud
$G_{\beta\gamma} + G_\alpha\text{-}GTP \rightleftharpoons G_\alpha\text{-}GTP\text{-}G_{\beta\gamma}$	k_{e+}	k_{e+}/K_{ut}	ut
$R + G_\alpha\text{-}G_{\beta\gamma} \rightleftharpoons R\text{-}G_\alpha\text{-}G_{\beta\gamma}$	k_{e+}	k_{e+}/K_{ru0}	ru0
$R + G_\alpha\text{-}GDP\text{-}G_{\beta\gamma} \rightleftharpoons R\text{-}G_\alpha\text{-}GDP\text{-}G_{\beta\gamma}$	k_{e+}	k_{e+}/K_{rud}	rud
$R + G_\alpha\text{-}GTP\text{-}G_{\beta\gamma} \rightleftharpoons R\text{-}G_\alpha\text{-}GTP\text{-}G_{\beta\gamma}$	k_{e+}	k_{e+}/K_{rut}	rut
$C + G_\alpha\text{-}G_{\beta\gamma} \rightleftharpoons C\text{-}G_\alpha\text{-}G_{\beta\gamma}$	k_{e+}	k_{e+}/K_{cu0}	cu0
$C + G_\alpha\text{-}GDP\text{-}G_{\beta\gamma} \rightleftharpoons C\text{-}G_\alpha\text{-}GDP\text{-}G_{\beta\gamma}$	k_{e+}	k_{e+}/K_{cud}	cud
$C + G_\alpha\text{-}GTP\text{-}G_{\beta\gamma} \rightleftharpoons C\text{-}G_\alpha\text{-}GTP\text{-}G_{\beta\gamma}$	k_{e+}	k_{e+}/K_{cut}	cut
$G_{\beta\gamma} + R\text{-}G_\alpha \rightleftharpoons R\text{-}G_\alpha\text{-}G_{\beta\gamma}$	k_{e+}	k_{e+}/K_{ur0}	ur0
$G_{\beta\gamma} + R\text{-}G_\alpha\text{-}GDP \rightleftharpoons R\text{-}G_\alpha\text{-}GDP\text{-}G_{\beta\gamma}$	k_{e+}	$0.1\ \gamma^{-1}\ k_{e+}/K_e$	urd
$G_{\beta\gamma} + R\text{-}G_\alpha\text{-}GTP \rightleftharpoons R\text{-}G_\alpha\text{-}GTP\text{-}G_{\beta\gamma}$	k_{e+}	k_{e+}/K_{urt}	urt
$G_{\beta\gamma} + C\text{-}G_\alpha \rightleftharpoons C\text{-}G_\alpha\text{-}G_{\beta\gamma}$	k_{e+}	k_{e+}/K_{uc0}	uc0
$G_{\beta\gamma} + C\text{-}G_\alpha\text{-}GDP \rightleftharpoons C\text{-}G_\alpha\text{-}GDP\text{-}G_{\beta\gamma}$	k_{e+}	$0.1\ \gamma^{-1}\ k_{e+}/K_e$	ucd
$G_{\beta\gamma} + C\text{-}G_\alpha\text{-}GTP \rightleftharpoons C\text{-}G_\alpha\text{-}GTP\text{-}G_{\beta\gamma}$	k_{e+}	k_{e+}/K_{uct}	uct
$R + L \rightleftharpoons C$	k_{l+}	k_{l-}	l
$R\text{-}G_\alpha + L \rightleftharpoons C\text{-}G_\alpha$	k_{l+}	k_{l+}/K_{l0}	l0
$R\text{-}G_\alpha\text{-}GDP + L \rightleftharpoons C\text{-}G_\alpha\text{-}GDP$	k_{l+}	k_{l+}/K_{ld}	ld
$R\text{-}G_\alpha\text{-}GTP + L \rightleftharpoons C\text{-}G_\alpha\text{-}GTP$	k_{l+}	k_{l+}/K_{lt}	lt
$R\text{-}G_\alpha\text{-}G_{\beta\gamma} + L \rightleftharpoons C\text{-}G_\alpha\text{-}G_{\beta\gamma}$	k_{l+}	k_{l+}/K_{lu0}	lu0
$R\text{-}G_\alpha\text{-}GDP\text{-}G_{\beta\gamma} + L \rightleftharpoons C\text{-}G_\alpha\text{-}GDP\text{-}G_{\beta\gamma}$	k_{l+}	k_{l+}/K_{lud}	lud
$R\text{-}G_\alpha\text{-}GTP\text{-}G_{\beta\gamma} + L \rightleftharpoons C\text{-}G_\alpha\text{-}GTP\text{-}G_{\beta\gamma}$	k_{l+}	k_{l+}/K_{lut}	lut
$E + G_\alpha\text{-}GTP \rightleftharpoons E\text{-}G_\alpha\text{-}GTP$ 104	k_{e+}	k_{e-}	e
$G_{\beta\gamma} + E' \rightleftharpoons E'\text{-}G_{\beta\gamma}$	k_{e+}	k_{e-}	ep

Table 3.2: **Kinetic parameters of the GPCR signal transduction model.**

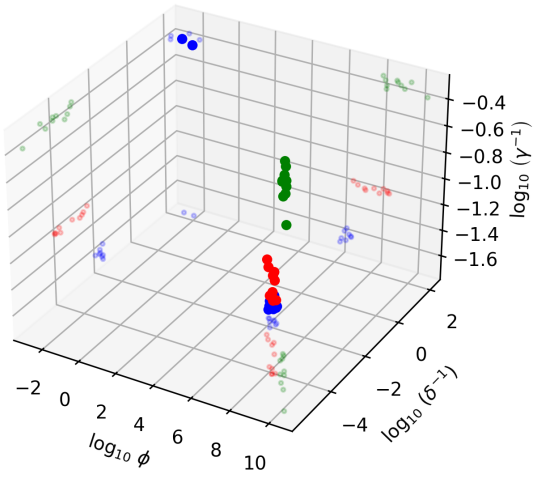


Figure 3.25: **Maximization of the GPCR signals in parameter space.** Each point is the result of an independent stochastic maximization (green: α regime, blue: β regime, red: $\alpha\beta$ regime). The values of the optimized parameters corresponding to maximum signals and the associated signal values are reported in table 3.3.

regime	$\log_{10}\,\phi$	$\log_{10}\left(\delta^{-1}\right)$	$\log_{10}\left(\gamma^{-1}\right)$	$\Delta\mathcal{S}_{\alpha}(\infty,\infty,\phi,\delta,\gamma)$	$\Delta\mathcal{S}_{\beta\gamma}(\infty,\infty,\phi,\delta,\gamma)$
α	10.25	-5.59	-0.40	0.53	0.09
β	5.92	-1.80	-1.66	0.43	0.23
$\alpha\beta$	8.44	-4.15	-1.20	0.49	0.21

Table 3.3: **Optimal parameter and signal values for each GPCR regime.** In figure 3.25, we realize multiple independent stochastic maximizations of the GPCR signal for each of the 3 different regimes. This table reports the parameters and signals associated with the best maximization in each regime.

with the fact that the activation of G_α, which determines the α component of the regimes, depends directly on the strengths of the GEF.

The preference of $G_{\beta\gamma}$ for the GDP state of G_α is present in all regimes ($\delta > 1$) but at different levels. In the α and $\alpha\beta$ regimes, there is a strong preference of $G_{\beta\gamma}$ for the GDP state of G_α with $\log_{10}\delta \simeq 5.8$ and $\log_{10}\delta \simeq 4$ respectively. However, in the β regime, there is only a slight preference of $G_{\beta\gamma}$ for the GDP state of G_α, as it displays a value of $\log_{10}\delta \simeq 1.8$. A strong preference for G_α-GDP may participate in stabilizing the trimeric G protein in complex with the active GPCR, reinforcing respectively the effect of ϕ. This would also explain why the two parameters seem correlated.

The cohesion of the trimeric G protein, measured by the parameter γ, varies less than the other parameters. Also, as opposed to the other parameters, the highest value of γ is reached for the β regime and the lowest, for the α regime, and seems correlated with the level of $G_{\beta\gamma}$ signal. The fact that the β regime has the highest value for γ seems counterintuitive. This effect might result from the complex interplay between the different parameters and deserves to be analyzed on its own.

In terms of both signal and parameter values, the α and β regimes are drastically opposed, while the $\alpha\beta$ regime stands in between.

3.7.3.3 Signal transduction profiles for the different regimes

The GPCR model is a full signal transduction system, with one input signal, the total density of agonist ligand L_0, and two output signals, $\mathcal{S}_\alpha$ associated with G_α's effector E, and $\mathcal{S}_{\beta\gamma}$ associated with $G_{\beta\gamma}$'s effector E'. In Fig. 3.26 we illustrate the temporal profiles of the $\mathcal{S}_\alpha$ and $\mathcal{S}_{\beta\gamma}$ signals for different values of the input strengths as heat maps of the functions $\mathcal{S}_\alpha(L_0, t)$ and $\mathcal{S}_{\beta\gamma}(L_0, t)$. This visualization highlights the differences in basal signals between the different regimes, especially for $\mathcal{S}_{\beta\gamma}$. For instance, the α regime has the highest base $\mathcal{S}_{\beta\gamma}$ signal, while the β regime has the lowest. It is interesting to note that the signal that maximizes $G_{\beta\gamma}$ signaling is also the one that has the lowest base signal. For $\mathcal{S}_\alpha$ however, it is more difficult to see differences in the base signal among the regimes. Still, we can observe how the $\alpha\beta$ regime has a slightly higher maximum signal than the others. The main point here is that we can numerically store the signaling functions associated with a given regime of the model, which may be useful for example when using the GPCR model within a larger biochemical model, as it spares the cost of integrating the rate equations of the GPCR model.

3.7.3.4 Cytoplasmic driving conditions are optimal for signal transduction within this model

The physical meaning of the signal transduction by GPCRs lies in their GEF function [11]. In the previous subsections, we have seen how the regulation of GTPases by GEFs is fundamentally a non-equilibrium phenomenon. This far-from-equilibrium process is driven by the concentrations of GTP, GDP and P_i, which are chemostatted far from their equilibrium values. In order to investigate how the signaling properties of the module are influenced by the non-equilibrium driving, the signal and the quantity of GTP consumed by the system are plotted as a function of the driving (figure 3.27). When GTP, GDP and P_i are chemostatted at their equilibrium values, for all regimes, at the steady state no GTP is consumed and no signal is produced, for G_α and $G_{\beta\gamma}$ alike. As the driving increases, so do the GTP consumption and signal production, until a plateau is reached. Interestingly, the driving corresponding to the cytoplasmic conditions happens to be close to the

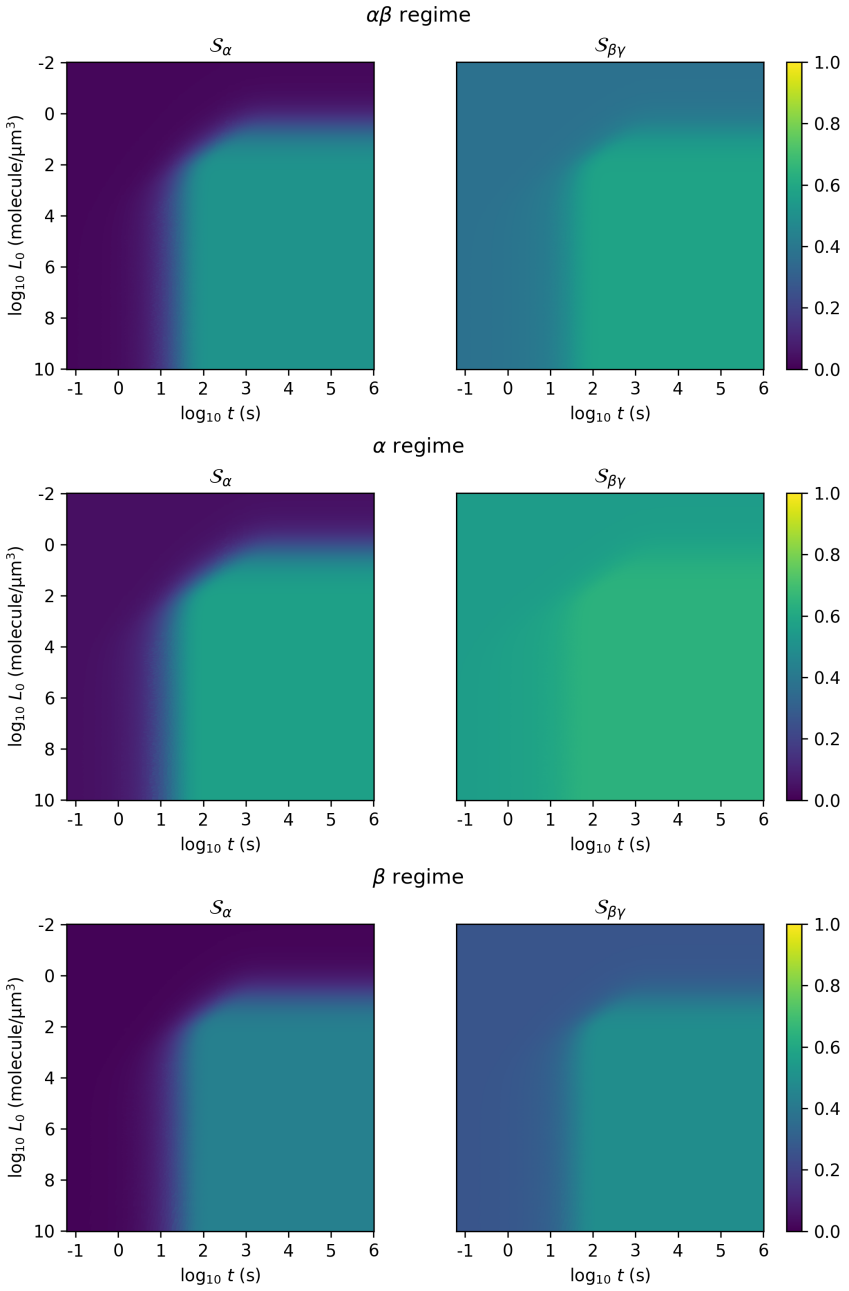


Figure 3.26: **Signal transduction profiles of the different regimes.** 2D plots of the output signaling the (t, L_0) plane. Plots on the left show G_α's signal, $\mathcal{S}_\alpha(L_0, t)$ and those one the right, $G_{\beta\gamma}$'signal, $\mathcal{S}_{\beta\gamma}(L_0, t)$. Each row corresponds to a different regime, from top to bottom, $\alpha\beta$, α and β. The signal is obtained by integrating the rate equations of the system. The values of the parameters used are those corresponding to the optimal regimes listed in table 3.3.

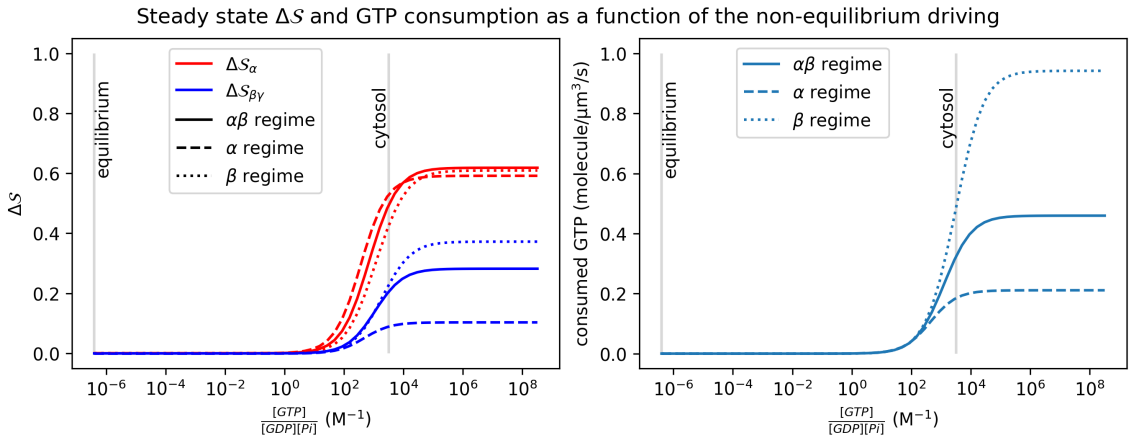


Figure 3.27: **Cytosolic driving is optimal for the GPCR signal transduction activity.** These plots show the effect of the non-equilibrium driving on the signal variation (left) and GTP consumption (right) for the three regimes: $\alpha\beta$ (maximizing both $\mathcal{S}_\alpha$ and $\mathcal{S}_{\beta\gamma}$), α (maximizing only $\mathcal{S}_\alpha$) and β (maximizing only $\mathcal{S}_{\beta\gamma}$). The signal and GTP consumption are evaluated at the stationary state.

(logarithmic) inflection point of the signal variation and GTP consumption curves. This suggests that cytoplasmic conditions are close to optimal for GPCR signal transduction, in terms of both signal strengths and of sensitivity to fluctuations in the chemical fuel that drives the signaling.

3.7.3.5 Stochastic dynamics

So far we have analyzed the dynamics of our systems by solving the corresponding rate equations. This is only correct if fluctuations in the copy number of the chemical species involved can be neglected and the reaction volume is large. These conditions are often not met in the cell. In this case, the full stochastic evolution of the system should be computed by solving the associated master equation. This can be done easily via kinetic Monte-Carlo simulations. In this work, we typically used the Gillespie algorithm [22] to investigate the signals of the GPCR network along with its fluctuations. In figure 3.28, we show stochastic trajectories of the system in each of the previously defined regimes. The trajectories start with the system in the basal steady state. At $t = 0$, the ligand is introduced inside the system, which then switches to the active steady state. For all regimes, the deterministic trajectories seem to match well the average values of the stochastic trajectories. These allow to investigate the fluctuations of the signal, which are absent at the mean-field level. An important observation is that $\mathcal{S}_{\beta\gamma}$'s fluctuations are quite large, relative to its variation from the basal to the active steady state. This is due to the relatively high base level of $G_{\beta\gamma}$ signal. For $\mathcal{S}_\alpha$, the fluctuations are visibly more important in the active steady state.

3.7.3.6 Fluctuations

In Fig. 3.29 we illustrate how the magnitude of fluctuations in the signals depends on the parameters of the model. As the density of input ligand L_0 increases in the system, the active stationary signal ($\mathcal{S}_\alpha$ and $\mathcal{S}_{\beta\gamma}$) also increases before reaching a plateau. At low input densities, the average active

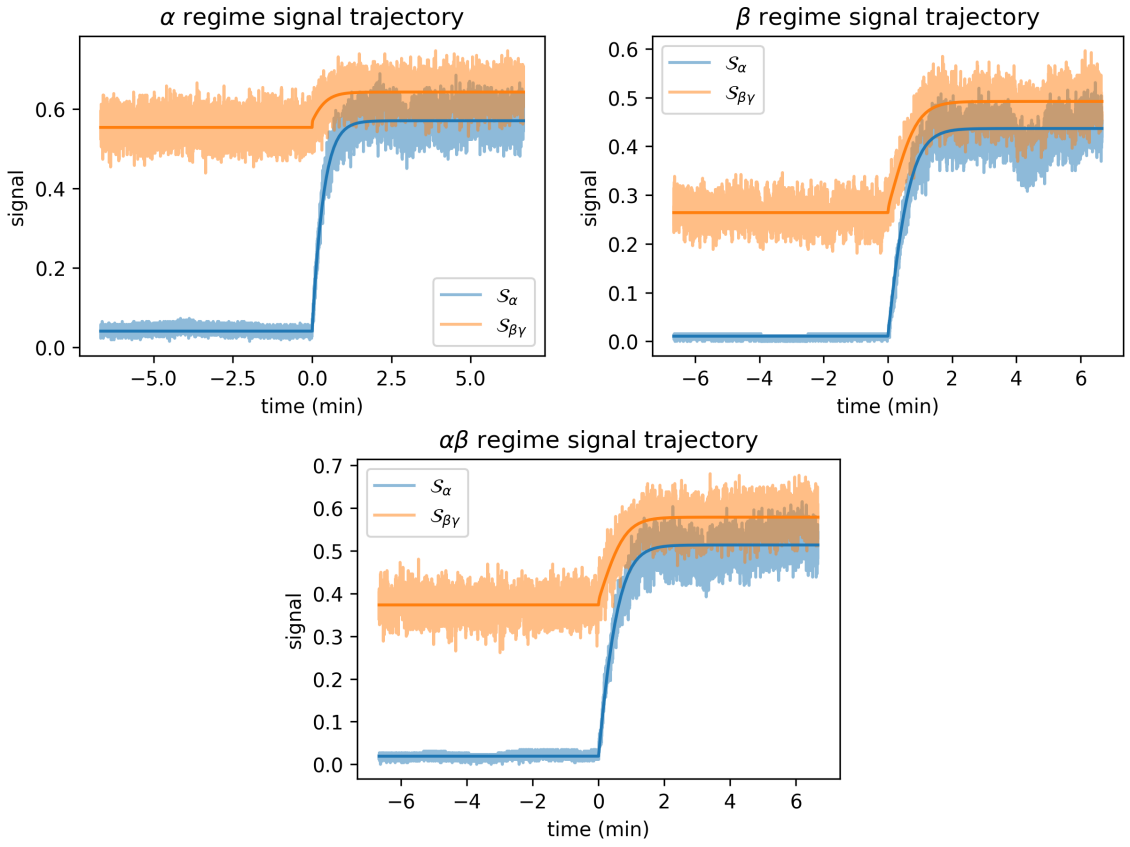


Figure 3.28: **Stochastic and deterministic trajectories of the system for the three different regimes $\alpha\beta$, α and β.** Each plot corresponds to a different regime. When $t < 0$, the system is at its basal steady state, with $L_0 = 0$. At $t = 0$, a total density of 100000 molecules is introduced into the system ($[L_0] = 100000$ molecules/μm^3). Orange and blue lines correspond respectively to $\mathcal{S}_\alpha$ and $\mathcal{S}_{\beta\gamma}$. Lighter, fluctuating lines represent the stochastic trajectories simulated with the Gillespie algorithm [22], while the darker lines show the deterministic numerical integrations of the system's rate equations. For each regime, parameter values for ϕ, f and γ are those reported in table 3.3.

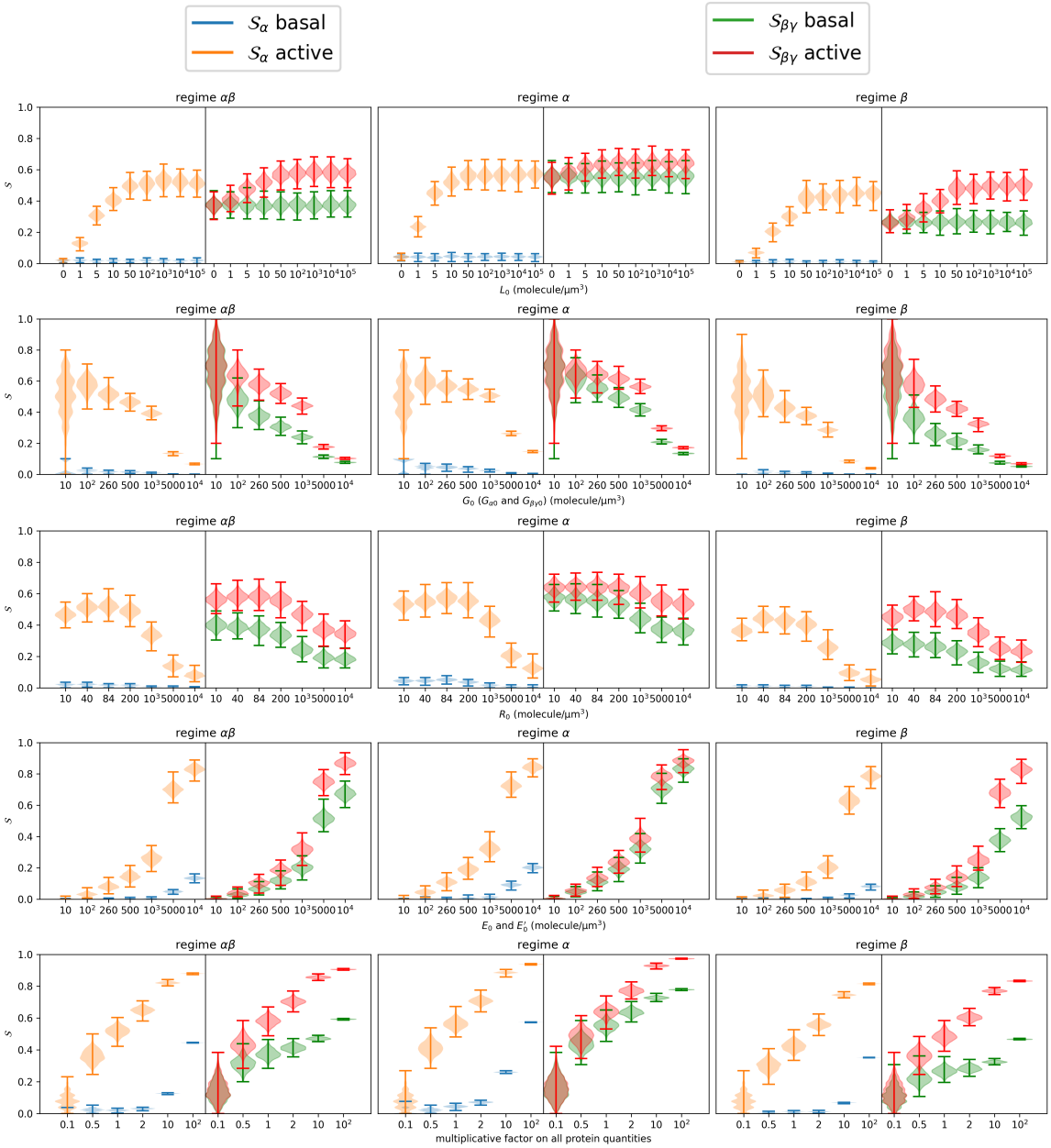


Figure 3.29: **Fluctuations of the signals, $\mathcal{S}_\alpha$ and $\mathcal{S}_{\beta\gamma}$, basal and maximal, in the different regimes as a function of the total densities of the different species.** Basal and active steady states are obtained deterministically, then the evolution of the system at steady state is simulated with the Gillespie algorithm [22]. Each plot shows the distribution of the stochastic signal at steady state. For each line, the signals are evaluated as a function of the variation of some species quantities. In the last line, all species, except *GTP*, *GDP* and P_i, are increased by some factor. For each regime, parameter values for ϕ, f and γ are those reported in table 3.3. We note that a full stoichiometric evolution from the onset might in principle lead the system to fluctuate about a different average value than the corresponding deterministic steady state. This steady state is presently underway.

signal is within the fluctuation interval of the base signal. As the signal increases, the active and base signal become more and more distinct. This is the case for $\mathcal{S}_\alpha$ for all regimes, but not for $\mathcal{S}_{\beta\gamma}$, for which the average active signal detaches from the base signal fluctuation interval only in the α and $\alpha\beta$ regimes. As the density of G proteins G_0 increases, the distributions of the steady state signal in the base and active states become more separated. For the total density of receptors R_0, there seems to be an optimal density past which the signal decreases and the distributions of the base and active signal converge. This is certainly due to the G proteins becoming saturated by the excess of receptors, decreasing the proportion of free G proteins. The density of effector E_0, when increased, also separates the distributions of the base and active signal distributions. Increasing the quantity of all proteins at once also results in a higher resolution of the signal, as expected.

3.7.3.7 Less $G_{\beta\gamma}$ means more G_α signal

The G_α subunit appears to be required for the GEF activity of the active GPCR [39]. Since we have built our model accordingly, the total quantity of $G_{\beta\gamma}$ has an impact on $\mathcal{S}_\alpha$. In order to investigate this property, the signals and proportion of G protein-bound GPCRs were evaluated as a function of the total quantity of $G_{\beta\gamma}$ units in the system (figure 3.30). Interestingly, these plots show a nonlinear effect on $\mathcal{S}_\alpha$ as $G_{\beta\gamma_0}$ - the total density of $G_{\beta\gamma}$- decreases below $G_{\alpha 0}$ - the total density of G_α. Indeed, as expected, with no $G_{\beta\gamma}$ at all, $\mathcal{S}_\alpha$ is at its basal value. However, relatively low quantities of $G_{\beta\gamma}$ (i.e. 50 $G_{\beta\gamma}$ molecules vs 260 G_α) lead the system to exhibit a higher G_α signal than for the standard quantities. A similar nonlinear phenomenon is observed for the fraction of receptors bound to G proteins. While overall the activation of the receptor leads to its separation from G proteins, the lowest fraction of G Protein-bound receptors is observed when the quantity of $G_{\beta\gamma}$ is low compared to G_α. The decrease in the fraction of bound receptors seems to match the increase in $\mathcal{S}_\alpha$. The $G_{\beta\gamma}$ signal, however, increases linearly with the total quantity of $G_{\beta\gamma}$.

3.7.3.8 Conclusion

Our GPCR model features 3 different regimes, which single out distinct regions of the parameter space 3.25. The α regime, which maximizes G_α signaling, is characterized by the highest G_α signal variation ($\Delta\mathcal{S}_\alpha$) (figure 3.25). It is also characterized by a very poor production of $G_{\beta\gamma}$ signal upon activation (figure 3.25): The average active $G_{\beta\gamma}$ signal is still within the fluctuation interval of the base signal, even with the highest densities of input ligand (figure 3.29). The β regime and the $\alpha\beta$ regime are more similar, each one maximizing slightly its respective signal(s). If the density of input ligand is large enough, both allow the G_α and $G_{\beta\gamma}$ signals to increase past their base fluctuation intervals. From a thermodynamic perspective, we showed that, for all regimes, the signal transduction is nearly optimal at the cytoplasmic GTP/GDP driving conditions. Indeed, the association constant maintained in cytoplasmic conditions is just large enough to reach the maximum sensitivity to signal variations.

3.8 Conclusion

While imposing detailed balance may shift the focus on the chemical equilibrium constants, it is fundamental to remember that far-from-equilibrium systems, as opposed to their isolated counterparts, have steady states that also depend on the reaction's kinetic constants themselves. As we have shown, the regulation of GTPases by GEFs, GAPs and GDIs can be explained without altering

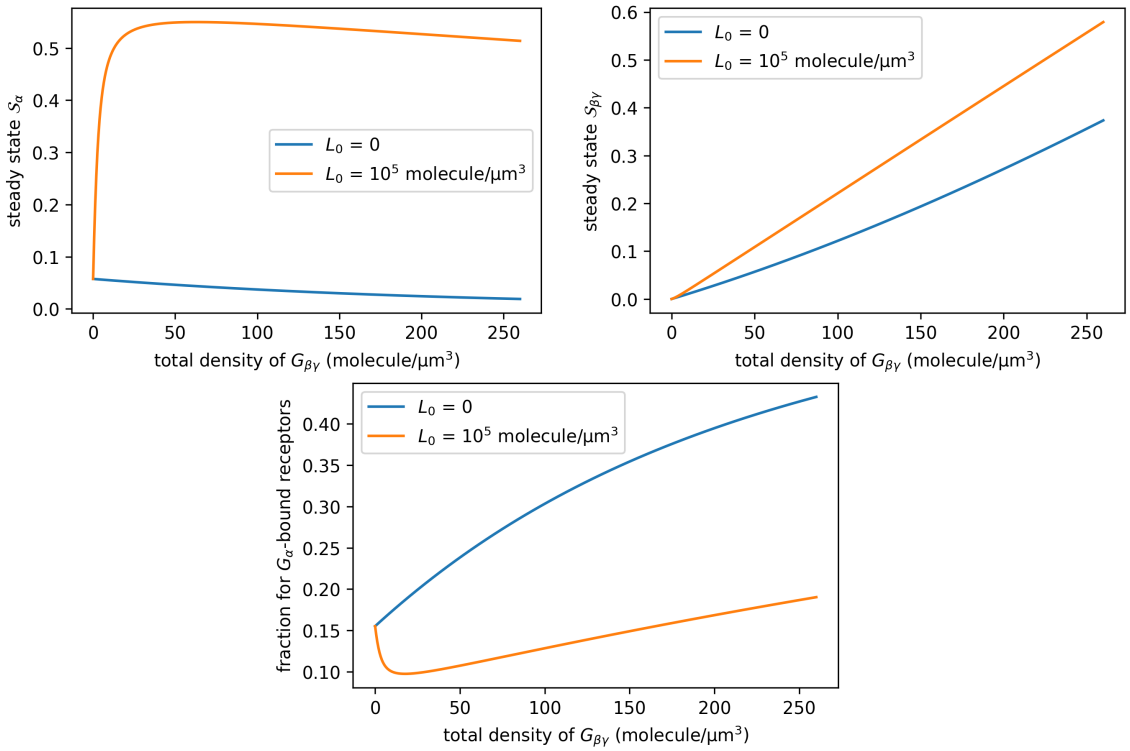


Figure 3.30: **The quantity of $G_{\beta\gamma}$ has a nonlinear effect on the signal.** The three plots show the signal for G_α and $G_{\beta\gamma}$, as well as the fraction of effectors bound to G_α, at steady state as a function of the total density of $G_{\beta\gamma}$ in the system, in the presence ($L_0 = 10^5$ molecule/μm^3) or absence of agonist ligand. These results are obtained for the $\alpha\beta$ regime. The value of G_0 used to normalize the signal is that from the standard set of parameters.

the equilibrium constants, simply by accelerating some reactions with respect to others: These are sheer non-equilibriums mechanisms. Previous works already highlighted how GTPases and their regulation by GEFs and GAPs are known for exhibiting non-equilibrium behaviors [41] [13].

Let us note that, while we focus here on GTPases, the very same base module is shared many important energy-consuming molecular machines in the cell, such as Hsp70, as previously highlighted by Nguyen et al [41]. Thus many of the observations made here should also be relevant for such systems.

Considering GTPase reaction networks as signal transduction systems, the output signal can be expressed as a function of the input signal and time, as well as the module quantitative and kinetic parameters: $out = \mathcal{S}(in, t, \overrightarrow{\lambda})$. When downstream effectors are present in the system, GTPase-bound effectors may be used to represent the signal. Otherwise, active GTPases may be used as a proxy. In this case, it is necessary to take into account the possible competition of the various GTPase partners with the appropriate downstream effector omitted.

The GPCR signal transduction system is a full example of signal transduction pathway, with the ligand as input signal, and two output signals, associated with G_α and $G_{\beta\gamma}$, through their respective effector. By simplifying the parameterization of the model, we studied 3 of its regimes: The α regime which maximizes G_α signaling, the β regime which maximizes $G_{\beta\gamma}$ signaling, and the $\alpha\beta$, regime which maximizes both simultaneously. Each regime corresponds to a separate location in the parameter space. The $\alpha\beta$ and β regimes both enable G_α and $G_{\beta\gamma}$ signaling, while the α regime only enables G_α signal increase. All of the three signals seem to work optimally within the cell cytoplasmic conditions, in terms of GTP, GDP and P_i driving.

Perspectives for future works would include:

- Comparing the behavior of the model to experimental data, such as immunofluorescence or immunoprecipitation kinetic measurements, and producing experimental screenings of reaction kinetic rates, affinities and species quantities.
- Incorporating diffusion, cell shapes and compartments into the models, as well as protein synthesis, degradation and trafficking.
- Considering other types of ligands (inverse agonists, partial agonists, etc.), as it was done by Stein et al [51].
- Addressing more precise GPCR species and G_α, G_β and G_γ types. More generally, exploring other GPCR networks.
- Investigating GPCR regulatory mechanisms (Arrestins [1](pages 848-849), G protein trafficking [62], regulators of G protein signaling (RGSs) [1](pages 832), receptor internalization [1](pages 848-849), ...).
- Modeling upstream and downstream events, such as those involved in chemotaxis.

Chapter 4

Conclusion

4.1 Summary and results

In the first chapter, we presented STReNGTHS, a simulation software that we developed. Modeling complex reaction or reaction-diffusion networks, such as the signal-transduction systems studied in the second chapter, requires specific tools with appropriate modeling features and supporting stochastic and deterministic methods. STReNGTHS is a Python package that serves such a purpose. It implements modeling classes representing the fundamental concepts of reaction-diffusion: reaction-diffusion networks, consisting of a set of reactions operating on a set of chemical species, discrete reaction-diffusion spaces, which are grids or graphs of individual reaction subvolumes, termed cells, across which diffusion is modeled as a first-order reaction [4], and finally, reaction-diffusion systems, which associate a reaction-diffusion network with a diffusion space. Such a system has a state, defined as the distribution of the quantity of the different species of the network in the space, represented as a matrix.

STReNGTHS offers a general interface for simulation algorithms, termed simulation engine. This abstract class may be implemented with different algorithms and make it easier to extend the software with new methods. So far, three engines have been implemented. One, deterministic, implements the simple Euler method, which integrates the rate equations of the system with a fixed time step. Two others, stochastic, implement respectively the Gillespie algorithm [22] and its τ-leap approximation [21]. All those methods generate discrete trajectories, which are sequences of system states sampled during the simulation.

In order to account for the different compartments of biochemical systems, it is possible to assign a cell of a reaction-diffusion space some type. We termed those reaction-diffusion environments, and they are one of the main features of the software. A lot of parameters can be set environment-wise, allowing for the creation of systems with complex, inhomogeneous structures. Additionally, it is possible to chemostate, locally or globally, the different species of a system to account for non-equilibrium driving.

Another interesting feature is the possibility to coarse-grain a grid space as a graph space. This is useful since graph spaces are quite inconvenient to define on their own. This can be useful to

optimize some systems, where the level of spatial resolution required is not the same everywhere in the system. It can also be used to remove inert cells from the system, reducing the number of cells to be considered by simulation algorithms.

STReNGTHS is open-source, with an online public repository open to contributions. It can be easily installed as most other packages. The software is an active project, and will hopefully continue to grow.

In the second chapter, we explored the signaling regimes of different GTPase reaction networks of increasing complexity, culminating with a full model of GPCR signal transduction.

We started by providing a formal definition of the different reaction networks and patterns we are dealing with. This started with the elementary GTPase module, which only contains one GTPase species and its substrates. We then added one partner, either generic, binding all states of the GTPase, or strictly state-specific for one given state, and then multiple partner species simultaneously. Finally, we introduced a kinetic model of GPCR signal transduction, which accounts for the full dissociation of the G_α and $G_{\beta\gamma}$.

One important aspect of this work is that we propose thermodynamically consistent models. To do so, we define for each network a set of constraints on its equilibrium constants imposing microscopic reversibility, or detailed balance, to the whole network. This requires identifying the different reaction cycles in the network, and imposing an absence of net reaction currents at thermodynamic equilibrium [8]. As a network grows in size, new constraints may appear, but those existing never disappear. As a consequence, defining a set of constraints may be done even more efficiently by considering large reaction networks as the composition of more fundamental patterns, for which we may already know the constraints. We provided protocols to build thermodynamically consistent sets of equilibrium constants from the constraints, for the different reaction networks we study. Those protocols were validated experimentally by showing that imposing the constraints indeed results in an absence of net reaction currents at equilibrium.

Signal transduction systems work as input-output processes. Input and output signals are represented by different chemical species. In particular, the output signal is a normalized value, between 0 and 1, where 0 represents the complete absence of signal and 1, the theoretical maximum signal. On the onset of some input signal, corresponding to some chemical species, the system switches from a basal steady state signal to another - active - steady state signal. We thus use a function $out = \mathcal{S}(in, t, \overrightarrow{\lambda})$ which defined the output signal out as a function of the input signal in, time t and the different parameters of the signal transduction systems represented by the vector $\overrightarrow{\lambda}$. For GTPase systems, the output signal is defined either as the normalized quantity of effectors or as the quantity of available active GTPases, while the input signal - when there is one - is generally the quantity of GEF. In the second case, this still requires to account for effector interactions, even if those are not present in the system. In the specific case of the GPCR signal transduction system, there are two parallel output signals, $\mathcal{S}_\alpha$ and $\mathcal{S}_{\beta\gamma}$, associated with the effector of G_α and $G_{\beta\gamma}$, and one input signal, the quantity of agonist ligand.

We explored the different regulatory regimes associated with the base GTPase module. In particular, we explored regimes associated with GEFs, GAPs and GDIs. We highlighted the fundamental non-equilibrium nature of these regimes, which are only possible when driven by non-equilibrium concentrations of GTP, GDP and P_i. All those regulatory regimes rely on modifications of the kinetic parameters of the fundamental GTPase cycle and do not necessarily require changing the

equilibrium constants.

Next, we explored those same regimes, but this time mediated by generic partners. The regimes themselves have the same properties as those from the base GTPase module. However, depending on whether or not we consider the regulatory partners to compete with effectors, the way the signal is measured changes, and we showed how this modifies completely the regulatory effect. We also showed how a generic effector with a strong preference over a specific state of the GTPase may be modeled as a strict-specific partner.

For the GPCR model, we defined the whole set of kinetic parameters as a function of only three free parameters: ϕ, the strengths of the GEF effect of the active GPCR, σ, the GDP-state preference of $G_{\beta\gamma}$, and γ, and the strengths of the G_α-GDP-$G_{\beta\gamma}$ interaction. Exploring this three-parameter space, we defined 3 different regimes for the model: the α regime, which maximizes G_α signaling, the β regime, which maximizes $G_{\beta\gamma}$ signaling, and the $\alpha\beta$ regime, which maximizes both G_α and $G_{\beta\gamma}$ signaling. We determined the signal transduction profiles associated with each regime and studied the corresponding fluctuations. We also showed that the cytoplasmic conditions provide an optimal driving for the activity of the GPCR network, for all regimes alike.

4.2 Limitations and perspectives

STReNGTHS, the modeling and simulation tool that we have developed, also still has a lot of room for improvement and future developments. The only rate equation integration method implemented by the software is the Euler method, which is far too rudimentary to be useful for real simulations. It is thus primordial to develop more advanced methods in the future. The stochastic methods currently implemented are quite powerful. However, the current implementation of the τ-leap algorithm [21] would greatly benefit from an adaptative time step. In fact, aside from the Original Gillespie algorithms [22], most deterministic and stochastic methods, including the Euler method, would be a lot more interesting with an adaptative time-step. When it comes to spatial resolution, the computational cost and duration of a simulation are linearly impacted by the number of cells in the system with traditional sequential CPU implementations. To solve this problem, it is possible to take advantage of Graphics Processing Units (GPU) and their myriad of computation units to massively parallelize the work. Thus, developing CPU-GPU massively parallel implementations of the methods is also one of the main plans for the future of STReNGTHS.

The construction of thermodynamically consistent reaction networks through DB constraints is necessary to properly understand how biochemical systems work far from equilibrium. In this work, we extend a previous method ([8]) with a modular approach based on the identification of known patterns in large reaction networks. However, ultimately, this still requires to build the final sets of DB constants by hand and also to find the proper way to apply them to the equilibrium constants. As a consequence, the next steps would involve the automatization of the process, with as ultimate goal being able to extract and apply constraints automatically for any arbitrary reaction-diffusion network.

Moreover, detailed balance is not the only criterion to evaluate the consistency of a reaction network. For instance, multimolecular reactions involving collisions are limited by the diffusion coefficients of their substrates. In the case of complex formation reactions, this means that while k_{off} may have any value, k_{on} may be unrealistic if too large. Such phenomena should also be considered when evaluating the consistency of a reaction network.

While our GPCR model is built on well-established knowledge on GPCRs and GTPases and relies on reasonable hypotheses, it still lacks experimental validation. This would require experimental data that we could compare with one of the outputs of the model. The difficulty comes from the fact that GPCRs, in living cells, are probably rarely isolated, due to coupling and feedback with upstream and downstream signaling events.

Also, here, we only studied GPCR systems consisting of homogeneous, well-mixed, reaction volumes. However, in reality, such systems involve multiple compartments. This includes membranes, such as the plasma membranes, vesicles, Golgi apparatus, endoplasmic reticulum or mitochondrial membranes. There is also, of course, the cytosol and the extracellular space, which are far from being simple environments. Proteins, from their synthesis to their degradation, undergo complex trafficking across different compartments. In future works, it may be interesting to account for such trafficking and diffusion in the GPCR model.

Bibliography

[1] Bruce Alberts, Alexander Johnson, Julian Lewis, David Morgan, Martin Raff, Keith Roberts, and Peter Walter. *Biologie moléculaire de la cellule*. Lavoisier Médecine Sciences, 6 edition, 2017. Translated by Michel Darmon.

[2] R. A. Alberty. Thermodynamics of systems of biochemical reactions. *J. Theor. Biol.*, 215(4):491–501, April 2002.

[3] Tim Bergbrede, Nam Chuky, Stefan Schoebel, Wulf Blankenfeldt, Matthias Geyer, Evelyn Fuchs, Roger S. Goody, Francis Barr, and Kirill Alexandrov. Biophysical analysis of the interaction of Rab6a GTPase with its effector domains. *J. Biol. Chem.*, 284(5):2628–2635, January 2009.

[4] David Bernstein. Simulating mesoscopic reaction-diffusion systems using the gillespie algorithm. *Phys. Rev. E*, 71:041103, Apr 2005.

[5] Lisa M. Bishop and Hong Qian. Stochastic Bistability and Bifurcation in a Mesoscopic Signaling System with Autocatalytic Kinase. *Biophysical Journal*, 98(1):1–11, January 2010. Publisher: Elsevier.

[6] David E. Clapham and Eva J. Neer. G protein βγ subunits. *Annual Review of Pharmacology and Toxicology*, 37(Volume 37, 1997):167–203, April 1997. Publisher: Annual Reviews.

[7] John Colicelli. Human RAS superfamily proteins and related GTPases. *Science's STKE*, 2004(250):re13–re13, 2004.

[8] David Colquhoun, Kathryn A. Dowsland, Marco Beato, and Andrew J.R. Plested. How to impose microscopic reversibility in complex reaction mechanisms. *Biophysical Journal*, 86(6):3510–3518, 06 2004.

[9] Geoffrey M. Cooper. Pathways of Intracellular Signal Transduction. In *The Cell: A Molecular Approach. 2nd edition*. Sinauer Associates, 2000.

[10] Paolo De Los Rios and Alessandro Barducci. Hsp70 chaperones are non-equilibrium machines that achieve ultra-affinity by energy consumption. *eLife*, May 2014.

[11] Denis J. Dupré, Mélanie Robitaille, R. Victor Rebois, and Terence E. Hébert. The role of Gβγ subunits in the organization, assembly, and function of gpcr signaling complexes. *Annual Review of Pharmacology and Toxicology*, 49(1):31–56, 2009. PMID: 18834311.

[12] Pugh Edward, Jr. Amplification and kinetics of the activation steps in phototransduction. *Biochim. Biophys. Acta*, 1141:111–49, March 1993.

[13] Andreas Ehrmann, Basile Nguyen, and Udo Seifert. Interlinked GTPase cascades provide a motif for both robust switches and oscillators. *Journal of the Royal Society Interface*, 16(157):20190198, August 2019.

[14] European Organization For Nuclear Research and OpenAIRE. Zenodo, 2013.

[15] Björn H. Falkenburger, Jill B. Jensen, and Bertil Hille. Kinetics of M1 muscarinic receptor and G protein signaling to phospholipase C in living cells. *J. Gen. Physiol.*, 135(2):81, February 2010.

[16] Dennis Fiegen, Lars-Christian Haeusler, Lars Blumenstein, Ulrike Herbrand, Radovan Dvorsky, Ingrid R. Vetter, and Mohammad R. Ahmadian. Alternative splicing of Rac1 generates Rac1b, a self-activating GTPase ∗. *J. Biol. Chem.*, 279(6):4743–4749, February 2004.

[17] Thibault Fillion and Francesco Piazza. Building and simulating a reaction-diffusion system — Strengths documentation, 2024. `https://strengths.readthedocs.io/en/latest/building_and_simulating_rds.html`. Accessed online (May 23, 2024).

[18] Thibault Fillion and Francesco Piazza. STReNGTHS, a Python package to model and simulate reaction-diffusion systems. *The Journal of Open Source Software*, 2024. To be published.

[19] Aki Fujioka, Kenta Terai, Reina E. Itoh, Kazuhiro Aoki, Takeshi Nakamura, Shinya Kuroda, Eisuke Nishida, and Michiyuki Matsuda. Dynamics of the Ras/ERK MAPK cascade as monitored by fluorescent probes ∗. *J. Biol. Chem.*, 281(13):8917–8926, March 2006.

[20] Takahiro K. Fujiwara, Kokoro Iwasawa, Ziya Kalay, Taka A. Tsunoyama, Yusuke Watanabe, Yasuhiro M. Umemura, Hideji Murakoshi, Kenichi G. N. Suzuki, Yuri L. Nemoto, Nobuhiro Morone, and Akihiro Kusumi. Confined diffusion of transmembrane proteins and lipids induced by the same actin meshwork lining the plasma membrane. *Molecular Biology of the Cell*, 27(7):1101–1119, April 2016.

[21] Daniel Gillespie. Approximate Accelerated Stochastic Simulation of Chemically Reacting Systems. *Journal of Chemical Physics*, 115:1716–1733, July 2001.

[22] Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, December 1977. Publisher: American Chemical Society.

[23] P. J. Goldschmidt-Clermont, M. I. Furman, D. Wachsstock, D. Safer, V. T. Nachmias, and T. D. Pollard. The control of actin nucleotide exchange by thymosin beta 4 and profilin. a potential regulatory mechanism for actin polymerization in cells. *Molecular Biology of the Cell*, 3(9):1015–1024, September 1992.

[24] Andrew B. Goryachev and Alexandra V. Pokhilko. Computational Model Explains High Activity and Rapid Cycling of Rho GTPases within Protein Complexes. *PLOS Computational Biology*, 2(12):e172, December 2006. Publisher: Public Library of Science.

[25] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant,

Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[26] 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. *Bioinformatics*, 32(21):3366–3368, November 2016.

[27] Johan Hattne, David Fange, and Johan Elf. Stochastic reaction-diffusion simulation with MesoRD. *Bioinformatics*, 21(12):2923–2924, June 2005.

[28] Martin Heck and Klaus Hofmann. Maximal rate and nucleotide dependence of rhodopsin-catalyzed transducin activation - initial rate analysis based on a double displacement mechanism. *The Journal of biological chemistry*, 276:10000–9, 03 2001.

[29] Moritz Hoffmann, Christoph Fröhner, and Frank Noé. ReaDDy 2: Fast and flexible software framework for interacting-particle reaction dynamics. *PLOS Computational Biology*, 15(2):e1006830, February 2019. Publisher: Public Library of Science.

[30] John D. Hunter. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, 9(3):90–95, May 2007. Conference Name: Computing in Science & Engineering.

[31] Vladimir L. Katanaev and Matey Chornomorets. Kinetic diversity in G-protein-coupled receptor signalling. *Biochemical Journal*, 401(Pt 2):485–495, January 2007.

[32] Vsevolod Katritch, Vadim Cherezov, and Raymond C. Stevens. Structure-function of the G protein–coupled receptor superfamily. *Annual Review of Pharmacology and Toxicology*, 53(1):531–556, 2013. PMID: 23140243.

[33] Holger Krekel, Bruno Oliveira, Ronny Pfannschmidt, Floris Bruynooghe, Brianna Laugher, and Florian Bruhin. pytest 8.2.1, 2004. `https://github.com/pytest-dev/pytest`. Accessed online (May 21, 2024).

[34] Eric O. LEBIGOT. Uncertainties : a Python package for calculations with uncertainties.

[35] Thomas Linnemann, Matthias Geyer, Birgit K. Jaitner, Christoph Block, Hans Robert Kalbitzer, Alfred Wittinghofer, and Christian Herrmann. Thermodynamic and kinetic characterization of the interaction between the Ras binding domain of AF6 and members of the Ras subfamily. *Journal of Biological Chemistry*, 274(19):13556–13562, May 1999.

[36] Boqun Liu, Bert Poolman, and Arnold J. Boersma. Ionic strength sensing in living cells. *ACS Chem. Biol.*, 12(10):2510, October 2017.

[37] Youtao Liu, Jesus Lacal, Richard A. Firtel, and Arjan Kortholt. Connecting G protein signaling to chemoattractant-mediated cell polarity and cytoskeletal reorganization. *Small GTPases*, 9(4):360–364, October 2016.

[38] Jeff S. McGough and Kyle Riley. Pattern formation in the Gray–Scott model. *Nonlinear Analysis: Real World Applications*, 5(1):105–121, February 2004.

[39] William E. McIntire. Structural determinants involved in the formation and activation of G protein $\beta\gamma$ dimers. *Neurosignals*, 17(1):82–99, 02 2009.

[40] Ron Milo and Bob Phillips. *Cell biology by the numbers*. Garland Science, December 2015.

[41] Basile Nguyen, David Hartich, Udo Seifert, and Paolo De Los Rios. Thermodynamic bounds on the ultra- and infra-affinity of Hsp70 for its substrates. *Biophysical Journal*, 113(2):362–370, July 2017.

[42] Nomenclature Committee of the International Union of Biochemistry and Molecular Biology (NC-IUBMB). EC 3.6.5 acting on GTP; involved in cellular and subcellular movement. (october 15, 2024).

[43] Birgitta Olofsson. Rho guanine dissociation inhibitors: Pivotal molecules in cellular signalling. *Cellular Signalling*, 11(8):545–554, August 1999.

[44] Manishha Patel and Jean-François Côté. Ras GTPases' interaction with effector domains. *Communicative & Integrative Biology*, 6(4):e24298, 2013. PMID: 23986800.

[45] Th Ruijgrok and M. Ruijgrok. A reaction-diffusion equation for a cyclic system with three components. *Journal of Statistical Physics*, 87:1145–1164, June 1997.

[46] Matthias Schaks, Grégory Giannone, and Klemens Rottner. Actin dynamics in cell migration. *Essays in Biochemistry*, 63(5):483–495, October 2019.

[47] Sigma-Aldrich. Adenosine 5'-triphosphate disodium salt. `https://www.sigmaaldrich.com/deepweb/assets/sigmaaldrich/product/documents/415/339/a2383pis.pdf`, accessed online on the 14th of may 2024.

[48] A. V. Smrcka. G protein $\beta\gamma$ subunits: Central mediators of G protein-coupled receptor signaling. *Cellular and Molecular Life Sciences*, 65:2191–2124, 02 2008.

[49] Siyang Song, Wenshu Cong, Shurong Zhou, Yujie Shi, Wenbing Dai, Hua Zhang, Xueqing Wang, Bing He, and Qiang Zhang. Small GTPases: Structure, biological function and its interaction with nanoparticles. *Asian Journal of Pharmaceutical Sciences*, 14(1):30–39, January 2019.

[50] Rob J. Stanley and Geraint M. H. Thomas. Activation of G proteins by guanine nucleotide exchange factors relies on GTPase activity. *PLOS ONE*, 11(3):1–12, 03 2016.

[51] Richard Stein and Frederick Ehlert. A kinetic model of GPCRs: Analysis of G protein activity, occupancy, coupling and receptor-state affinity constants. *The FASEB Journal*, 29(S1), April 2015.

[52] Viktoriya Syrovatkina, Kamela O. Alegre, Raja Dey, and Xin-Yun Huang. Regulation, signaling, and physiological functions of G-proteins. *Journal of Molecular Biology*, 428(19):3850–3868, 2016. Molecular Basis of Signal Transduction.

[53] Nicolas Tapon and Alan Hall. Rho, Rac and Cdc42 GTPases regulate the organization of the actin cytoskeleton. *Current Opinion in Cell Biology*, 9(1):86–92, 1997.

[54] Zakir Tnimov, Zhong Guo, Yann Gambin, Uyen T. T. Nguyen, Yao-Wen Wu, Daniel Abankwa, Anouk Stigter, Brett M. Collins, Herbert Waldmann, Roger S. Goody, and Kirill Alexandrov. Quantitative analysis of prenylated RhoA interaction with its chaperone, RhoGDI $*$. *J. Biol. Chem.*, 287(32):26549–26562, August 2012.

[55] Thomas W. Traut. Physiological concentrations of purines and pyrimidines. *Molecular and Cellular Biochemistry*, 140(1):1–22, 1994.

[56] N. G. Van Kampen. *Stochastic processes in physics and chemistry*. Elsevier, 3 edition, 2007.

[57] Beatriz Vega, Laura Martínez Muñoz, Borja L. Holgado, Pilar Lucas, J. M. Rodríguez-Frade, Ana Calle, J. L. Rodríguez-Fernández, Laura M. Lechuga, José F. Rodríguez, Ricardo Gutiérrez-Gallego, and Mario Mellado. Technical advance: Surface plasmon resonance-based analysis of CXCL12 binding using immobilized lentiviral particles. *J. Leukocyte Biol.*, 90(2):399–408, May 2011.

[58] Priam Villalonga and Anne J. Ridley. Rho GTPases and actin cytoskeleton dynamics. In William J. Lennarz and M. Daniel Lane, editors, *Encyclopedia of Biological Chemistry*, pages 714–718. Elsevier, New York, 2004.

[59] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, and Paul van Mulbregt. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272, March 2020. Publisher: Nature Publishing Group.

[60] Biran Wang, Peng Guo, and Debra Auguste. Mapping the CXCR4 receptor on breast cancer cells. *ResearchGate*, August 2017.

[61] Olgatina Bucco Wayne R. Leifert, Amanda L. Aloia and Edward J. McMurchie. GPCR-induced dissociation of G-protein subunits in early stage signal transduction. *Molecular Membrane Biology*, 22(6):507–517, 2005.

[62] Philip B. Wedegaertner. G Protein Trafficking. *Sub-cellular biochemistry*, 63:193, 2012. Publisher: NIH Public Access.

[63] Lawrence Weinstein and John Adam. *Guesstimation: Solving the World's Problems on the Back of a Cocktail Napkin*. Princeton University Press, 2008.

[64] Stefan Wils and Erik De Schutter. STEPS: modeling and simulating complex reaction-diffusion systems with Python. *Frontiers in Neuroinformatics*, 3, June 2009. Publisher: Frontiers.

[65] Baolin Zhang, Yaqin Zhang, Zhi xin Wang, and Yi Zheng. The role of Mg2+ cofactor in the guanine nucleotide exchange and GTP hydrolysis reactions of rho family GTP-binding proteins. *Journal of Biological Chemistry*, 275(33):25299–25307, August 2000.

Appendices

Appendix A

Modeling GTPase Signal Transduction Reaction Networks

A.1 Adapting a pair of reaction rate constants to match some equilibrium dissociation constant

Let us say we have K_d, k_{on} and k_{off} such that

$$K_d \neq \frac{k_{off}}{k_{on}}. \tag{A.1}$$

We thus search for (α, β) such that

$$K_d = \frac{k_{off}(1+\alpha)}{k_{on}(1+\beta)}. \tag{A.2}$$

that minimizes (α, β). The latter can be written in the form

$$(\alpha, \frac{k_{off}(1+\alpha)}{K_d k_{on}} - 1) \tag{A.3}$$

we thus aim to minimize the norm of this vector

$$M(\alpha) = \sqrt{\alpha^2 + \left(\frac{k_{off}(1+\alpha)}{K_d k_{on}} - 1\right)^2} \tag{A.4}$$

The minima of $M(\alpha)$ correspond to that of $M(\alpha)^2$ which is a second degree polynom

$$M(\alpha)^2 = (1 + \frac{{k_{off}}^2}{{k_{on}}^2 {K_d}^2})\alpha^2 + (\frac{2{k_{off}}^2}{{k_{on}}^2 {K_d}^2} - \frac{2k_{off}}{k_{on}K_d})\alpha + \frac{{k_{off}}^2}{{k_{on}}^2 {K_d}^2} - \frac{2k_{off}}{k_{on}K_d} + 1 \tag{A.5}$$

for which the minimum can be found as the root of its derivative

$$\frac{d}{d\alpha} M(\alpha)^2 = (1 + \frac{{k_{off}}^2}{{k_{on}}^2 {K_d}^2})2\alpha + (\frac{2{k_{off}}^2}{{k_{on}}^2 {K_d}^2} - \frac{2k_{off}}{k_{on}K_d}) = 0 \tag{A.6}$$

which can be solved by

$$\alpha = \frac{-(\frac{2{k_{off}}^2}{{k_{on}}^2{K_d}^2} - \frac{2k_{off}}{k_{on}K_d})}{2(1 + \frac{{k_{off}}^2}{{k_{on}}^2{K_d}^2})} \tag{A.7}$$

or more simply written

$$\begin{aligned} \alpha &= \frac{-(2U^2 - 2U)}{2(1 + U^2)} \\ U &= \frac{k_{off}}{K_d k_{on}} \end{aligned} \tag{A.8}$$

We eventually also get β as

$$\beta = U(1 + \alpha) - 1 \tag{A.9}$$

We now have $k'_{off} = k_{off}(1 + \alpha)$ and $k'_{on} = k_{on}(1 + \beta)$ that satisfies $K_d = \frac{k'_{off}}{k'_{on}}$.

Thibault FILLION

Effets de non-équilibre en biologie cellulaire : Modélisation mathématique et informatique de la transduction du signal par les protéines G et autres réseaux biochimiques de réaction-diffusion.

Résumé :

Les propriétés diverses des organismes vivants émergent de réseaux complexes de réactions biochimiques. Ces systèmes reposent fortement sur des effets de non-équilibre, résultant de la conduction hors-équilibre soutenue de différentes espèces, des photons arrivant continuellement du soleil aux nucléotides di et triphosphate maintenus à des niveaux de non-équilibre dans le cytosol par le métabolisme. En effet, les systèmes biochimiques sont des systèmes ouverts, qui maintiennent leur structure en dissipant l'énergie en provenance de l'extérieur par des mécanismes complexes dont il reste encore beaucoup à comprendre. Lors de la construction de modèles aux propriétés émergentes, il est alors important de tenir rigoureusement compte des sources de non-équilibre, afin de faire la distinction entre relaxation à l'équilibre et comportements de non-équilibre et de quantifier la dissipation d'énergie associée à des conditions stationnaires de non-équilibre spécifiques.

Les cellules ont la capacité de percevoir et répondre à des signaux chimiques externes. Cette capacité est désignée sous le terme de transduction du signal, et représente un concept central en biologie cellulaire. Comme pour les autres processus biochimiques, la transduction du signal repose sur des réseaux complexes de réactions chimiques, impliquant principalement des enzymes distribuées dans de différents compartiments biologiques, en particulier l'environnement extracellulaire, le cytoplasme et la membrane plasmique. Il existe deux superfamilles proches de protéines jouant un rôle fondamental dans la transduction du signal : les GTPase et les Récepteurs Couplés aux Protéines G (RCPG). Nous étudions des modèles de réseaux de GTPases de complexité croissante, et présentons un modèle cinétique de signalisation par RCPG. Chaque réseau est construit de manière thermodynamiquement consistante. La réversibilité microscopique est imposée via des contraintes sur les ensembles de constantes d'équilibre. Nous présentons une méthode pour construire des ensembles complexes de contraintes par une approche modulaire. En utilisant une approche informatique, nous étudions la dynamique des différents systèmes, soulignant leurs différents régimes, correspondant à des régions caractéristiques de leur espace de paramètres cinétiques associés avec des comportements biologiques spécifiques. Dans le cas du modèle RCPG, de taille conséquente, nous avons été capables de réduire l'ensemble de ses paramètres cinétiques à trois paramètres libres spécifiques, et avons étudiés les régimes correspondants.

L'étude de ces système biologique requiert des approches informatiques reposant sur des outils dédiés. Durant la thèse, nous avons développé un tel outil, que nous avons dénommé STReNGTHS. Il s'agit d'un package Python open-source qui offre une interface pour la modélisation et la simulation de systèmes biologiques de réaction-diffusion dans des environnements complexes. Les systèmes sont décrits comme des espaces discrets dans lesquels les espèces peuvent réagir et diffuser. Chaque nœud de réaction individuel est appelé une «cellule». Les compartiments biologiques sont modélisés en définissant différent types de cellules, qui sont désignées sous le terme d'environnements de réaction-diffusion. Des chémostats peuvent également être définis, localement ou globalement, pour imposer des conditions de non-équilibre. Le logiciel propose une interface générique pour les algorithmes de simulation, qui est utilisée pour implémenter à la fois des moteurs d'évolution temporelle déterministes et stochastiques. Dans cette thèse, nous présentons le logiciel, ainsi que le cadre théorique correspondant, accompagné de quelques exemples d'applications choisis à des fins d'illustration.

Mots clés : réaction-diffusion dans des environnements complexes, physique statistique de non équilibre, transduction du signal, méthodes de Monte-Carlo cinétiques, processus stochastiques, approches informatiques

Non-equilibrium effects in cell biology: Mathematical and computational modeling of G-protein signal transduction and other biochemical reaction-diffusion networks.

Summary:

The diverse properties of living organisms emerge from complex networks of biochemical reactions. Those systems heavily rely on non-equilibrium effects, resulting from the sustained driving of different species, from photons continuously arriving from the sun, to di and triphosphate nucleotides kept at non-equilibrium levels in the cytoplasm by the metabolism. Indeed, biochemical systems are open systems, which maintain their structure by dissipating the energy coming from the outside through complex mechanisms which are still to be understood. When building bottom-up models for such systems, it is then important to properly account for the sources of non-equilibrium driving, in order to make a distinction between equilibrium relaxation and non-equilibrium behavior and quantify the energy dissipation associated with specific stationary non-equilibrium conditions.

Cells have the ability to sense and respond to external chemical signals. This ability is referred to as signal transduction, and is a central concept in cell biology. As for other biochemical processes, signal transduction relies on complex networks of chemical reactions, involving mostly enzymes spread across different biological compartments, specifically the extracellular space, the cytoplasm and the plasma membrane. There are two closely-related kinds of proteins which play a fundamental role in signal transduction: GTPases and G-protein Protein Coupled Receptors (GPCRs).

- On the one hand, GTPases are binary switch enzymes, which, once activated by upstream Guaine nucleotide Exchange Factors (GEFs), transduce the signal by activating a downstream effector.
- On the other hand, GPCRs are transmembrane receptors coupled with G-proteins, which are GTPases themselves. Upon activation by an agonist ligand, GPCRs activate the G proteins which then carry-on the signal transduction.

In this thesis, we study models of GTPase networks of increasing complexity, and eventually introduce our own kinetic model of GPCR signaling. The different models use parameters gathered from the scientific literature and others from educated guesses. Each network is also thermodynamically consistent. Microscopic reversibility is imposed by constraining the sets of equilibrium constants. We present a method to build complex sets of constraints through a modular approach. Using a computational approach, we study the dynamics of the different systems, underlying their different regimes, corresponding to characteristic regions of their kinetic parameter space associated with specific biological behaviors. In the case of the large GPCR model, we were able to reduce the ensemble of its kinetics parameters to a set of three specific free parameters, and studied the corresponding regimes.

The study of such biochemical systems requires computational approaches relying on dedicated tools. During the PhD, we developed such a tool, that we termed STReNGTHS. This is an open-source Python Package that offers an interface for the modeling and the simulation of biochemical reaction-diffusion systems in complex environments. Systems are described as discrete spaces in which species react and diffuse. Each individual reaction node is called a cell. Biological compartments are modeled by defining different types of cells, referred to as reaction-diffusion environments. Chemostats can also be set, locally or globally, to impose non-equilibrium conditions. The software proposes a generic interface for simulation algorithm, that is used to implement both deterministic and stochastic time-evolution engines. In this thesis, we present the software, along with the corresponding theoretical framework and a few selected applications as an illustration.

Keywords: reaction-diffusion in complex environments, non-equilibrium statistical physics, signal transduction, kinetic Monte-Carlo methods, stochastic processes, computational approaches

Centre de Biophysique Moléculaire - Rue Charles Sadron CS 80054 - 45071 Orléans Cedex 2 - France