POLITECNICO DI TORINO

MASTER's Degree in
PHYSICS OF COMPLEX SYSTEMS

MASTER's Degree Thesis

A lattice-gas model for Rab5 domain formation on cell membranes

Supervisors

ANDREA GAMBA
CARLA BOSIA

Candidate

MATTEO BASSO

JULY 2025

Abstract

This thesis investigates the process of active Rab5 domain formation on the membrane of early endosomes. From a physical perspective, this process can be described as a phase separation, typical of self-organizing systems. Our goal is to observe this phenomenon using a reaction-diffusion computational model that simulates the temporal evolution of a chemical reaction system on a spherical surface, allowing for exchanges with the cytosol. To do this, we considered a set of reactions including the main regulators of the process: GEF, GAP, and GDI, and we identified a realistic set of system parameter values, using the work of Bezeljak et al. (2020) as a reference. A systematic parametrization of the model was then carried out to adapt it to the requirements imposed by our algorithm, and simulations were performed to see whether the system evolves spontaneously producing Rab5 domain formation, starting from a uniform distribution of the membrane species.

The simulations show that phase separation occurs within a finite range of GDI and GEF concentrations. The fact that, at low concentrations of both species, phase separation is not observed, and that it appears as their amount is gradually increased, is consistent with previous experimental observations obtained in vitro by Cezanne et al. (2020), and shows how a balance in the concentrations and activities of the various elements constituting the system is crucial for the emergence of this phenomenon.

This work therefore confirms the effectiveness of our model in exploring and reproducing self-organization phenomena on cellular membranes, and in predicting effects, such as the dependence on GDI and GEF, that have been observed experimentally. Moreover, it specifically shows that the parameter set considered, starting from the reference of Bezeljak et al. (2020), indeed leads to domain formation, in addition to the collective Rab activation, which was the subject of their study.

Table of contents

Chapter 1

Introduction

The origins of life, and of living matter itself, have always fascinated human beings. Before the rise of science, the only means of exploring such questions was human experience, expressed through images and words, not necessarily preceded by logical thoughts. As *Karl Kerényi* describes in his book "*Dionysos - Archetypal image of indestructible life*" [21], the ancient Greeks used two distinct words for *life*, preserved in parallel while having completely different forms and meanings: *bios* and *zoé*. The term *bios* referred to an individual, specific, finite life of a particular being, a *characterized life*. In contrast, the word *zoé* denoted life in a more absolute sense: with no limits or boundaries, without further characterizations. It's the life that continues no matter what, does not admit an end in its definition and is perceived as such by humans: as an *infinite life*. Indeed, in Greek *zoé* stands against *thanatos*, excluding it. For these reasons, the definition that best captures the meaning of *zoé* is "non-death".

Because of these considerations, Kerényi pointed out that the life dealt with by modern biology is not actually the *bios*, but rather the *zoé*. He also reported a minimal definition for life as "assimilation and inheritance", two defining features of life, against inert matter. Inheritance, in particular, implies that life overcomes the singular individual, proving itself to be *indestructible* in any specific case and persisting through time. This indestructible, infinite character of life is embodied in the figure of Dionysos, the Greek God who is not only associated with a temporary inebriation, but more profoundly with this ever-present drive towards vitality, what some have called "divine creative madness", the irrational foundation of the world. This indestructible nature of Dionysos is also evident from many mythical images and rituals associated with him, as deeply illustrated by Kerényi in the cited book [21].

From a biological perspective, we might identify this restless force that characterizes life and that leads matter to organize itself, with the "struggle against disorder" of which *E.Schrödinger* talks about in *What is Life?* [26]. Indeed, living organisms have evolved developing strategies to counteract the homogenizing effect of thermal disorder. Above all these strategies there are the "inventions" of macromolecules and the cell itself, basic units of life, characterized by an extraordinary degree of spatial organization. On a smaller scale, inside cells, in order to organize all the biochemical reactions in space, cells have created compartments, or *organelles*, which are distinct chemical environments characterized by two important properties: they have boundaries that separate them from their surroundings, and their internal components can diffuse freely to enable chemical reactions. Each organelle is enriched in specific molecules, making it functionally distinct from others. Such compartments may be separated by membranes (*membrane-bound compartments*, like organelles) or not (*non-membrane-bound compartments*) [19].

Behind this complex process of compartmentalization of the eukaryotic cell in a multiplicity of specialized subcellular systems, a process has emerged during the last decades as an overarching organizing principle: *phase separation*. Indeed, the formation of these inner structures often follows sequences of symmetry breaking events. The original spatial symmetry of the cell must be broken to adapt to a highly structured anisotropic environment, needed for life. Physically, these spatial organization phenomena resemble self-organized phase transitions, where the cell, either spontaneously or because it is driven by an external field, evolves into a state characterized by multiple coexisting chemical phases. These phases are spatially localized in different regions, establishing fronts and rears, tops and bottoms, outer and inner parts.

We can see *symmetry breaking* at work in several biological processes. For instance, stem cells divide asymmetrically to generate differentiated daughter cells, endowing them with specific fate determinants (Figure 1.1, top). Cells migrating in response to a chemical gradient break symmetry in order to develop a front and a rear, chemically distinct regions which allow directional movement (Figure 1.1, center). Epithelial cells develop chemically specialized apical, basal, and lateral regions to fit into the fine organized planar structure of an epithelial tissue (Figure 1.1, bottom). In all these examples, the plasma membrane is subdivided into specialized, chemically differentiated domains by a self-organized phase separation process. These polarized states are then maintained across cell generations by a complex machinery of protein sorting known generally as *molecular sorting*.

Differently from purely chemical systems, biological systems employ active, energy-consuming, out-of-equilibrium processes to sustain the compartments' heterogeneity. These active processes allow the existence of a much wider variety of stationary states than those available in close-to-equilibrium conditions. In summary, they play a key role in creating and sustaining the compartmentalized state necessary for the cell to perform its complex, vital functions [14].

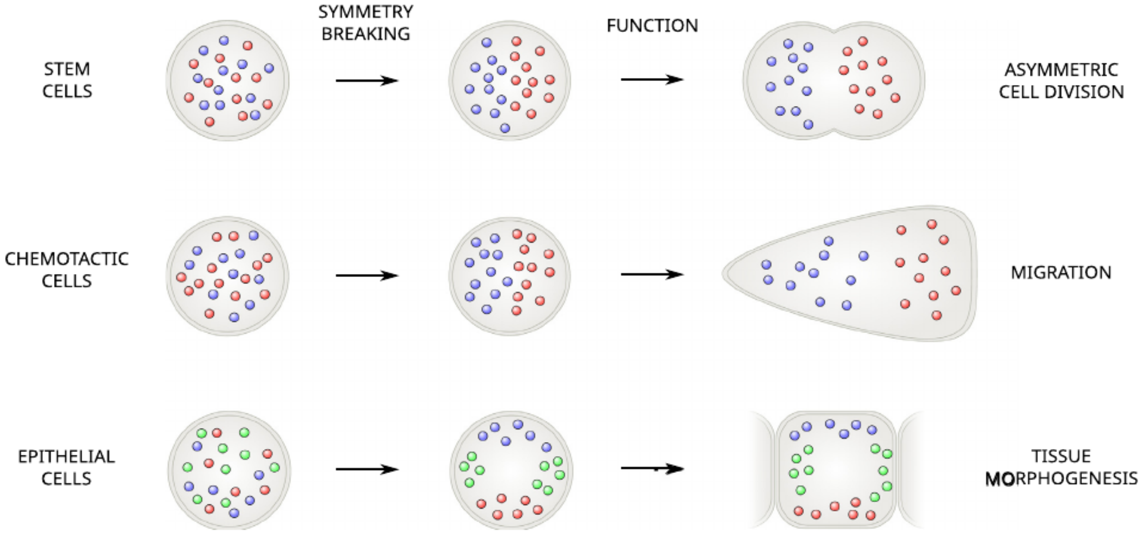


Figure 1.1. Symmetry breaking and generation of cellular functions. The image is taken from [14], Figure 1.

As we said, the spatio-temporal regulation of many intracellular processes depends on the emergence of patterns of specific proteins. The formation of patterns generated by interactions of diffusing proteins is a paradigmatic example of *self-organization*, which, in biological systems, refers precisely to the emergence of spatial and temporal structures. Therefore, mass-conserving reaction-diffusion equations provide the most appropriate mathematical framework to study these phenomena. They allow us to model and understand how spatial structures emerge within cells, leading to the conclusion that directed transport, such as diffusion along actively maintained gradients, is essential for pattern formation. In other words, the basic principle behind biological self-organization is the generation and maintenance of directed transport by intracellular protein dynamics [17].

These active phase separation processes can occur both in the cytosol and on the membrane, including the plasma membrane and the membrane of internal organelles. In this thesis, we focus our attention on membrane compartmentalization, a critical phenomenon for a variety of biological functions at multiple scales, from sub-cellular structures to multi-cellular organisms. In particular, we focus on the membrane compartmentalization occurring on cytoplasmic organelles, like early endosomes (EE), in which a protein named Rab5 exists in domains where it regulates vesicle tethering and fusion [6]. Our final goal is to use a computational model to replicate the experimentally observed formation of Rab5 patterns on endosomal membranes, shedding light on the mechanisms underlying this fundamental biological process.

Chapter 2

Small GTPases as regulators of intracellular dynamics: the case of Rab5

2.1 The role of small GTPases in intracellular communication

Intracellular organelles have a distinctive spatial distribution and communicate through a finely tuned system of vesicular and tubular transport. An interesting and fundamental question concerns the specificity of this process, i.e. how do these transport carriers, generated in one compartment, correctly recognize and fuse with their target organelle?

A typical transport reaction is generally divided into four steps: the formation of vesicles or tubules, their movement towards their target compartment, tethering/docking with the acceptor membrane, and finally the fusion of the lipid bilayers.

The specificity of membrane tethering and fusion is critical to preserve organelle identity and the proper flow of cargo within the cell [30]. The first step of tethering is mediated by Rab GTPases, which recognizes the incoming vesicle and ensures the attachment to the appropriate acceptor membrane. Then, the combined action of SNARE complexes (soluble NSF attachment protein receptors), Rab GTPases and their effectors guarantees precision in the fusion process. SNAREs are present across all organelles and they contribute to the specificity of the fusion process and to identifying the correct target. An additional layer of fidelity in the process is provided by Rab GTPases and their effectors, which directly regulate SNARE activity [30].

Moreover, other observations suggest that Rab GTPases are major determinants of organelle identity in eukaryotic cells. This is supported by the remarkable heterogeneity of Rab effectors (63 family members are estimated to exist in humans) and by their distribution in specific intracellular compartments. Here, they regulate transport between organelles, playing a fundamental role in cellular organization and trafficking [30].

Rab5, in particular, is associated with the plasma membrane and early endosomes, acting as a regulator of the kinetics of transport between the two membranes. It represents a regulatory factor in the early endocytic pathway [5]. Beyond mediating the transport of cargo from the plasma membrane to early endosomes, Rab5 also regulates homotypic fusion between early endosomes, a critical process for preserving the identity of early endosomes and ensuring their proper maturation [5]. This maturation process is then coordinated by other Rab proteins, such as Rab7, which guide endosomes through their transition towards lysosomal systems [3].

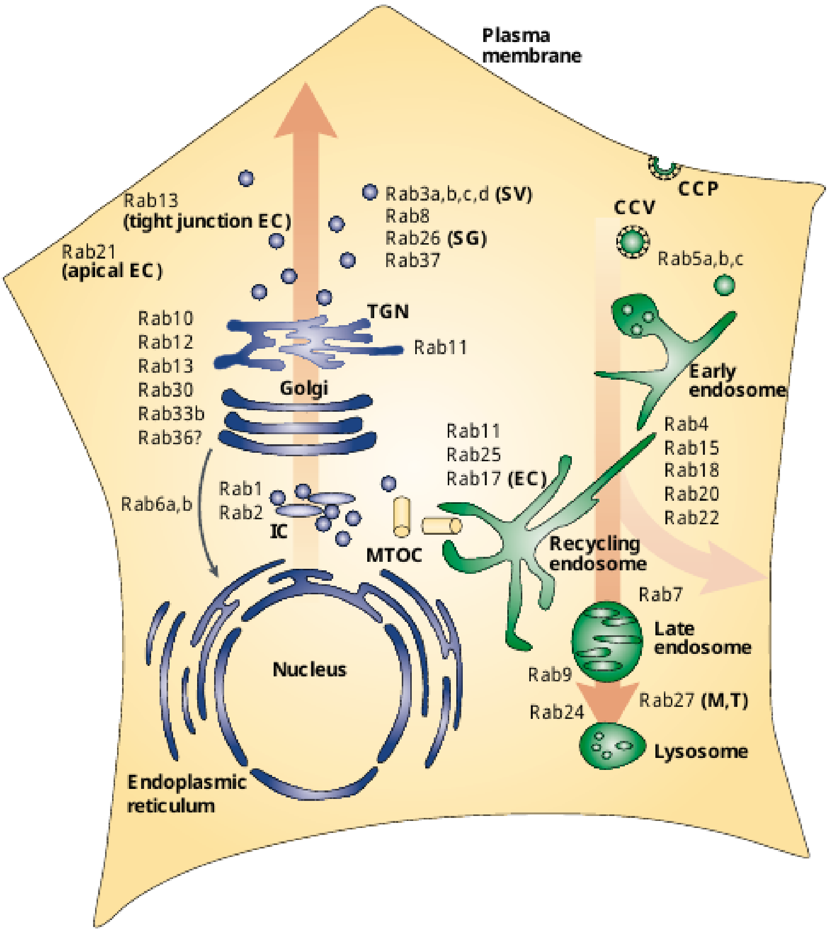


Figure 2.1. Map of intracellular localization of Rab proteins. The image is taken from the work by *Zerial et al.*(2001) [30], Figure 1. It summarizes the intracellular localization of Rab proteins in mammalian cells. Some proteins are cell-specific, like Rab3a in neurons, or show cell-type-specific localization, like Rab13 in tight junctions.

2.2 The switching behavior of small GTPases

The regulatory principle of Rab proteins, like other GTPases, is strongly related to their behavior as molecular switches [30]. They alternate between GTP- and GDP-bound conformations. Specifically, they are *inactive* when bound to GDP (guanosine diphosphate) and become active upon binding to GTP (guanosine triphosphate).

Because of their high affinity for GDP, these proteins require the assistance of *guanine nucleotide exchange factors* (GEFs) that promote the release of GDP to allow GTP binding and activation. In contrast, *GTP-ase activating proteins* (GAPs) stimulate GTP hydrolysis, switching the protein back to its inactive state. The combined action of GEFs and GAPs constitutes the core of this molecular switch [8].

Many small GTPases, such as Rab proteins, are post-translationally modified with hydrophobic prenyl groups (most commonly farnesyl or geranylgeranyl chains) at their C-terminus, ensuring the attachment of their active form to endomembranes. For this kind of protein, the nucleotide switching (GDP/GTP) is coupled with membrane association/dissociation; this transition between membrane-bound and cytosolic states is mediated by *guanine dissociation inhibitors* (GDIs). These proteins bind to the highly hydrophobic prenyl group of Rab proteins, shielding them from the aqueous cytosol and maintaining them in a soluble, inactive form.

Therefore, GDIs facilitate a controlled extraction of Rab proteins from membranes and their redistribution in other membrane regions, contributing to both spatial and functional regulation [8].

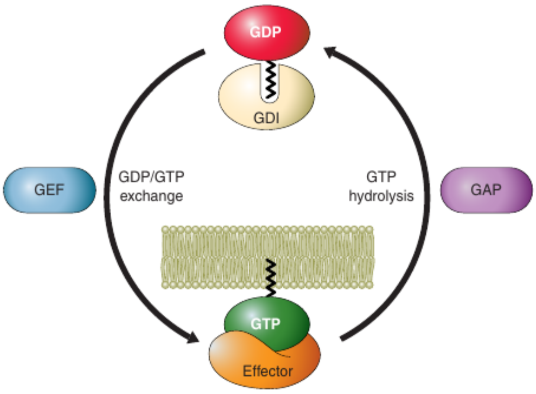


Figure 2.2. Regulation of the switching between GDP- and GTP-bound forms by GEFs, GAPs, and GDIs. All small GTPases are activated by GDP/GTP exchange stimulated by GEFs (blue) and inactivated by GTP hydrolysis stimulated by GAPs (violet). Several GTPase families also combine their GDP/GTP switch with a cytosol/membrane cycling regulated by GDIs or GDI-like proteins (beige). Each small GTPase subfamily is regulated by specific GEF and GAP subfamilies.

2.3 Positive feedback in the recruitment of active Rab5 to membranes

Positive-feedback loops are widely recognized as key mechanisms for amplifying molecular signals within cells. They occur whenever the product of a reaction promotes its own production; in this case, the active form of Rab5 promotes further activation by recruiting its GEF, thereby amplifying an initial burst of activation [8]. Specifically, Rab5, in its GTP-bound form, can bind to its effector Rabaptin5, which is associated with the GEF Rabex5, thereby localizing the GDP/GTP exchange in its immediate vicinity. In this way, Rab5 establishes a spatially localized positive feedback loop for its own activation [6].

This mechanism is particularly important on early endosomes, where Rab5-GTP is inherently unstable and undergoes continuous cycling between its active and inactive forms by the action of GAPs, such as RN-Tre for Rab5, and GEFs (see Figure 2.2) [30]. Because of this intrinsic instability, active Rab5 needs to recruit Rabaptin5 very rapidly in order to maintain sufficient levels of activation on the membrane. The formation of a ternary complex (Rab5/Rabex5/Rabaptin5) is essential for the creation of the positive feedback loop [6], which not only amplifies the GTPase activation but also ensures spatial and temporal precision, crucial for the efficiency and fidelity of endosomal docking and fusion processes [18].

Recent evidence has further suggested that during nucleotide exchange, Rab5 might be directly handed over from Rabex5 to Rabaptin5, providing a structural mechanism underlying the positive feedback loop [6]. Interestingly, the prevalence of GEF/effector coupling in small GTPase systems suggests that this may be a general mechanism for symmetry breaking and spatial organization of GTPases in cellular membranes. Indeed, self-organizing systems that are able to form patterns on their membranes often exhibit such non-linear dynamics of membrane recruitment and activation [6].

2.4 Membrane lipid composition influences Rab5 domain formation

Recent studies have shed light on the importance of membrane lipid composition for the processes of Rab5 recruitment and domain formation. Most small GTPases are post-translationally modified by lipid chains that allow them to associate with membranes. This lipid anchor, typically branched isoprenyl groups like geranylgeranyl or farnesyl chains, is covalently attached to a C-terminal hypervariable region (HVR), which has a disordered structure and is thus highly flexible. The HVR is the linker between the catalytic GTPase domain (G domain) and the membrane anchor. Molecular dynamics (MD) simulations reveal a significant accumulation of cholesterol (CH) and PI(3)P in close proximity to the peptide [12]. Both lipids appear to increase membrane lipid ordering, stabilize membrane anchoring, slow down protein diffusion, and probably create docking platforms for Rab5-induced signaling in early endosomes (EEs). These findings suggest that protein-lipid interactions are a central factor in the accumulation of Rab5 in PI(3)P microdomains [12].

A deeper insight into the lipid contributions to Rab5 recruitment and domain formation comes from the work of *Cezanne et al.*(2020) [6], where they employ in vitro biochemical reconstitution systems. First of all, they observe that PI(3)P and cholesterol enhance the membrane recruitment of Rab5, further confirming the importance of these two lipids in these phenomena (from Figure 2.3(A) to Figure 2.3(D)). Moreover, they argue that since PI(3)P is mainly produced by the activity of the class II PI3K complex, Vps34/Vps15, which is regulated by a direct interaction between Rab5 and Vps15, it seems reasonable to assert that Rab5 directly modifies the local lipid environment to stabilize itself on the membrane, thus providing yet another level of positive feedback in vivo.

They also observe that domain formation is lipid composition dependent. They observe that simple, highly diffusive, PC/PS membranes do not support domain formation (Figure 2.3A and 2.3C). This can be achieved only by adding cholesterol, which increases the complexity of the lipid membrane, but the domains have low GFP signal intensity. The same GFP-Rab5 signal intensity, as observed with the full EE lipid composition, is reached by adding sphingomyelin (and not ethanolamine plasmalogen) (Figure 2.3). Both sphingomyelin and ethanolamine plasmalogen increase the rigidity of the membrane, reducing the diffusivity of the membrane. This leads them to conclude that it is the presence of saturated acyl chains and the capacity for lateral lipid packing that enables domain formation rather than a simple reduction in global membrane diffusivity [6].

Sphingomyelins (SM) are important lipids of eukaryotic cellular membranes and neuronal tissues; together with cholesterol, SM molecules have been observed to be enriched in ordered lateral membrane domains, ‘rafts’, which have been suggested to play an important role in a wide range of cellular processes such as membrane trafficking, protein sorting, and transcytosis [13][28]. The association with cholesterol is highly favored by the fact that CH prefers to interact with saturated phospholipids and, SM is often the only one fully saturated phospholipid in cell membranes [28].

In general, sphingolipids appear to have a rather strong tendency for self-aggregation and show evidence of domain formation. Moreover, studies and simulations have proven that lateral diffusion coefficients in SM bilayers are generally lower than those in similar PC bilayers [13].

These membrane microdomains have been studied extensively in many cellular systems, and have been implicated in GTPase signaling, as we pointed out above. The work of *Slaughter et al.*(2013) [27] also favors a model where targeted delivery of Cdc42, another small GTPase, through vesicle trafficking is accompanied by the deposition of lipid components, such as PS, which help to provide an initial Cdc42 diffusion trap. Further recruitment of the Cdc42 effectors reinforces the microdomains that slow Cdc42 diffusion through additional protein and lipid interactions. These results, in summary, predict a role for membrane microdomains in modulating the diffusion properties of the membrane [27].

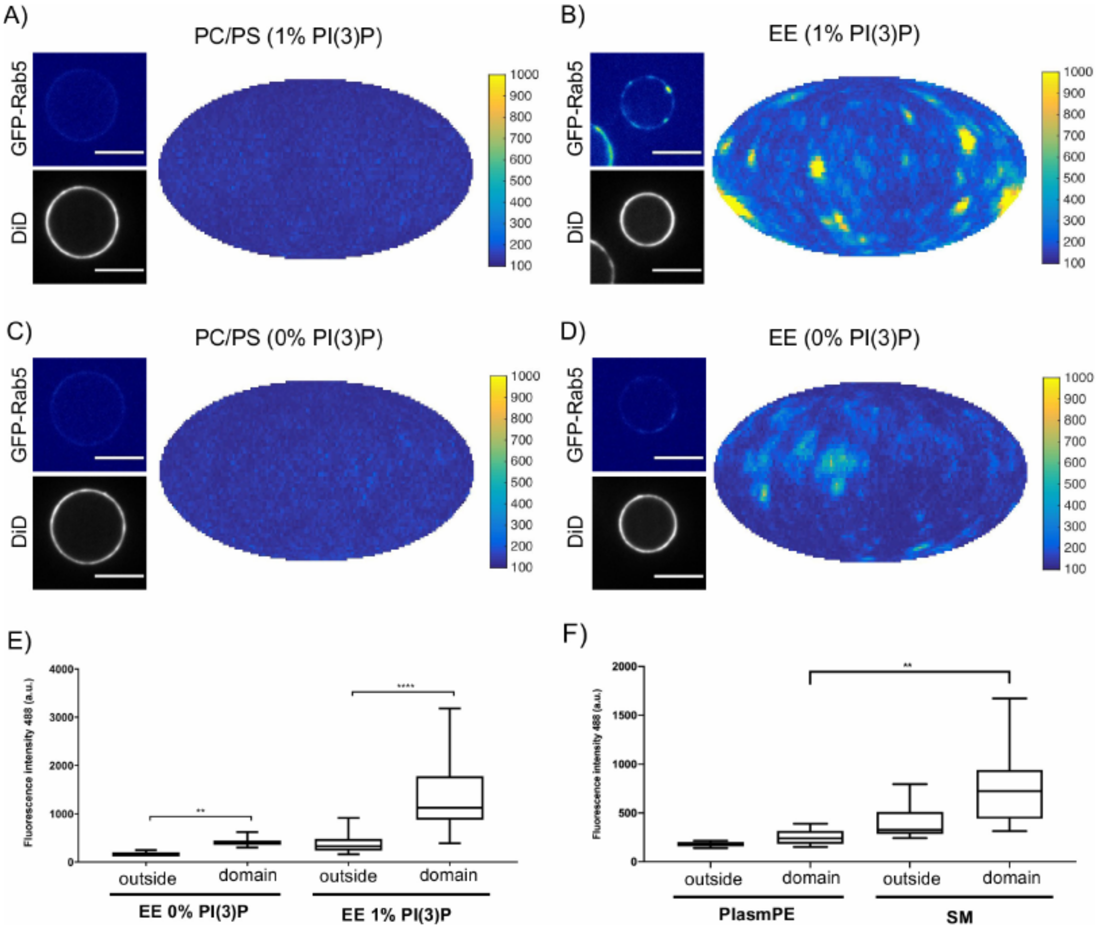


Figure 2.3. The figure is taken from the reference [6], Figure 6. Membrane-coated beads (MCBs) with PC/PS (where PC: phosphatidylcholine and PS: phosphatidylserine, and they represent typical component of generic cell membranes) and EE lipid composition containing 1 mol% PI(3)P (A) and B) respectively) and MCBs with PC/PS and EE lipid composition containing 0 mol% PI(3)P (C) and D) respectively) were incubated with 10 nM GFP-Rab5/GDI, 1 mM GDI, 100 nM Rabex5/Rabaptin5-RFP and 1 mM GTP for 15 min at 23°C. (E) Mean GFP-Rab5 signal intensity outside of and within segmented domains in B) and D). (F) Mean GFP-Rab5 signal intensity outside of and within segmented domains on MCBs with PC/PS/CH/PlasmPE and PC/PS/CH/SM lipid composition containing 1 mol% PI(3)P. In fact, the simple PC/PS lipid composition is sequentially modified to include the three most abundant lipids in the EE lipid composition: cholesterol (CH; 32.3 mol%), sphingomyelin (SM; 12.6 mol%) and ethanolamine plasmalogen (PlasmPE; 12.9 mol%).

Chapter 3

Analytical Model

From the point of view of physicists, this kind of phenomena can be analyzed through the concepts of phase separation and symmetry breaking, which provide us the appropriate framework to build a model that describes the underlying physics behind these fascinating cellular processes.

The main ingredients behind the spontaneous formation of localized enriched domains (symmetry breaking) are:

- reinforcing enzymatic feedback loops
- a finite cytosolic pool

The cytosolic pool is assumed to be finite because from experimental observation we see that this process of domain formation reaches saturation at certain time. Otherwise, the more stable phase would continue to grow and eventually cover the whole membrane. Indeed, the situation we are considering is phase separation occurring via nucleation, i.e. when a germ (droplet) of a stable phase is formed by thermal fluctuations in the "sea" of a metastable phase.

Now we are ready to formalize the (bio)physical picture based on these ideas. Let's consider a membrane of surface S in contact with a cytosolic pool as in Figure 3.1.

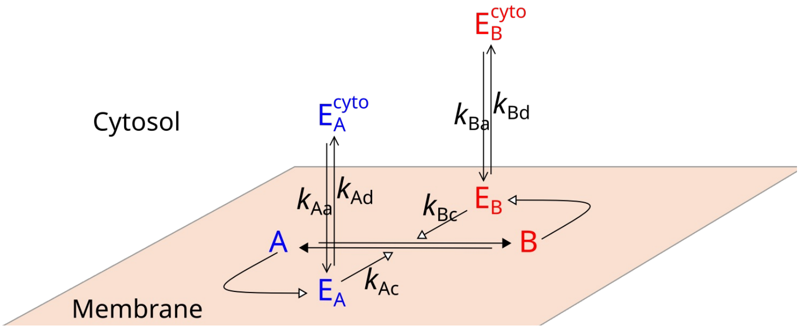


Figure 3.1. Representation of the model describing Rab5 activation on the membrane.

We consider a reaction-diffusion model in which membrane-bound molecules can be in two conformational states ϕ^-, ϕ^+ (A,B in the figure). The membrane is in contact with a cytosolic reservoir in which enzymes $E^-_{\text{cyto}}, E^+_{\text{cyto}}$ are free to diffuse and eventually anchor to the membrane-bound molecules ϕ^-, ϕ^+. Here, E^- catalyzes the conversion of ϕ^+ into ϕ^-, vice versa E^+ catalyzes the conversion of ϕ^- into ϕ^+. In summary we consider the following quantities:

$$\begin{aligned}
\phi^- &: \text{ surface concentration of } - \text{ molecules (e.g. Rab5:GDP)} \\
\phi^+ &: \text{ surface concentration of } + \text{ molecules (e.g. Rab5:GTP)} \\
E^-_{\text{cyto}} &: \text{ volume concentration field of } - \text{ enzymes} \\
E^+_{\text{cyto}} &: \text{ volume concentration field of } + \text{ enzymes} \\
E^- &: \text{ surface concentration of } - \text{ enzymes} \\
E^+ &: \text{ surface concentration of } + \text{ enzymes}
\end{aligned}$$

Now we have to turn this scheme into equations.

The enzymatic reactions are assumed to follow Michaelis-Menten kinetics, so that we can write the following equations for the production of ϕ^+, ϕ^- in time:

$$\begin{aligned}\frac{\partial\phi^+(\vec{r},t)}{\partial t} &= k_c^+ E^+ \frac{\phi^-}{K_M+\phi^-}\\ \frac{\partial\phi^-(\vec{r},t)}{\partial t} &= k_c^- E^- \frac{\phi^+}{K_M+\phi^+}\end{aligned}$$

We recall here that the Michaelis-Menten kinetics is obtained when the number of enzymes limits the production, in the sense that we have a small number of enzymes with respect to the number of substrate molecules ($\phi^\pm$in this case); and that the Michaelis-Menten constant is the concentration of the substrate for which we have half of the maximal rate of production, and it represents the change in behavior of the production $\frac{\partial\phi^\pm(\vec{r},t)}{\partial t}$ from linear in the concentration of the substrate $\phi^\mp$ to a regime of saturation, in which the rate of production becomes constant because we have too much substrate with respect to the enzymes.

Actually, we have to consider both processes to describe the full variation of the $\phi^\pm$ molecules. Then, using mass-action kinetics we write:

$$\begin{aligned}\frac{\partial\phi^+(\vec{r},t)}{\partial t} &= k_c^+ E^+ \frac{\phi^-}{K_M+\phi^-} - k_c^- E^- \frac{\phi^+}{K_M+\phi^+} = g(\phi^\pm, E^\pm)\\ \frac{\partial\phi^-(\vec{r},t)}{\partial t} &= k_c^- E^- \frac{\phi^+}{K_M+\phi^+} - k_c^+ E^+ \frac{\phi^-}{K_M+\phi^-} = -g(\phi^\pm, E^\pm)\end{aligned}$$

Another fundamental ingredient of our model is diffusion. Since cellular membranes are generally curved we should include diffusion using a Laplace-Beltrami operator in the following way:

$$\frac{\partial\phi^\pm(\vec{r},t)}{\partial t} = g(\phi^\pm, E^\pm) + D_m \Delta\phi^\pm$$

To complete the scheme, we also have to consider the variation in time of the enzymes, both in the cytosol and on the membrane. To do this we introduce the following quantity which represents the flux from cytosol to membrane of +/- enzymes. It is defined according to mass-action kinetics:

$$F^\pm = k_a^\pm E^\pm_{\text{cyto}}\phi^\pm - k_d^\pm E^\pm$$

The differential equation for enzymes are then:

$$\begin{aligned}\frac{\partial E^\pm}{\partial t} &= D_m \Delta E^\pm + F^\pm\\ \frac{\partial E^\pm_{\text{cyto}}}{\partial t} &= D_{\text{cyto}} \Delta E^\pm_{\text{cyto}} - F^\pm\end{aligned}$$

Observe that we can also add a boundary condition for the flux. In fact, the diffusive flux according to Fourier's law is:

$$F^\pm = \vec{F}^\pm \cdot (-\hat{n}) = D_{\text{cyto}} \vec{\nabla} E^\pm_{\text{cyto}} \cdot \hat{n}$$

where $\hat{n}$ is the inner normal to the membrane (directed to the cytosol), so that $F^\pm$ correctly represents the flux toward the membrane.

In summary, the equations describing the whole system are:

$$\begin{aligned}\frac{\partial\phi^\pm(\vec{r},t)}{\partial t} &= g(\phi^\pm, E^\pm) + D_m \Delta\phi^\pm - F^\pm\\ \frac{\partial E^\pm}{\partial t} &= D_m \Delta E^\pm + F^\pm\\ F^\pm &= k_a^\pm E^\pm_{\text{cyto}}\phi^\pm - k_d^\pm E^\pm\\ \frac{\partial E^\pm_{\text{cyto}}}{\partial t} &= D_{\text{cyto}} \Delta E^\pm_{\text{cyto}} - F^\pm\\ F^\pm &= D_{\text{cyto}} \vec{\nabla} E^\pm_{\text{cyto}} \cdot \hat{n}\end{aligned}$$

Let us make some assumptions to simplify the mathematical description:

1. Flat membrane (neglecting curvature effects). In this way the Laplace-Beltrami operator becomes simply a Laplacian ($\Delta \to \vec{\nabla}^2$).

2. $D_{\text{cyto}} \gg D_m$, because the membrane is a more viscous environment. For simplicity we can take the limit $D_{\text{cyto}} \to \infty$, so that we can consider $E^{\pm}_{\text{cyto}}$ as uniform fields (or simply as constants).
3. Quasi equilibrium of the cytosol/membrane exchange. This means that association/dissociation processes are faster than membrane redistribution (via diffusion) of $\phi^{\pm}, E^{\pm}$ molecules, so that we can consider $F^{\pm} = 0 \Longrightarrow k_a^{\pm} E^{\pm}_{\text{cyto}} \phi^{\pm} = k_d^{\pm} E^{\pm}$ and so we get:
$$E^{\pm} = \frac{k_a^{\pm} E^{\pm}_{\text{cyto}}}{k_d^{\pm}} \phi^{\pm} = \frac{E^{\pm}_{\text{cyto}}}{K_d^{\pm}} \phi^{\pm}$$
where we define $K_d^{\pm} = \frac{k_d^{\pm}}{k_a^{\pm}}$ which is the *dissociation constant*. The equation above tells us that the enzyme concentration fields $E^{\pm}$ on the membrane are "slaved" to the concentration field of their binding sites $\phi^{\pm}$.
4. The number of $\phi^{\pm}$ is much larger than the number of $E^{\pm}$ (as required also by Michaelis-Menten kinetics) so that we can neglect the flux terms $F^{\pm}$ in the equations for the concentration fields $\phi^{\pm}$.

Taking into account all of these assumptions, our system of equations becomes:

$$\frac{\partial \phi^+(\vec{r}, t)}{\partial t} = D_m \vec{\nabla}^2 \phi^+ + g(\phi^{\pm}, E^{\pm}) \tag{3.1}$$
$$\frac{\partial \phi^-(\vec{r}, t)}{\partial t} = D_m \vec{\nabla}^2 \phi^- - g(\phi^{\pm}, E^{\pm}) \tag{3.2}$$

If we sum up them:
$$\frac{\partial}{\partial t}(\phi^- + \phi^+) = D_m \vec{\nabla}^2 (\phi^- + \phi^+)$$
which has trivial, uniform steady states:
$$\phi^-(\vec{r}, t) + \phi^+(\vec{r}, t) = c$$
which tells us that in the model we are considering there is no destruction or creation of ϕ molecules, only mutual conversion of ϕ^+ and ϕ^- molecules. Hence, conservation law is respected. We can also subtract (3.2) from (3.1):
$$\frac{\partial}{\partial t}(\phi^+ - \phi^-) = D_m \vec{\nabla}^2 (\phi^+ - \phi^-) + 2g(\phi^{\pm}, E^{\pm})$$
and define $\phi(\vec{r}, t) = \phi^+(\vec{r}, t) - \phi^-(\vec{r}, t)$, which is a natural candidate to be our *order parameter field*. We also introduce $f(\phi) = 2g(\phi^{\pm}, E^{\pm})$ and we call it *catalytic force*. In conclusion, our system, under these simplifying assumption is described by:
$$\frac{\partial \phi}{\partial t} = D_m \vec{\nabla}^2 \phi + f(\phi) \tag{3.3}$$
which is a time-dependent Ginzburg-Landau equation.

This means that the formation of polarity domains on cell membranes can be described using the Ginzburg-Landau approach.

<u>Note</u> : Exploiting our assumptions we can rewrite the catalytic force
$$f(\phi) = 2k_c^+ \frac{E^+ \phi^-}{K_M + \phi^-} - 2k_c^- \frac{E^- \phi^+}{K_M + \phi^+}$$
Substituting $E^{\pm} = \frac{E^{\pm}_{\text{cyto}}}{K_d^{\pm}} \phi^{\pm}$ and $\phi^{\pm} = \frac{1}{2}(c \pm \phi)$ we get:
$$\begin{aligned} f(\phi) &= 2k_c^+ \frac{\frac{E^+_{\text{cyto}}}{K_d^+} \cdot \frac{\phi^2 - c^2}{4}}{K_M + \frac{c-\phi}{2}} - 2k_c^- \frac{\frac{E^-_{\text{cyto}}}{K_d^-} \cdot \frac{\phi^2 - c^2}{4}}{K_M + \frac{c+\phi}{2}} \\ &= (c^2 - \phi^2) \cdot \left[\frac{\alpha^+}{2K_M + c - \phi} - \frac{\alpha^-}{2K_M + c + \phi} \right] \end{aligned} \tag{3.4}$$

where we have defined $\alpha^+ = \frac{k_c^+ E_{\text{cyto}}^+}{K_d^+}$ and $\alpha^- = \frac{k_c^- E_{\text{cyto}}^-}{K_d^-}$.

Note : it can be seen that the $\alpha^\pm$ coefficients are, themselves, functionals of the field ϕ . Indeed, if we introduce the following notation for the spatial average over the membrane of a generic functional $h[\phi]$:

$$\langle h[\phi] \rangle = \frac{1}{S} \int_S h(\phi(\vec{r}, t))\, d\sigma$$

we can write

$$\langle E^\pm(\vec{r}, t) \rangle = \frac{E_{\text{cyto}}^\pm}{K_d^\pm} \langle \phi^\pm(\vec{r}, t) \rangle \tag{3.5}$$

Now, if we impose the conservation of the total number of enzymes:

$$\begin{aligned}
\frac{1}{S}\int_S E^\pm d\sigma + \frac{1}{S}\int_V E_{\text{cyto}}^\pm dv &= \frac{1}{S}\int_V E_{\text{tot}}^\pm dv = \text{const} \\
\langle E^\pm \rangle + \frac{V}{S} E_{\text{cyto}}^\pm &= \frac{V}{S} E_{\text{tot}}^\pm \\
\theta \langle E^\pm \rangle + E_{\text{cyto}}^\pm &= E_{\text{tot}}^\pm
\end{aligned}$$

where we have defined the *geometric factor* $\theta = \frac{S}{V}$. Substituting (3.3) we get:

$$\theta \frac{E_{\text{cyto}}^\pm}{K_d^\pm} \langle \phi^\pm(\vec{r}, t) \rangle + E_{\text{cyto}}^\pm = E_{\text{tot}}^\pm$$

and then

$$E_{\text{cyto}}^\pm = \frac{E_{\text{tot}}^\pm}{1 + \frac{\theta}{K_d^\pm} \langle \phi^\pm \rangle} \tag{3.6}$$

Finally, inserting (3.6) into the expressions of $\alpha^\pm$ coefficients and restoring $\phi^\pm = \frac{1}{2}(c \pm \phi)$

$$\alpha^\pm[\phi] = \frac{k_c^\pm E_{\text{cyto}}^\pm}{K_d^\pm} = \frac{\frac{k_c^\pm}{K_d^\pm} E_{\text{tot}}^\pm}{1 + \frac{\theta}{K_d^\pm} \frac{c \pm \langle \phi^\pm \rangle}{2}} \tag{3.7}$$

In summary, our *mass-conserved reaction-diffusion model* is represented by the following equations:

$$\begin{aligned}
\frac{\partial \phi}{\partial t} &= D_m \vec{\nabla}^2 \phi + f(\phi) \\
f(\phi) &= (c^2 - \phi^2) \cdot \left[\frac{\alpha^+}{2K_M + c - \phi} - \frac{\alpha^-}{2K_M + c + \phi} \right] \\
\alpha^\pm[\phi] &= \frac{k_c^\pm E_{\text{tot}}^\pm}{K_d^\pm + \frac{\theta}{2}(c \pm \langle \phi \rangle)}
\end{aligned}$$

Our model is characterized by a time-dependent Ginzburg-Landau equation containing a non-linear term for enzyme-driven conversion $f(\phi)$ and by an integral constraint which guarantees that the total number of enzymes shuttling between membrane and cytosol is conserved.

3.1 Uniform solutions

We want to study this model. The first thing we can do is to look for uniform solutions, i.e. $\phi(\vec{r}, t) = \phi(t)$ with constant $\alpha^\pm$. In this case the dynamic equation (3.3) becomes simply:

$$\frac{\partial \phi(t)}{\partial t} = f(\phi(t)) \tag{3.8}$$

The steady-state solutions correspond to the fixed points of $f(\phi) = 0$, so the equation I have to solve is:

$$(c^2 - \phi^2) \cdot \left[\frac{\alpha^+}{2K_M + c - \phi} - \frac{\alpha^-}{2K_M + c + \phi} \right] = 0$$

We can immediately notice that the equation has two trivial fixed points at $\phi = \pm c$. The third root comes from:

$$\begin{aligned}
\left[\frac{\alpha^+}{2K_M + c - \phi} - \frac{\alpha^-}{2K_M + c + \phi}\right] &= 0 \\
\frac{\alpha^+}{\alpha^-}(2K_M + c + \phi) &\geqslant 2K_M + c - \phi \\
\phi\left(\frac{\alpha^+}{\alpha^-} + 1\right) &\geqslant (2K_M + c)\left(1 - \frac{\alpha^+}{\alpha^-}\right)
\end{aligned}$$

Then, the third root is

$$\phi_0 = (2K_M + c)\frac{1 - \rho}{1 + \rho}$$

where we have defined the *catalytic ratio* ρ as:

$$\rho = \frac{\alpha^+}{\alpha^-} = \frac{k_c^+ E_{\text{cyto}}^+ K_d^-}{k_c^- E_{\text{cyto}}^- K_d^+}$$

Hence, the term in square brackets is positive when $\phi \geqslant \phi_0$ and negative otherwise. At this point, we are ready to study the sign of $f(\phi)$ in order to investigate the stability of our solutions. In fact, looking at equation (3.8), we see that the sign of $f(\phi)$ is directly related to the sign of $\frac{\partial \phi}{\partial t}$.

It is convenient to introduce the potential

$$V(\phi) = -\int f(\phi)\, d\phi$$

and the inverted potential

$$U(\phi) = -V(\phi)$$

Recall that $\phi = \phi^+ - \phi^-$, and that $\phi^+ \geqslant 0$, $\phi^- \geqslant 0$ because they represent physical concentrations. Finally, they must also satisy the conservation law $\phi^+ + \phi^- = c$. Putting all this together, we obtain a physical interval for the order parameter, that is:

$$-c \leqslant \phi \leqslant c$$

This tells us that the third root ϕ_0 is a physical solution only when it is inside this interval. In this case, studying of the sign of $f(\phi)$ leads to the following results:

	$[-\infty, -c]$	$[-c, \phi_0]$	$[\phi_0, +c]$	$[+c, +\infty]$
$c - \phi > 0$	$+$	$+$	$+$	$-$
$c + \phi > 0$	$-$	$+$	$+$	$+$
$\phi - \phi_0 > 0$	$-$	$-$	$+$	$+$
$f(\phi)$	$+$	$-$	$+$	$-$

Table 3.1. Sign of $f(\phi)$ in the case $-c \leqslant \phi_0 \leqslant c$.

Now we can provide a graphical representation of $f(\phi)$ and $V(\phi)$:

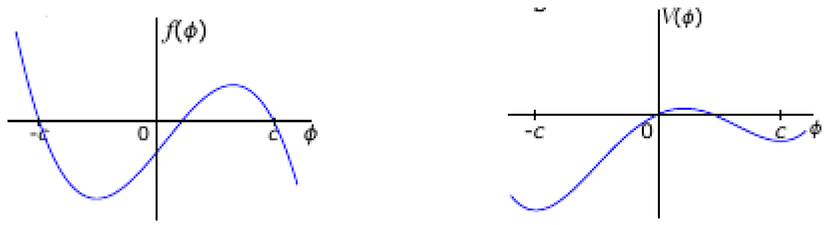


Figure 3.2. (left) Plot of $f(\phi)$ in case $-c \leqslant \phi_0 \leqslant c$ (ϕ_0 is the third root of $f(\phi)$=0). (right) The corresponding plot of $V(\phi)$. In this example, we can see that the negative phase ($\phi = -c$) is favored.

As we said previously, the sign of $f(\phi)$ is related to the stability of the solutions through equation (3.8). As a consequence, when $-c \leqslant \phi_0 \leqslant c$, there are two stable fixed points (steady-state solutions of the dynamical equation in the uniform case), namely $\phi = +c$, $\phi = -c$, and one unstable fixed point at $\phi = \phi_0$. With this kind of potential we have, therefore, *bistability* according to our model.

Note : $\phi = \phi^+ - \phi^-$, and recall that $\phi^+ + \phi^- = c$. Hence, the solution $\phi = +c$ means that $\phi^+ = c$ and $\phi^- = 0$: the entire lipid population is in the ϕ^+ conformation (e.g., all Rab5 are in the active state). Conversely, for $\phi = -c$ the lipid population is in the ϕ^- conformation. Note that both phases are stable. This means that, depending on the initial conditions, the system will evolve toward one of the two.

Although ϕ_0 is a physical solution only in this case, it can also take non-physical values. In those cases, the only physical steady-state solutions of the dynamical equation are $\phi = \pm c$, but only one of them is stable (*monostability*) In particular, when $\phi_0 < -c$, only $\phi = +c$ is a stable fixed point, while $\phi = -c$ is unstable. This means that the ϕ^+ phase is the only stable one. When $\phi_0 > c$, instead, we are instead in the opposite situation. Observe that the value of ϕ_0 depends on the catalytic ratio ρ and on the values of K_M and c. We can therefore construct a phase diagram. Let's find the values of ϕ_0 corresponding to each of these three regimes.

$$-c \leqslant \phi_0 \leqslant c$$

$$-c \leqslant (2K_M + c)\frac{1-\rho}{1+\rho} \leqslant c$$

Let us focus on the first inequation:

$$\begin{array}{rcl} -c(1+\rho) & \leqslant & (2K_M + c)(1-\rho) \\ -c & \leqslant & 2K_M - 2\rho K_M + c \\ \rho & \leqslant & \dfrac{K_M + c}{K_M} \end{array}$$

Introduce the dimensionless parameter $\kappa = \frac{K_M}{c}$:

$$\rho \leqslant \frac{1+\kappa}{\kappa}$$

Similarly, for the second inequality we find:

$$\rho \geqslant \frac{\kappa}{1+\kappa}$$

This means that the values of the catalytic ratio for which we have bistability are:

$$\frac{\kappa}{1+\kappa} \leqslant \rho \leqslant \frac{1+\kappa}{\kappa}$$

Outside this interval, the system is in a regime of monostability, as discussed previously. Summarizing, this is the phase diagram for the uniform case:

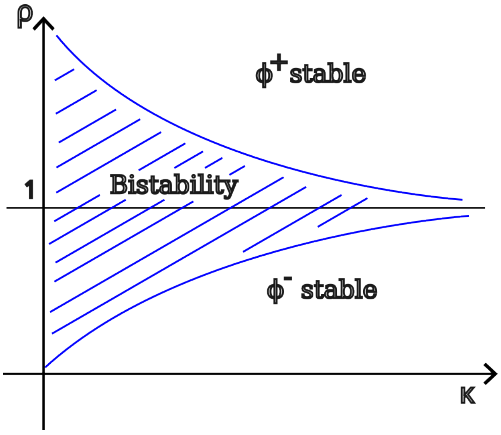


Figure 3.3. Phase diagram for uniform (steady-state) solutions.

3.2 The study of the non-uniform case

So far, we have looked only at uniform solutions. However, bistability implies the possibility of observing two different phases coexisting in different membrane regions, since they are both stable in some parameter region. This is the phenomenon we are interested in: the coexistence of domains with different chemical identities. In our case, these two phases are represented by the ϕ^+, ϕ^- phases. For this purpose, it is convenient to adopt a variational approach, that is, by re-expressing the equation in a variational form:

$$\frac{\partial\phi}{\partial t} = D\overrightarrow{\nabla}^2\phi - V'(\phi) = -\frac{\delta F[\phi]}{\delta\phi} \tag{3.9}$$

where $F[\phi]$ is the "effective" free energy functional (it can be shown assuming constants $\alpha^{\pm}$)

$$F[\phi] = \int\left[\frac{D}{2}(\nabla\phi)^2 + V(\phi)\right]dx_1\ldots dx_{\mathcal{D}}$$

where, for convenience, we consider a generic spatial dimension $\mathcal{D}$ (for a membrane, $\mathcal{D}=2$). It is important to observe that the system we are considering is out of equilibrium; that is why F is only an "effective" free energy. We cannot apply thermodynamics in this context.

Equation (3.9) tells us that the dynamics tend to minimize the effective energy functional F. We are interested in a particular situation, i.e., when one phase is more stable than the other, although both are stable. In particular, let us consider the case in which the ϕ^+ phase is more stable than the ϕ^- phase. In this case, the potential takes the following form:

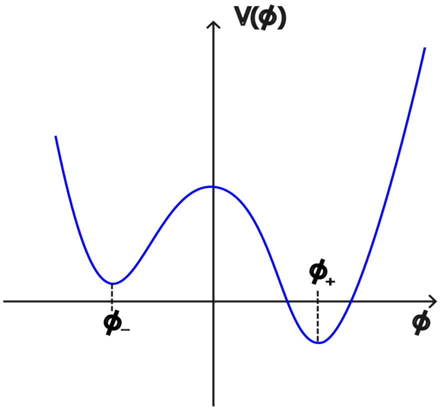


Figure 3.4. Asymmetric potential, the ϕ^+-phase is more stable than the ϕ^--phase.

We can define the *metastability degree*:

$$\Delta V = V(-c) - V(+c)$$

With this kind of potential, the system can also remain for a long time in the less stable phase. The question we want to answer here is: what happens if a "germ" of the more stable phase is created, through thermal fluctuations, within the "sea" of the metastable state? Theoretically, this process is known as *nucleation*. In order to gain some intuition about the process, we start by studying a very simple one-dimensional (1D) case.

3.3 Phase propagation in 1D

Let us consider a situation in which an infinite system is in the ϕ^+ phase on the left and in the ϕ^- phase on the right. In one dimension, my dynamical equation is:

$$\frac{\partial\phi}{\partial t} = D\frac{\partial^2\phi}{\partial x^2} + f(\phi) \tag{3.10}$$

This equation allows for *propagating front* solutions:

$$\phi(x,t) = \varphi(x - \mathrm{vt})$$

In this case, $\frac{\partial\phi}{\partial t} = -v\varphi'$ and (3.10) becomes:

$$-v\varphi'(x-\mathrm{vt}) = D\varphi''(x-\mathrm{vt}) + f(\varphi(x-\mathrm{vt}))$$

By calling $\xi = x - \mathrm{vt}$ and recalling that $f(\phi) = -V'(\phi) = U'(\phi)$:

$$D\frac{d^2\varphi}{d\xi^2} = -v\frac{d\varphi}{d\xi} - U'(\varphi) \tag{3.11}$$

At this point, we can observe that this is equivalent to Newton's second law for a fictitious particle moving in the inverted potential U', with a viscous friction coefficient v. Then, by exploiting this mechanical analogy, we identify:

$$\begin{aligned} \varphi &: \text{position of the fictitious particle} \\ \xi &: \text{fictitious time} \\ v &: \text{coefficient of viscous friction} \\ D &: \text{mass} \end{aligned}$$

We are looking for solutions where:

$$\begin{aligned} \varphi(\xi = -\infty) &= +c \\ \varphi(\xi = +\infty) &= -c \end{aligned}$$

so that I can replicate the pattern I want, that is, an infinite system in the $\phi^+ = +c$ phase on the left and in the ϕ^- phase on the right. According to our mechanical analogy, we need a particular value $v = v_*$ with which my fictitious particle is able to reach the other stable position (a phase for us) $\varphi = -c$ at infinite time $\xi = +\infty$.

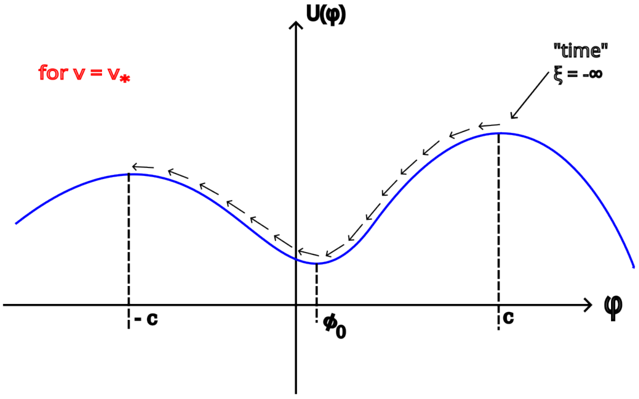


Figure 3.5. Inverted potential $U(\varphi)$. When $v = v_*$ we are able to reach $\varphi = -c$ at infinite time $\xi = +\infty$.

Observe that in my analogy v is the velocity of the propagation front. Then, for $v = v_*$ we get the unique solution ϕ :

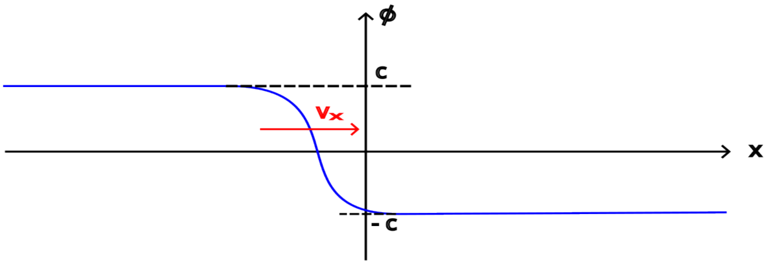


Figure 3.6. Graphical representation of ϕ with propagation front velocity $v = v_*$.

Now we can compute the propagation velocity from (3.11), imposing $v = v_*$:

$$\begin{aligned} D\frac{d^2\varphi}{d\xi^2} &= -v_*\frac{d\varphi}{d\xi} - U'(\varphi) \\ D\frac{d^2\varphi}{d\xi^2}\frac{d\varphi}{d\xi} &= -v_*\left(\frac{d\varphi}{d\xi}\right)^2 - U'(\varphi)\frac{d\varphi}{d\xi} \end{aligned}$$

Integrating ξ in $[-\infty,+\infty]$:

$$\begin{aligned}\int_{-\infty}^{+\infty}\left[D\frac{d^2\varphi}{d\xi^2}\frac{d\varphi}{d\xi}+U'(\varphi)\frac{d\varphi}{d\xi}\right]d\xi &= -\int_{-\infty}^{+\infty}v_*\left(\frac{d\varphi}{d\xi}\right)^2 d\xi\\ \int_{-\infty}^{+\infty}\frac{d}{d\xi}\left[\frac{D}{2}\left(\frac{d\varphi}{d\xi}\right)^2+U(\varphi)\right]d\xi &= -\int_{-\infty}^{+\infty}v_*\left(\frac{d\varphi}{d\xi}\right)^2 d\xi\end{aligned}$$

where, continuing the mechanical analogy, we can identify the first two terms as the "kinetic energy" and the "potential energy", while the term on the r.h.s. represents the energy dissipation. We can go further in the equations above,

$$v_*\int_{-\infty}^{+\infty}\left(\frac{d\varphi}{d\xi}\right)^2 d\xi = -\left[\frac{D}{2}\left(\frac{d\varphi}{d\xi}\right)^2+U(\varphi)\right]_{-\infty}^{+\infty}$$

At this point, imposing the initial conditions (at $\xi=\pm\infty$)

$$\frac{d\varphi}{d\xi}(-\infty)=0 \qquad \frac{d\varphi}{d\xi}(-\infty)=0 \qquad \varphi(-\infty)=+c \qquad \varphi(+\infty)=-c$$

we get:

$$\begin{aligned}v_*\int_{-\infty}^{+\infty}\left(\frac{d\varphi}{d\xi}\right)^2 d\xi &= -U(\varphi(+\infty))+U(\varphi(-\infty))\\ &= -U(-c)+U(+c)\\ &= V(-c)-V(+c)\\ &= \Delta V\end{aligned}$$

Then, the velocity of the propagating front is proportional to the metastability degree, and, in this mechanical analogy in 1D, they are related by the following formula:

$$v_*=\frac{\Delta V}{\int_{-\infty}^{+\infty}\left(\frac{d\varphi}{d\xi}\right)^2 d\xi}=\frac{\Delta V}{\gamma} \tag{3.12}$$

where we call γ the integral representing the energy dissipation.

3.4 Phase coexistence in 1D

We want to study in detail the case of a stationary front/interface. This means that $v_*=0$ and, as a consequence, $\Delta V=0$. Then, both phases are equally favored by the dynamics. In this case, we also have that $\xi=x-v_*t=x$ and equation (3.10) becomes, at stationarity:

$$\frac{\partial\phi}{\partial t}=D\frac{\partial^2\phi}{\partial x^2}-V'(\phi)=0$$

$$D\frac{d^2\phi}{dx^2}-V'(\phi)=0$$

Now we integrate over ξ (in this case equivalent to integrating over x) and we get:

$$\frac{D}{2}\left(\frac{d\phi}{dx}\right)^2-V(\phi)=\text{const}=0$$

where we impose const $=0$ because we have integrated over the whole space $\xi=x$ and we know that at the two extremes of integration, our argument is zero. Indeed, both $\frac{d\phi}{dx}$ and $V(\phi)$ are zero at $\xi=\pm\infty$. Then,

$$\begin{aligned}\frac{D}{2}\left(\frac{d\phi}{dx}\right)^2 &= V(\phi)\\ \frac{d\phi}{dx} &= \pm\sqrt{\frac{2}{D}V(\phi)}\\ \int\frac{d\phi}{\sqrt{V(\phi)}} &= \pm\sqrt{\frac{2}{D}}\int dx\end{aligned} \tag{3.13}$$

where in the last step we have separated variables and integrated both members. The 1D case for a stationary front is then solved by equation (3.13).

Solving the equation with the potential of my polarization model (3.4) is rather complex. Then, we adopt a simplified potential, the so-called ϕ^4-*potential*, i.e.

$$V(\phi) = (1-\phi^2)^2 \tag{3.14}$$

In this case, equation (3.13) becomes:

$$\begin{aligned}\int \frac{d\phi}{\sqrt{(1-\phi^2)^2}} &= \pm\sqrt{\frac{2}{D}}\,x + \text{const}\\ \int \frac{d\phi}{1-\phi^2} &= \pm\sqrt{\frac{2}{D}}\,x + \text{const}\\ \text{arctanh}(\phi) &= \pm\sqrt{\frac{2}{D}}\,x + \text{const}\\ \phi(x) &= \pm\tanh\left[\sqrt{\frac{2}{D}}\,(x-x_0)\right]\end{aligned}$$

Since $\phi(-\infty) = c = 1$ we choose the $-$ sign, then:

$$\phi(x) = -\tanh\left[\sqrt{\frac{2}{D}}\,(x-x_0)\right] \tag{3.15}$$

From this solution we see that the thickness of the interface is proportional to the square root of the diffusivity:

$$w = \sqrt{\frac{D}{2}}$$

This is valid at steady state, but likely also for fronts propagating with small enough velocity.

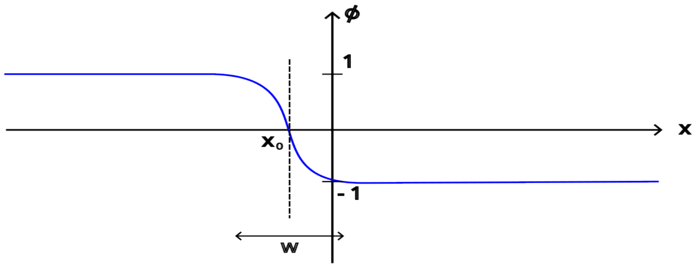


Figure 3.7. $\phi(x)$ in case of phase coexistence and with a ϕ^4-potential.

<u>*Note*</u> : $[D] = \frac{L^2}{T}$. This means that (3.15) does not have the correct physical dimension, the argument of an hyperbolic tangent has to be non-dimensional. To take into account the correct physical dimension it is convenient to rewrite our catalytic force as:

$$f(\phi) = k\,c\,\tilde{f}\left(\frac{\phi}{c}\right) \tag{3.16}$$

where $[k] = T^{-1}$ and $\tilde{f}\left(\frac{\phi}{c}\right)$ is the dimensionless catalytic force. Substituting (3.16) into (3.10), we get:

$$\begin{aligned}\frac{\partial\phi}{\partial t} &= D\frac{\partial^2\phi}{\partial x^2} + k\,c\,\tilde{f}\left(\frac{\phi}{c}\right)\\ \frac{\partial}{\partial(kt)}\frac{\phi}{c} &= \frac{D}{k}\frac{\partial^2}{\partial x^2}\frac{\phi}{c} + \tilde{f}\left(\frac{\phi}{c}\right)\end{aligned}$$

Using the ϕ^4-*potential* we get that:

$$f(\phi) = -V'(\phi) = 4(1-\phi^2)\phi$$

and by restoring the correct physical dimensions:

$$f\left(\tilde{\phi}=\frac{\phi}{c}\right)=4kc^2(1-\tilde{\phi}^2)\tilde{\phi} \tag{3.17}$$

Finally, the solutions become:

$$\phi(x)=-c\tanh\left(\frac{x-x_0}{\sqrt{\frac{D}{2k}}}\right) \tag{3.18}$$

where we can see that now my interface width has the correct physical dimension of a length: $\left[\sqrt{\frac{D}{2k}}\right]=L$.

Of course, our goal is to find the solutions of the equation with the much more complex potential of the polarization model, i.e.

$$f(\phi)=(c^2-\phi^2)\cdot\left[\frac{\alpha^+}{2K_M+c+\phi}-\frac{\alpha^-}{2K_M+c-\phi}\right]$$

We cannot solve the equation analytically, even in one dimension, but we can use some strategies to generalize the simple case of a ϕ^4-*potential* to the more complex case of my catalytic force. One way is to Taylor-expand $f(\phi)$ for large $\kappa=\frac{K_M}{c}$, considering $\alpha^+=\alpha^-=\bar{\alpha}$:

$$\begin{aligned}
f(\phi) &\simeq (c^2-\phi^2)\left[\frac{\bar{\alpha}}{2K_M+c}\left(1+\frac{\phi}{2K_M+c}\right)-\frac{\bar{\alpha}}{2K_M+c}\left(1-\frac{\phi}{2K_M+c}\right)\right]\\
&= (c^2-\phi^2)\frac{2\bar{\alpha}\phi}{(2K_M+c)^2}\\
&= \frac{2c^2\,\bar{\alpha}}{(2\kappa+1)^2}(1-\tilde{\phi}^2)\tilde{\phi}
\end{aligned}$$

where $\tilde{\phi}=\frac{\phi}{c}$. If we compare the expression above with (3.17) we see that they are equal if

$$k=\frac{1}{2}\frac{\bar{\alpha}}{(2\kappa+1)^2} \tag{3.19}$$

This means that in this limit of large κ (and with $\alpha^+=\alpha^-=\bar{\alpha}$), the force (and thus the potential) for membrane polarity reduces to a ϕ^4-*potential* with this value of k. Now we can substitute (3.19) in (3.18) and obtain:

$$\phi(x)=-c\tanh\left[\sqrt{\frac{1}{D}\left(\frac{\bar{\alpha}}{(2\kappa+1)^2}\right)}(x-x_0)\right]=-c\tanh\left(\frac{x-x_0}{w}\right) \tag{3.20}$$

This is the solution of my full polarization model in 1D and in the limit of a large κ and equal $\alpha^\pm$ coefficients. We can also consider that they are not exactly equal and write:

$$w=(2\kappa+1)\sqrt{\frac{2D}{\alpha^++\alpha^-}} \tag{3.21}$$

According to (3.21), the interface width depends not only on diffusivity, but also on the catalytic activity. In fact, we recall here the expressions for our $\alpha^\pm$ coefficients:

$$\alpha^\pm=\frac{k_c^\pm E_{\text{cyto}}^\pm}{K_d^\pm} \tag{3.22}$$

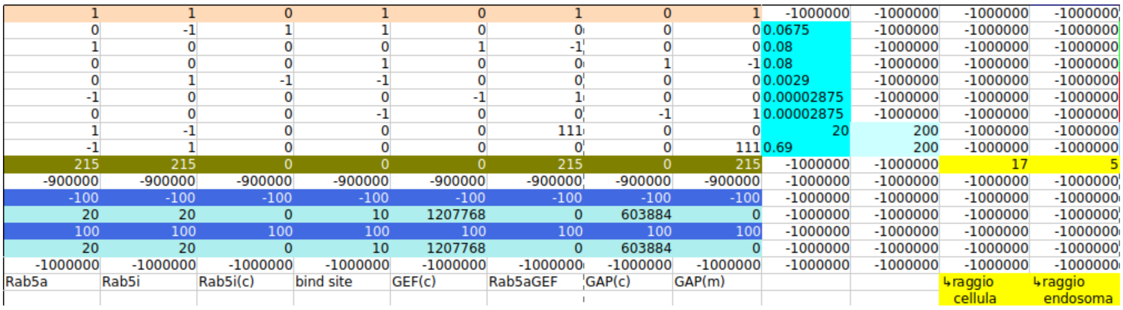
Chapter 4
Computational model

As discussed in the previous chapter, the main paradigm for the study of phase separation phenomena in biology is a mass-conserved reaction-diffusion model. To this purpose, we adopt a Gillespie algorithm, a common choice to simulate stochastic processes in continuous time. Our goal is to show that, starting from a reaction scheme based on our biological knowledge about Rab5 proteins and an initially uniform distribution of the reactants, the system will exhibit phase separation (i.e., domain formation) within specific regions of the parameter space.

The algorithm reads a file containing all the necessary inputs: reactants (local or global), reactions, rate values, populations of reactants, diffusion coefficients, the change in populations determined by each reaction, and the radii of the cell and of the endosome (Figure 4.1). Then, it classifies the reactions in input. They can be:

- dissociation reactions
- association reactions
- catalytic reactions
- diffusion reactions

In principle, the algorithm can handle also other reactions, like creation/destruction and mutations. However, in our case, we assume mass conservation as discussed in Chapter 3, and we also assume that mutations occur only via activation/deactivation reactions, i.e., via catalytic reactions.



1	1	0	1	0	1	0	1	-1000000	-1000000	-1000000	-1000000
0	-1	1	1	0	0	0	0	0.0675	-1000000	-1000000	-1000000
1	0	0	0	1	-1	0	0	0.08	-1000000	-1000000	-1000000
0	0	0	1	0	0	1	-1	0.08	-1000000	-1000000	-1000000
0	1	-1	-1	0	0	0	0	0.0029	-1000000	-1000000	-1000000
-1	0	0	0	-1	1	0	0	0.00002875	-1000000	-1000000	-1000000
0	0	0	-1	0	0	-1	1	0.00002875	-1000000	-1000000	-1000000
1	-1	0	0	0	111	0	0	20	200	-1000000	-1000000
-1	1	0	0	0	0	0	111	0.69	200	-1000000	-1000000
215	215	0	0	0	215	0	215	-1000000	-1000000	17	5
-900000	-900000	-900000	-900000	-900000	-900000	-900000	-900000	-1000000	-1000000	-1000000	-1000000
-100	-100	-100	-100	-100	-100	-100	-100	-1000000	-1000000	-1000000	-1000000
20	20	0	10	1207768	0	603884	0	-1000000	-1000000	-1000000	-1000000
100	100	100	100	100	100	100	100	-1000000	-1000000	-1000000	-1000000
20	20	0	10	1207768	0	603884	0	-1000000	-1000000	-1000000	-1000000
-1000000	-1000000	-1000000	-1000000	-1000000	-1000000	-1000000	-1000000	-1000000	-1000000	-1000000	-1000000
Rab5a	Rab5i	Rab5i(c)	bind site	GEF(c)	Rab5aGEF	GAP(c)	GAP(m)			↳raggio cellula	↳raggio endosoma

Figure 4.1. Example of the table read by the algorithm. In this example, we consider [Rab5: GDP] = [Rab5: GTP] = 16.96nM, [$b.s.$] = 8.48nM, [GEF] = 100nM, [GAP] = 50nM, [GDI] = 300nM (the GDI concentration is contained in the parametrization of the first rate, as explained in Section 5.1). The other parameters are easily obtained by consulting Table 4.1.

The first row of the table is filled with 0/1 values, which indicate whether a reactant is global (0) or local (1). Global reactants are dispersed in the cytosol, while local ones are confined to the membrane. According to this, the populations of reactants are (light blue at the bottom of the table) are to be intended as populations per site in the local case, and as global populations in the global case. Observe that they are expressed as the number of proteins (per site, in the local case), so one needs to multiply them by the volume and by Avogadro's number to obtain their equivalent expression as concentrations.

On the right-hand side of the table we have the reaction rates and the Michaelis-Menten constants (for the catalytic reactions). Indeed, all catalytic reactions in our model are assumed to follow Michaelis-Menten kinetics. The gold row between the "change" matrix and the reactants' populations contains the diffusion coefficients of each reactant. Note that we define the diffusion coefficient only for the local reactants, since we assume infinite diffusivity in the cytosol, as we did in the analytical model. In this way the enzymes dispersed in the cytosol are available everywhere at any time for the recruitment to the membrane. This assumption is reasonable because we know that the membrane environment is more viscous than the cytosol, allowing the attachment of the reactants and their subsequent diffusion over it.

The “change” matrix contains all the changes in populations that occur when a reaction is performed. The number ’111’ is associated to the enzyme in the catalytic reactions and represents a sort of flag for the algorithm, in order to identify the enzyme which mediates that catalysis.

Finally, two yellow cells at the bottom right, represent the radii of the cell and of the endosome, which are used to compute respectively the system’s volume and the membrane surface area, respectively. In fact, we want to replicate a realistic situation of an endosome (in particular, an early endosome in the case of Rab5) surrounded by the cellular environment (cytosol). The endosome membrane is modeled as a honeycomb lattice of 10,242 sites (hexagonal, apart from 12 pentagonal defects), and, as mentioned, the number of molecules of any given type in each site is represented as an integer variable.

4.1 Dimensional analysis and rate rescaling in discrete space

We report here all implicit units of measurement associated with the input parameters:

Parameter	Unit of measure
r_{cell}	μm
r_{endo}	μm
k_{diss}	s^{-1}
k_{ass}	$s^{-1}\text{nM}^{-1}$
k_{cat}	s^{-1}
K_M	nM
D	$10^{-6}\,\frac{\mu m^2}{s}$

Table 4.1. Units of measure of the parameters used by our algorithm.

Note that diffusion coefficients must be provided with an implicit 10^{-6} scaling factor due to internal computational requirements.

As we anticipated earlier, the algorithm uses the radius to compute the cytosolic volume and the membrane surface area of the endosome. In both cases, we assume a spherical shape, which leads to:

$$V = \frac{4}{3}\pi(r_{\text{cell}}^3 - r_{\text{endo}}^3) \quad [\mu m^3]$$

$$S = 4\pi r_{\text{endo}}^2 \quad [\mu m^2]$$

where the volume represents the region occupied by the cytosol. Hence, we have to subtract the endosome volume from the total cell volume and consider the volume of a spherical shell. Notice that typically $V_{\text{endo}} \ll V_{\text{cell}}$, so that one could also consider just the cell volume. We use the following notation to represent the nondimensional quantities handled by the algorithm: $V = v \cdot \mu m^3$; $S = a \cdot \mu m^2$.

Since we have to deal with a lattice, we must adapt the rates to the discrete case because they are defined for continuous space. The following rescalings are adopted in the algorithm:

```
for( i=1 ; i <= nR ; i++ )
   {
   switch( r[i][0] ) {
      case DESORPTION:      //Kdiss=[1/s]
         break;
      case ADSORPTION:      //Kass=[1/(s*nM)]
         kappa[i]/=(.1*volume*na);
         break;
      case CATALYSIS:      //Kcat=[1/s]
         Km[i]*=(.1*na*volume/(double)N_SITES);      //Km=[nM]
         break;
      case DIFFUSION:      //D=[um^2/s]
         kappa[i]/=(sqrt(3)*surface/nx);
         break;
      case MUTATION:
         break;
      default:
         printf("Error␣%d␣%d\n",r[i][0],i);
         break;
   }
   }
```

where i runs over all reactions in our model. As we can see, some rates do not need to be rescaled, such as the dissociation and catalytic rates. In fact, they do not depend directly on the concentrations of the reactants. This means that the probability of reaction is independent from the fact that we are in a discrete or continuous case, because we don't have to consider an interaction probability.

Instead, every rate that depends on reactant concentration needs to be rescaled in order to adapt to the discrete case, in which we want to deal with concentrations expressed as molecules per site. Let us consider the kinetics of a generic association reaction in which a cytosolic molecule B associates with a membrane-bound molecule A, i.e. $A^{(m)} + B^{(c)} \rightarrow AB^{(m)}$:

$$\begin{aligned}
\frac{d}{dt}[AB] &= k_{\text{ass}}[A][B] \\
\frac{d}{dt}[AB]\cdot S &= k_{\text{ass}}[A][B]\cdot S \\
\frac{d}{dt}n_{AB} &= k_{\text{ass}}\, n_A[B]\cdot\frac{V}{V} \\
\frac{d}{dt}n_{AB} &= \frac{k_{\text{ass}}}{V}\, n_A\, n_B^{\text{cyto}}
\end{aligned}$$

where $[AB] = \frac{n_{AB}}{S}$; $[A] = \frac{n_A}{S}$ since they are surface concentrations, $[B] = \frac{n_B^{\text{cyto}}}{V}$ since it is a volume concentration. We see that, to handle concentrations expressed as number of molecules, we must divide the rate by the volume. Furthermore, we still have to rescale k_{ass} because it is expressed in $s^{-1}\text{nM}^{-1}$ but we require a rate in $s^{-1}\text{molecules}^{-1}$. Let us call the (non-dimensional) input rate as $k_{\text{ass}}^{\text{phys}}$ and introduce the (non-dimensional) rate that will be used by our algorithm, $k_{\text{ass}}^{\text{model}}$. We introduce these nondimensional rates in order to explicit the physical dimensions, we have the following relation:

$$\begin{aligned}
\left(\frac{k_{\text{ass}}^{\text{model}}}{V}\quad\frac{1}{s\cdot\text{molecules}}\right) &= \left(\frac{k_{\text{ass}}^{\text{phys}}}{v\cdot\mu m^3}\quad\frac{1}{s\cdot\text{nM}}\right) \\
\left(\frac{k_{\text{ass}}^{\text{model}}}{V}\quad\frac{1}{s\cdot\text{molecules}}\right) &= \left(\frac{k_{\text{ass}}^{\text{phys}}}{v\cdot 10^{-15}L}\quad\frac{1}{s\cdot 10^{-9}\frac{\text{mol}}{L}\cdot\text{na}\cdot 10^{23}\,\frac{\text{molecules}}{\text{mol}}}\right) \\
\left(\frac{k_{\text{ass}}^{\text{model}}}{V}\quad\frac{1}{s\cdot\text{molecules}}\right) &= \left(\frac{k_{\text{ass}}^{\text{phys}}}{v\cdot\text{na}\cdot 10^{-1}}\quad\frac{1}{s\cdot\text{molecules}}\right)
\end{aligned}$$

where $\text{Na} = 6,022\cdot 10^{23}\,\frac{\text{molecules}}{\text{mol}} = \text{na}\cdot 10^{23}\,\frac{\text{molecules}}{\text{mol}}$. We used the notation introduced before for the volume and considered the equivalence $1\mu m^3 = 10^{-15}L$. This explains the rescaling adopted in the algorithm.

The other parameter that needs a rescaling since it is explicitly a concentration is the Michaelis-Menten constant. As shown in Table 4.1, it is expressed in nM; we have to convert it to molecules per site:

$$1\,\text{nM} = 10^{-9}\frac{\text{mol}}{L} = 10^{-9}\frac{\text{mol}}{10^{15}\mu m^3} = 10^{-9}\frac{\text{na}\cdot 10^{23}\,\text{molecules}}{10^{15}\mu m^3} = 10^{-1}\cdot\text{na}\cdot\frac{\text{molecules}}{\mu m^3}$$

where $\text{na} = 6.022$, as we said before. Now we have to multiply by the volume $V = v\cdot\mu m^3$ and divide by the number of sites $n_x = 10.242$ to obtain the equivalent concentration in molecules per site:

$$1\,\text{nM} = 10^{-1}\cdot\text{na}\cdot\frac{\text{molecules}}{\mu m^3} = 10^{-1}\cdot\text{na}\cdot\frac{v}{n_x}\cdot\frac{\text{molecules}}{\text{site}} \tag{4.1}$$

Looking at Table 4.1 we see that this is precisely the factor multiplying the input M.M. constant.

Finally, the last parameter that requires rescaling is the diffusion coefficient, since it originates from a continuous-space formulation. Indeed, in the discrete case membrane-bound molecules are not allowed to go in any direction with any spatial step, they are instead forced to move to an adjacent hexagonal site (we have also 12 pentagonal sites, which can be neglected for the rescaling). Hence, we have to consider an appropriate geometric factor to reflect the continuous diffusion coefficient in my discrete space. As shown in the algorithm above, the rescaling is:

$$D^{\text{model}} = \frac{D^{\text{phys}}}{\sqrt{3} \cdot \frac{a}{n_x}} \tag{4.2}$$

where $a = \frac{S}{\mu m^2}$ is the nondimensional surface and n_x is the number of sites. In the algorithm the diffusion coefficients (one for every local reactant) are stored in the array *kappa* after the list of reaction rates; here, instead, we denote the coefficient with the usual D. The geometric factor is obtained by multiplying the distance between two neighbouring hexagonal sites ($\sqrt{3}$) by the surface area of a single membrane site, $\frac{a}{n_x}$.

These rescaled rates and units form the basis upon which our algorithm operates, ensuring consistency between physical parameters and the discrete computational model.

4.2 Evolution of the system

In this section, we describe how the algorithm handles the temporal evolution of the system. The probability for each reaction to occur in a site x is decomposed in the product of a global part (site independent) and a local part (site dependent). Global rates depend on the total number of available reactants, whereas local rates are functions of their concentrations at each specific site [15]. The global component is used to select, at each step, which reaction will be performed, while the local component is used to select the site x on which it will take place.

The global reaction rates are computed as follows:

```
f[0]=0;       // contains the total frequency of all reactions
for( i=1 ; i <= nR ; i++ ) {
   f[i] = kappa[i]*w[i]*weight(i,0,r,n,Km,global);
   f[0] += f[i];
}
```

where `kappa[i]` is the rate constant associated with reaction i (or the diffusion coefficient, if the reaction is diffusive) and `w[i]` is the sum over all local weights for reaction i. These local weights are computed using the function `weight(i,x,r,n,Km,global)`, which takes into account both local populations and reaction parameters. In this context, the function is evaluated at $x=0$ to represent the global contribution of the reaction, by convention (see Figure 4.2, which shows the implementation of the `weight` function). Then, we define `f[0]` as the sum of the global reaction weights `f[i]`.

These instructions are executed at the beginning of each iteration of the main loop that governs the system's time evolution. This ensures that reaction frequencies are continuously updated to reflect changes in the system state.

The next step is to select which reaction will occur. This is done in the following way:

```
xtmp=f[0]*Xrandom();
i=1;
while( (xtmp -= f[i]) > 0 ) i++;
```

where `Xrandom()` generates a random number between 0 and 1. By multiplying it by the total reaction frequency, we obtain a value `xtmp` between 0 and `f[0]`. The subsequent while loop searches the interval in which `xtmp` "falls". This process effectively selects a reaction i with probability proportional to its global frequency.

Once the reaction is selected, the site x where it will occur is chosen. This is done immediately after the while loop above, through the instruction:

```
x=wpop(i);
```

The function wpop(i) uses a similar method based on cumulative weights to randomly select a site according to the local frequencies of reaction i (see Figure 4.3).

Finally, the reaction or diffusion step is performed by modifying the reactants' populations according to the selected reaction and by updating the weights associated with the reactions to reflect the changes in population (see Appendix A for details).

Time advances as a Poisson process with mean $f[0]^{-1}$, which correponds to the average waiting time between two events.

```
double weight(int i,int x,int **r,int **n, double *Km,int *global) {
   double tmp;
   int itmp;
   switch( r[i][0] ) {
      case DESORPTION:     // (1) → (2) + (3)(cyto)
         if( x == 0 )   return 1.;
         else           return (double)n[x][ r[i][1] ];
         break;
      case ADSORPTION:     // (1)(cyto) + (2) → (3)
         if( x == 0 )   return (double)n[0][ r[i][1] ];
         else           return (double)n[x][ r[i][2] ];
         break;
      case CATALYSIS:      // (1) + (3) → (2) + (3)
         if(global[ itmp=r[i][1] ]==1)
            {
               if( x == 0 )   return 1.;
               else
                       {
                          tmp=n[x][itmp];
                          return n[x][ r[i][3] ]*tmp / (Km[i]+tmp);   //M.M
                       }
            }
         if(global[ itmp=r[i][1] ]==0)
         //else
            {
               tmp = n[0][ itmp ]/(double)N_SITES;
               if( x == 0 )   return tmp;
               else return  n[x][ r[i][3] ] / (Km[i]+tmp);
                 }
         break;
      case DIFFUSION: // (1)(x) → (1)(y)
         if( x == 0 )   return 1.;
         else           return (double)n[x][ r[i][1] ];
         break;
      case MUTATION: // (1) → (2)
         if( x == 0 )   return 1.;
         else           return (double)n[x][ r[i][1] ];
         break;
      default:
         printf("Error␣%d␣%d\n",r[i][0],i);
         break;
   }
  return 0.;
}
```

Figure 4.2. The weight function computes the reaction weights according to the mass-action kinetics.

```
int wpop(int i)
  {
   int x,m;
   double r;
   r=B[i][0]*Xrandom();
   for (x=0,m=1;m<K;m<<=1) if (r<B[i][x+m]) x+=m; else r-=B[i][x+m];
   if (r<W[i][x+K]) x+=K;
   return x;
  }
```

Figure 4.3. The wpop function selects the site where the selected reaction will occur.

4.3 Visualization

At the end of the main computational cycle, the algorithm calls the function `printeps`, which saves the state of the system in a text file (see Appendix A). It generates, at regular intervals of `DELTAT` seconds, a file named `Detailed_State_time.txt`, which contains the number of reactants for each site arranged in a matrix of 10,242 rows and a number of columns equal to the number of reactants; and a file named `Global_State_time.txt`, which contains the global population of each reactant at that time (see Figure 4.4).

Once the simulation is over, the text files are processed by a Python script within a Jupyter Notebook, which populates a sphere with the information contained in `Detailed_State_time.txt` files. To do this, it needs additional text files that specify: the three coordinates (with the origin at the center of the sphere) of each site center; the coordinates of the vertices (two per site); the indices of the neighbors' faces (one per edge, so six per site). After that, the algorithm converts the hexagonal mesh into triangles and maps the local concentrations of a particular reactant onto these triangles, using the function `plot` from the `meshplot` library. This process produces interactive 3D images, allowing the user to rotate the sphere and explore regions that would otherwise be hidden in a static representation (more details about this code are in the Appendix A). These images display the spatial distribution of the chosen parameter on the spherical surface, representing, in this case, the early endosome, over time.

We will use two types of variables to visualize the simulation results. The most natural one is the difference between Rab5:GTP and Rab5:GDP at each site, consistent with the definition of the *order parameter* introduced in the theoretical framework. This quantity captures local asymmetries and, it is, therefore, a common choice to investigate phase separation phenomena. It provides an effective tool for detecting domains enriched in either the active or the inactive form of Rab5.

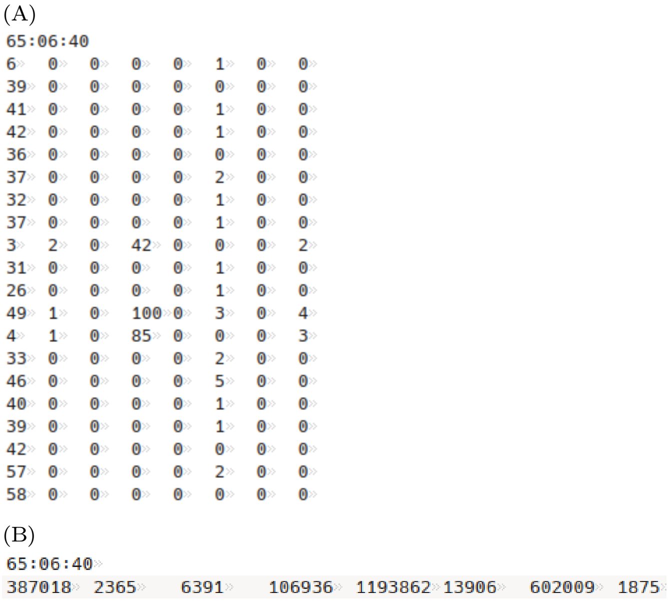
The second option for visualization is to populate the sphere with the total concentration of membrane-bound Rab5, i.e., the sum of Rab5:GTP and Rab5:GDP at each site. This representation provides complementary information because it shows the overall Rab5 recruitment to the membrane, regardless of its activation state. It is particularly relevant for comparing our simulations to experimental observations, such as those reported by *Cezanne et al.*(2020) [6], where they mark with GFP all the Rab5 proteins, both in GTP- or GDP-bound states.

In both cases, the computed concentrations are normalized within each time frame, and a colormap is generated to reflect the distribution at that specific time. In this way, we are sure that the color scale adapts to the range of values in each snapshot, enabling an immediate visual interpretation of local enrichments. The colormap is chosen so that it undergoes a transition from dark blue (low values) to yellow (high values); the intermediate values are green or light blue. Since we independently generate a colorbar for each frame, the interpretation of the spatial patterns must be done in reference to the corresponding colorbar, which encodes the actual concentration visualized at that time point.

These two visualization methods provide complementary information: the first one highlights compositional asymmetries, while the second one shows the global redistribution of Rab5 over the membrane.

Since the simulation outputs consist of text files, we also extract a quantitative indicator to infer whether phase separation has occurred while the simulation is running. This value represents the fraction of the total weight associated with the first twenty spherical harmonics of the spatial distribution of the order parameter. This value reflects well the large-scale spatial organization on the membrane and provides a clear signature of phase-separated domains.

A high value of this percentage signals a stronger spatial organization, consistent with the emergence of patterns, whereas a low value indicates a uniform distribution. Moreover, we use this value to assess whether the system has reached a steady state; indeed, when it oscillates within a narrow range for a sufficiently long period, we can assume that the dynamics have stabilized.



(A)

```
65:06:40
6    0    0    0    0    1    0    0
39   0    0    0    0    0    0    0
41   0    0    0    0    1    0    0
42   0    0    0    0    1    0    0
36   0    0    0    0    0    0    0
37   0    0    0    0    2    0    0
32   0    0    0    0    1    0    0
37   0    0    0    0    1    0    0
3    2    0    42   0    0    0    2
31   0    0    0    0    1    0    0
26   0    0    0    0    1    0    0
49   1    0    100  0    3    0    4
4    1    0    85   0    0    0    3
33   0    0    0    0    2    0    0
46   0    0    0    0    5    0    0
40   0    0    0    0    1    0    0
39   0    0    0    0    1    0    0
42   0    0    0    0    0    0    0
57   0    0    0    0    2    0    0
58   0    0    0    0    0    0    0
```

(B)

```
65:06:40
387018  2365    6391    106936  1193862 13906   602009  1875
```

Figure 4.4. (A) Example of a `Detailed_state` file. We report only the first 20 lines, i.e. 20 lattice cells. The example is taken from the simulation described in Section 6.1, at the final time considered, when equilibrium, characterized by phase separation, is reached. (B) Example of `Global_state` from the same simulation and instant time.

For both files each column identifies a specific reactant, in order, from left to right they are: Rab5:GTP, Rab5:GDP, Rab5:GDP(cyto), binding sites, RR(cyto), Rab5:GTP:RR, GAP(cyto), GAP(m), where RR is, as we already know, the complex GEF/effector of Rab5 protein, i.e., Rabex5:Rabaptin5.

Chapter 5

Parameter estimation

In the previous chapter, we discuss the generic workflow and features of our algorithm. In the following, we become more specific and analyze what the reactions, the reactants and the parameters constituting our system might be. Of course, there is not a unique way to do it. We combine references from the literature, especially the works by *Cezanne et al.*(2020) [6] and *Bezeljak et al.*(2020) [3], with theoretical considerations in order to design a workflow as realistic as possible.

In particular, we rely on the paper by *Cezanne et al.* because of the importance of its studies on how the lipid composition of the EE membrane affects the formation of Rab5 domains (see Sections 2.4 and 5.4) and also because from it we can extract some images of their results using in vitro reconstitution systems. From these, we are able to estimate the interface width of an active Rab5 domain and put some analytical constraints on the value of the diffusion coefficients for membrane-bound species (see Section 5.3).

The other main reference is the work by *Bezeljak et al.*, as mentioned. From it, we take the reactions and the corresponding reaction rates constituting our model, with some additional considerations and assumptions (see Section 5.1).

5.1 Reaction scheme

We have to choose the reactions constituting our Rab5 activation model. A reasonable reaction scheme is the one proposed by *Bezeljak et al.* in [3]: (we add the superscript B to the rate constants for clarity)

$$\begin{aligned}
\text{Rab5: GDP: GDI} &\underset{k_2^B}{\overset{k_1^B}{\rightleftharpoons}} \text{Rab5: GDP} + \text{GDI} \\
\text{Rab5: GDP} + \text{RR} &\overset{k_4^B}{\rightarrow} \text{Rab5: GTP} + \text{RR} \\
\text{Rab5: GTP} + \text{RR} &\underset{k_6^B}{\overset{k_5^B}{\rightleftharpoons}} \text{Rab5: GTP: RR} \\
\text{Rab5: GDP} + \text{Rab5: GTP: RR} &\overset{k_7^B}{\rightarrow} 2\text{Rab5: GTP} + \text{RR} \\
\text{Rab5: GTP} &\overset{k_3^B}{\rightarrow} \text{Rab5: GDP}
\end{aligned}$$

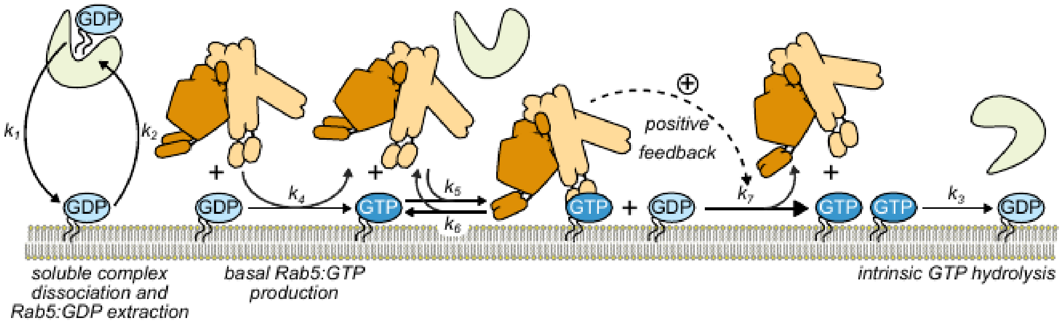


Figure 5.1. Schematic representation of the Rab5 activation model used by *Bezeljak et al.* ([3], Figure 1(G)).

The first step is to adapt this model to our algorithm. Notably, the authors of this work do not consider the spatial component, because their aim is to shed light onto the role of stochasticity in the collective activation of a Rab5 regulatory network. In contrast, our goal is to observe that, beyond their activation on membranes, Rab5 proteins can also self-organize in spatial domains. This means that we need to take into account also the diffusion of the reactants over the membrane.

Another important observation is that some reactions cannot be directly implemented in our algorithm. In fact some of them are actually combinations of more than one elementary reaction. In particular, let us consider the 4th reaction, the one whose aim is to model the positive-feedback loop. We observe that it is the sum of two reactions:

$$\begin{aligned} \text{Rab5: GDP} + \text{Rab5: GTP: RR} &\overset{k_7}{\rightarrow} \text{Rab5: GTP} + \text{Rab5: GTP: RR} \\ &\overset{k_6^B}{\rightarrow} \text{Rab5: GTP} + \text{Rab5: GTP} + \text{RR} \end{aligned} \tag{5.1}$$

The first is a catalytic reaction in which the complex Rab5:GTP:RR promotes the nucleotide exchange that activates a molecule of Rab5, and the second reaction is a dissociation of the complex into an active molecule of Rab5 and RR (Rabex5:Rabaptin5, the Rab5 GEF/effector complex). The dissociation corresponds to another reaction in our scheme, so we assign it the same rate k_6^B. Instead, the first one is an extra reaction whose rate k_7 is to be determined. Obviously it will be related to k_7^B, but we have to pay attention to the fact that k_7^B is expressed in $s^{-1}\text{nM}^{-1}$ (see Figure 5.2). In fact, our algorithm assumes that the catalytic reactions follow Michaelis-Menten kinetics, with rates expressed in s^{-1}. Therefore, to adapt k_7^B to our algorithm, we need to multiply it by a concentration. Assuming Michaelis-Menten kinetics for (5.1):

$$\frac{d}{dt}[\text{Rab5: GTP}] = k_7 \cdot [\text{Rab5: GTP: RR}] \frac{[\text{Rab5: GDP}]}{K_M + [\text{Rab5: GDP}]}$$

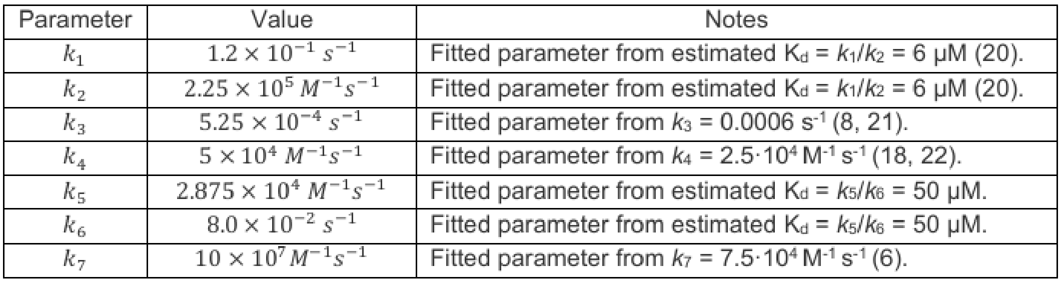
From the physical dimension of k_7^B, we argue that the authors may have included the denominator in the rate, so that we can identify:

$$k_7 = k_7^B \cdot (K_M + [\text{Rab5: GDP}])$$

Furthermore, since this rate remains constant during the simulation according to our algorithm, it is better to consider only constant values in its parametrization. For this reason we identify:

$$k_7 \approx k_7^B \cdot K_M$$

Note that, the occurrence probability for this reaction still depends on [Rab5: GDP] through Michaelis-Menten kinetics. Similarly, since also k_4^B is expressed in $s^{-1}\text{nM}^{-1}$, one can apply the same reasoning to the other catalytic reaction defining a new parameter k_4, suitable with our algorithm, and obtain $k_4 \approx k_4^B \cdot K_M$. However, we will not consider this contribution since from Figure 5.2 we see that $k_7^B \gg k_4^B$ and they are essentially parametrizing the same event, the activation of a Rab5 protein. The only difference is that in one case it is a transitory GEF molecule that mediates the activation, while in the other case it is the membrane-bound complex Rab5:GTP:RR that mediates it (positive-feedback loop). Hence, we expect that the activation contribution comes mainly from the positive-feedback loop.



Parameter	Value	Notes
k_1	$1.2 \times 10^{-1}\ s^{-1}$	Fitted parameter from estimated $K_d = k_1/k_2$ = 6 μM (20).
k_2	$2.25 \times 10^5\ M^{-1}s^{-1}$	Fitted parameter from estimated $K_d = k_1/k_2$ = 6 μM (20).
k_3	$5.25 \times 10^{-4}\ s^{-1}$	Fitted parameter from k_3 = 0.0006 s^{-1} (8, 21).
k_4	$5 \times 10^4\ M^{-1}s^{-1}$	Fitted parameter from $k_4 = 2.5 \cdot 10^4\ M^{-1}\ s^{-1}$ (18, 22).
k_5	$2.875 \times 10^4\ M^{-1}s^{-1}$	Fitted parameter from estimated $K_d = k_5/k_6$ = 50 μM.
k_6	$8.0 \times 10^{-2}\ s^{-1}$	Fitted parameter from estimated $K_d = k_5/k_6$ = 50 μM.
k_7	$10 \times 10^7\ M^{-1}s^{-1}$	Fitted parameter from $k_7 = 7.5 \cdot 10^4\ M^{-1}\ s^{-1}$ (6).

Figure 5.2. Table of the rates used for the simulations in the reference article [3]. The table has been taken from the supplementary material of the paper (Table S1).

The next reactions we have to adapt are the first ones, characterized by rates k_1^B and k_2^B, which describe the dissociation and association of the inactive Rab5 (Rab5:GDP) from and to the membrane. By looking at Figure 5.1 we understand that the authors assume that whenever a Rab5 protein dissociates from GDI, it is immediately recruited to the membrane. However, this assumption oversimplifies the process. In reality, after dissociation from GDI, the Rab5 protein remains in the inactive state in the cytosol and eventually associates with the membrane through its lipid anchor. Moreover, in our algorithm, membrane recruitment is explicitly modeled as the association of some protein (Rab5 in our case) dispersed in the cytosol with specific binding sites. Instead, in the reference article, since the spatial component is not taken into account, there is no distinction between cytosolic reactants and membrane-bound reactants. In our case, however, the spatial component is crucial to observe domain formation, hence we have to adapt these two reactions to our model.

Let us rewrite both equations (direct, inverse) as a sum of two equations in the following way:

$$\begin{aligned} \text{Rab5: GDP: GDI} &\underset{k_a}{\overset{k_d}{\rightleftharpoons}} \text{Rab5: GDP}^{\text{cf}} + \text{GDI} && (5.2)\\ \text{Rab5: GDP}^{\text{cf}} + \text{bind site} &\underset{k_-}{\overset{k_+}{\rightleftharpoons}} \text{Rab5: GDP}^{m} && (5.3)\end{aligned}$$

where we separate Rab5:GDP molecules in membrane-bound (m) and cytosolic-free (cf). Our final goal is to write a double reaction ($\rightleftarrows$) that suits our computational model and that reflects correctly the full process, i.e.:

$$\text{Rab5: GDP}^{c} + \text{bind site} \underset{k_2}{\overset{k_1}{\rightleftharpoons}} \text{Rab5: GDP}^{m} \qquad (5.4)$$

where we define $\text{Rab5: GDP}^{c} = \text{Rab5: GDP}^{\text{cf}} + \text{Rab5: GDP: GDI}$, that is the full concentration of inactive Rab5 dispersed in the cytosol, composed of free Rab5 molecules and Rab5 molecules that form a complex with GDI. In this way we can model the first two equations in a suitable way for our algorithm and we can also take into account the role of GDI in the process. In fact, we still have to determine the new rates k_1 and k_2, but it is quite evident that they will be related to the original rates in a GDI dependent way.

From (5.2) and (5.3) we can derive the following kinetics:

$$\begin{aligned} \frac{d}{dt}\left[\text{Rab5: GDP}^{\text{cf}}\right] &= k_d[\text{Rab5: GDP: GDI}] - k_a\left[\text{Rab5: GDP}^{\text{cf}}\right] - k_+\left[\text{Rab5: GDP}^{\text{cf}}\right][b.s.] && (5.5)\\ \frac{d}{dt}\left[\text{Rab5: GDP}^{m}\right] &= k_+[b.s.]\left[\text{Rab5: GDP}^{\text{cf}}\right] - k_-\left[\text{Rab5: GDP}^{m}\right] && (5.6)\end{aligned}$$

Since from the reference article we understood that they consider an approximately istantaneous recruitment of the cytosolic-free Rab5:GDP to the membrane, we can assume that our second reaction is in equilibrium, so that:

$$\begin{aligned} k_+[b.s.]\left[\text{Rab5: GDP}^{\text{cf}}\right] &= k_-\left[\text{Rab5: GDP}^{m}\right]\\ \left[\text{Rab5: GDP}^{\text{cf}}\right] &= K_-\frac{\left[\text{Rab5: GDP}^{m}\right]}{[b.s.]} && (5.7)\end{aligned}$$

where we define $K_- = \frac{k_-}{k_+}$, the dissociation constant of the second equation. Now we write also the kinetics of the reference reaction:

$$\begin{aligned} \frac{d}{dt}\left[\text{Rab5: GDP}^{m}\right] &= k_1^B[\text{Rab5: GDP: GDI}] - k_2^B[\text{GDI}]\left[\text{Rab5: GDP}^{m}\right]\\ &= k_1^B\Big(\left[\text{Rab5: GDP}^{c}\right] - \left[\text{Rab5: GDP}^{\text{cf}}\right]\Big) - k_2^B[\text{GDI}]\left[\text{Rab5: GDP}^{m}\right]\end{aligned}$$

where we expressed the concentration of the complex as the difference between the full concentration of the cytosolic Rab5:GDP and the concentration of the free Rab5:GDP molecules. In this way we can substitute (5.7) and get:

$$\begin{aligned}\frac{d}{dt}\big[\mathrm{Rab5\colon GDP}^{m}\big] &= k_1^B\Big(\big[\mathrm{Rab5\colon GDP}^{c}\big]-\frac{K_-}{[b.s.]}\big[\mathrm{Rab5\colon GDP}^{m}\big]\Big)-k_2^B[\mathrm{GDI}]\big[\mathrm{Rab5\colon GDP}^{m}\big]\\ &= k_1^B\big[\mathrm{Rab5\colon GDP}^{c}\big]-\Big(k_2^B[\mathrm{GDI}]+\frac{k_1^B K_-}{[b.s.]}\Big)\big[\mathrm{Rab5\colon GDP}^{m}\big] \qquad (5.8)\end{aligned}$$

If we now consider also the kinetics of our “goal” equation (5.4):

$$\frac{d}{dt}\big[\mathrm{Rab5\colon GDP}^{m}\big]=k_1[b.s.]\big[\mathrm{Rab5\colon GDP}^{c}\big]-k_2\big[\mathrm{Rab5\colon GDP}^{m}\big] \qquad (5.9)$$

By comparing (5.8) and (5.9) we find a relation between our rates k_1, k_2 with the reference rates, that are known from Figure 5.2. We have:

$$\begin{aligned}k_1 &= \frac{k_1^B}{[b.s.]}\\ k_2 &= k_2^B[\mathrm{GDI}]+\frac{k_1^B K_-}{[b.s.]}\end{aligned}$$

Observe that we want a large concentration of binding sites w.r.t. the concentration of cytosolic Rab5:GDP, in order not to limit the recruitment of inactive Rab5 to the membrane. Furthermore, since we aim for immediate association with the membrane, we assume $k_+ \gg k_-$ so that we can take the limit $K_- \to 0$. As a result,

$$\begin{aligned}k_1 &= \frac{k_1^B}{[b.s.]}\\ k_2 &= k_2^B[\mathrm{GDI}]\end{aligned}$$

This means that although we don't consider GDI as an explicit reactant, its effect is taken into account through our effective rate k_2. In fact, k_2 is proportional to [GDI] and the fact that it models membrane dissociation further supports our approach.

Finally, the last reaction that we have to discuss is the last one of our scheme, the one with rate k_3^B which inactivates a Rab5 molecule on the membrane. This is a catalytic reaction; the authors of the article do not specify which enzyme mediates it, but we know that it is a GAP protein from our biology knowledge. Observe that we could also treat this equation as a mutation in our algorithm but we prefer to explicitly include the enzyme and treat the reaction as catalytic because we know that cytosolic enzyme diffusion is crucial for the redistribution of Rab proteins on the membrane. In fact it is known that the combination of a local positive-feedback loop with global inhibition supports the formation of spatiotemporal patterns [3].

Hence, we have to add two reactions to our scheme, i.e. association/dissociation of the GAP to and from the membrane, and we have to rewrite the inactivation reaction as a catalytic reaction, expliciting the enzyme involved:

$$\mathrm{GAP}^{m} \underset{k_{\mathrm{ass}}}{\overset{k_{\mathrm{diss}}}{\rightleftharpoons}} \mathrm{GAP}^{c}+b.s. \qquad (5.10)$$

$$\mathrm{Rab5\colon GTP}+\mathrm{GAP}^{m} \overset{k_{\mathrm{cat}}}{\to} \mathrm{Rab5\colon GDP}^{m}+\mathrm{GAP}^{m} \qquad (5.11)$$

where we have introduced new reaction rates that we have to relate to the rates coming from the reference article. Let us start by writing the kinetics of the reference's reaction:

$$\frac{d}{dt}\big[\mathrm{Rab5\colon GDP}^{m}\big]=k_3^B[\mathrm{Rab5\colon GTP}] \qquad (5.12)$$

In our model, however,

$$\frac{d}{dt}\big[\mathrm{Rab5\colon GDP}^{m}\big]=k_{\mathrm{cat}}\big[\mathrm{GAP}^{m}\big]\frac{[\mathrm{Rab5\colon GTP}]}{K_M+[\mathrm{Rab5\colon GTP}]}$$

It is straightforward to identify:

$$k_3^B=k_{\mathrm{cat}}\frac{\big[\mathrm{GAP}^{m}\big]}{K_M+[\mathrm{Rab5\colon GTP}]}$$

Moreover, since they write the equation as linear in the substrate concentration and since we model the catalytic reactions through Michaelis-Menten, we assume that we are in the linear regime, i.e. we assume that $[\text{Rab5:GTP}] \ll K_M$. Then,

$$k_3^B \approx k_{\text{cat}} \frac{\left[\text{GAP}^m\right]}{K_M} \tag{5.13}$$

If we now impose the equilibrium of (5.10):

$$k_{\text{diss}}\left[\text{GAP}^m\right] = k_{\text{ass}}\left[\text{GAP}^c\right][b.s.]$$

from which

$$\left[\text{GAP}^m\right] = \frac{\left[\text{GAP}^c\right][b.s.]}{K_d^{\text{GAP}}} \tag{5.14}$$

where we define the dissociation constant $K_d^{\text{GAP}} = \frac{k_{\text{diss}}}{k_{\text{ass}}}$.

Let us introduce $\left[\text{GAP}^{\text{tot}}\right] = \left[\text{GAP}^m\right] + \left[\text{GAP}^c\right]$ and substitute it in (5.14):

$$\begin{aligned}
\left[\text{GAP}^m\right] &= \frac{\left(\left[\text{GAP}^{\text{tot}}\right] - \left[\text{GAP}^m\right]\right)[b.s.]}{K_d^{\text{GAP}}} \\
\left[\text{GAP}^m\right]\left(1 + \frac{[b.s.]}{K_d^{\text{GAP}}}\right) &= \frac{\left[\text{GAP}^{\text{tot}}\right][b.s.]}{K_d^{\text{GAP}}} \\
\left[\text{GAP}^m\right] &= \frac{\left[\text{GAP}^{\text{tot}}\right][b.s.]}{K_d^{\text{GAP}} + [b.s.]} \qquad (5.15)
\end{aligned}$$

If we now substitute (5.15) into (5.13):

$$k_3^B = \frac{k_{\text{cat}}}{K_M} \frac{\left[\text{GAP}^{\text{tot}}\right][b.s.]}{K_d^{\text{GAP}} + [b.s.]}$$

or, by isolating k_{cat}:

$$k_{\text{cat}} = k_3^B \frac{(K_d^{\text{GAP}} + [b.s.])K_M}{\left[\text{GAP}^{\text{tot}}\right][b.s.]} \tag{5.16}$$

In this way we have transformed the reaction in order to be suitable for our algorithm without losing the link with *Bezeljak*'s reaction rates. Observe that we have introduced two rates k_{diss}, k_{ass} that are completely arbitrary. Since we have no particular reason to assign specific values to them, we decide to stay close to their counterparts, i.e. the association/dissociation rates of the GEF/effector complex of Rab5 (Rabex5:Rabaptin5, RR in our notation). Therefore, we identify:

$$\begin{aligned}
k_{\text{diss}} &= k_6^B \\
k_{\text{ass}} &= k_5^B
\end{aligned}$$

In conclusion, the reaction scheme that we will consider in our computational model is: (note that we have changed some numbers identifying the rates w.r.t. discussion above, from now on we will consider the following ones)

$$\begin{aligned}
\text{Rab5:GDP}^m &\underset{k_4}{\overset{k_1}{\rightleftharpoons}} \text{Rab5:GDP}^c + \text{bind site} \\
\text{Rab5:GTP:RR} &\underset{k_5}{\overset{k_2}{\rightleftharpoons}} \text{Rab5:GTP} + \text{RR} \\
\text{GAP}^m &\underset{k_6}{\overset{k_3}{\rightleftharpoons}} \text{GAP}^c + \text{bind site} \qquad (5.17) \\
\text{Rab5:GDP} + \text{Rab5:GTP:RR} &\overset{k_7}{\rightarrow} \text{Rab5:GTP} + \text{Rab5:GTP:RR} \\
\text{Rab5:GTP} + \text{GAP}^m &\overset{k_8}{\rightarrow} \text{Rab5:GDP} + \text{GAP}^m
\end{aligned}$$

with rates

$$\begin{aligned}
k_1 &= k_2^B[\mathrm{GDI}] \\
k_4 &= \frac{k_1^B}{[b.s.]} \\
k_2 &= k_6^B \\
k_5 &= k_5^B \\
k_3 &= k_6^B \\
k_6 &= k_5^B \\
k_7 &= k_7^B \cdot K_M \\
k_8 &= k_3^B \frac{(K_d^{\mathrm{GAP}} + [b.s.])K_M}{\left[\mathrm{GAP}^{\mathrm{tot}}\right][b.s.]}
\end{aligned} \tag{5.18}$$

where $K_d^{\mathrm{GAP}} = \frac{k_3}{k_6}$.

5.2 Estimation of the effective cytosolic volume

For our simulations, we use $r_{\mathrm{endo}} = 5\,\mu m$, since we want to replicate the set-up used by *Cezanne et al.*, in which they employ silica beads of $10\,\mu m$ in diameter (see [6], "Results" section). To determine the value of r_{cell} we have to estimate the effective distance at which soluble species in the cytosol contribute to the membrane-bound process. Beyond this distance molecules are unlikely to participate because of diffusion limitations. In our model, we assume infinite diffusivity for cytosolic molecules, in the sense that it is much larger compared to diffusivity on the membrane. However, this assumption makes sense as long as we consider a relatively small cytosolic volume. Therefore, we introduce an effective distance l, which reflects how far a molecule can diffuse before being recruited by the membrane. To estimate l, we use the standard random walk relation for the characteristic displacement of a diffusing particle over a time τ:

$$l \sim \sqrt{D \cdot \tau} \tag{5.19}$$

where D is the diffusion coefficient of the species in the cytosol and τ is the characteristic time for the adsorption process at the membrane. The diffusion coefficient for various species can be inferred from the literature. We use the results obtained by *Arrio-Dupont et al.* in their work [2], where they detected a strong size dependence of the diffusion coefficients of globular proteins in the cytoplasm of cultured muscle cells. We report here (Table 5.2, below) a sketch of their results. Since we know, by looking at Table 5.1, that the molecular weights of our cytosolic species (Rab:GDP/GDI, GEF, GAP) vary from 75 kDa to 165 kDa, we argue that a reasonable cytosolic diffusion coefficient would be approximately $D = 10 \mu m^2/s$.

Moreover, in the final Discussion, they also pointed out that the diffusivity of proteins in native muscle fibers could be reasonably smaller than in cultured muscle cells. Of course, the exact value of D can vary depending on the molecular size, morphology and local environment. For example, *Jimenez et al.* found that the diffusion coefficient of proteins in human fibroblasts exhibits almost no dependence on the molecular weight and they made an estimation of $D \simeq 1\text{-}2\,\mu m^2/s$ [20]. Since it is hard to find an exact value we prefer to overestimate a bit the effective region that contributes to our system and adopt $D = 10\,\mu m^2/s$.

The characteristic time τ is obtained by exploiting the fact that at equilibrium we expect that the rates of association and dissociation are approximately equal. Thus, the corresponding timescales can also be assumed to be comparable:

$$\tau \sim \frac{1}{k_{\mathrm{ass}}[X^{\mathrm{cyto}}]} = \frac{1}{k_{\mathrm{diss}}} \tag{5.20}$$

This provides a direct way to estimate the time scale for interactions, avoiding the need to estimate the concentration of molecules in the cytosol, which could introduce additional uncertainties. To ensure that we capture all relevant interactions, we select the smallest dissociation rate k_{diss}, which results in the largest τ, and therefore the largest effective distance. Using equations 5.19 and 5.20, we compute the effective distance l. Hence, the radius of the cell is:

$$r_{\text{cell}} = r_{\text{endo}} + l \tag{5.21}$$

Finally, the volume that we want our algorithm to use is the volume of the spherical shell. This means that it will be computed following:

$$V = \frac{4}{3}\pi(r_{\text{cell}}^3 - r_{\text{endo}}^3) \tag{5.22}$$

This construction is consistent with the assumptions of fast cytosolic diffusion and allows us to define a finite, physically motivated effective volume.

Protein	Id	Lenght (AA)	Mass (kDa)
Rab5	RAB5A_HUMAN	215	23,7
GDI	GDIA_HUMAN	447	50,6
Rabex5	RABX5_HUMAN	491	56,9
Rabaptin5	RABE1_HUMAN	862	99,3
GAP	GAPD1_HUMAN	1478	165

Table 5.1. Molecular weights of the proteins of the system. The data are taken from the online database UniProt [29]. The 'Id' column contains the entry identifiers of the corresponding proteins. The length of each protein (in number of amino acids (AA)) and the relative molecular weights (in kDa) are provided in the last two columns

Protein	M_r(kDa)	$D_{\text{cyto}}(\mu m^2/s)$
Myokinase	21	$46 < D_{\text{cyto}} < 93$
Phosphoglucomutase	60	16.5 ± 3
β-Enolase	90	10.8 ± 2
IgG	160	5.5 ± 1
β-Galactosidase	540	0.004 ± 0.0007
EGFP	27	15.8 ± 3

Table 5.2. Sketch of "Table 1: *Mobility of proteins in the cytoplasm of cultured muscle cells*, Results section" from the paper by *Arrio-Dupont et al.* [2]. The measurement details are described in the "Materials and methods" section, "Diffusion measurements"; the values of the diffusion coefficients are determined by the Padé-Laplace formalism.

5.3 Measurements of interface width set constraints on the diffusion coefficient

According to the model of cell polarity, the interface width is related to diffusivity through the equation (3.21). We use this fact to set some constraints on the values of the diffusion coefficient. We estimate the interface width of a Rab5 domain based on the experimental results reported by *Cezanne et al.* (2020) [6] and take this as a reference (Figure 5.3).

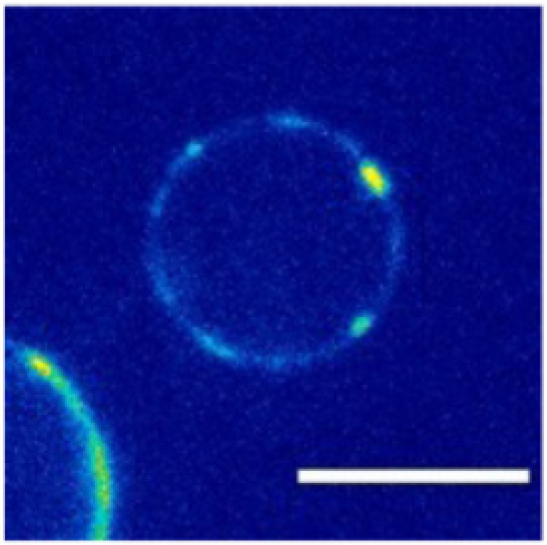


Figure 5.3. Figure 6(B) of the reference article [6], Rab5 domain formation in vitro. In this case, MCBs (membrane-coated beams) have EE lipid composition containing 1 mol% PI(3)P. Beads are presented as equatorial slices in GFP channel. The scale bar (white, bottom right) is $10\,\mu m$.

Using ImageJ, an application for the analysis of scientific images, we measure that the scale bar is ≈ 115 pixels. We know from the article that the scale bar length is $10\,\mu m$, then $1\,\text{pixel} \approx \frac{10\,\mu m}{115} \approx 0.0877\,\mu m$. The next step is to estimate the interface width in pixels and then convert the value in μm. We draw a line along the Rab5 domain and extract the corresponding intensity values through a tool of ImageJ ("Plot Profile") which generates an intensity profile with the spatial component in pixels as horizontal axis. We repeat the process 19 times in order to obtain an average estimation and minimizing the error introduced by our measurement. We also generate a mean profile of our measurements with the relative error (Figure 5.4).

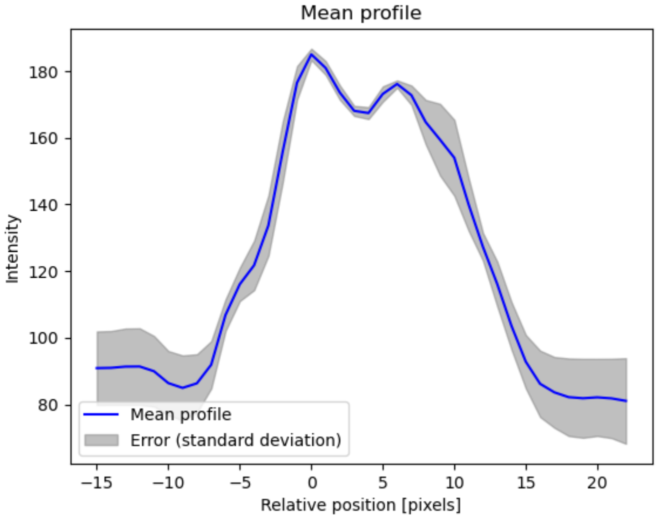


Figure 5.4. Mean profile generated in Python. The profiles have been aligned w.r.t. their maximum.

For the estimation of the interface width we proceed in the following way: since we have two interfaces per profile we separate the values of the left interface from the ones of the right interface; then, for each profile we estimate the left and the right interface by a fitting procedure with a hyperbolic tangent and, finally, we compute the average and the standard deviation of the 19 estimations for both the interfaces.

It is reasonable to adopt an hyperbolic tangent as function for the fitting process as we discussed in Section 3.4. We fit the profile through the function `curve_fit` of the library `scipy.optimize` in Python. The results of our estimations are the following:

$$w_{\text{left}} \simeq (\texttt{3.46} \pm \texttt{0.69})\,\text{pixels}$$

$$w_{\text{right}} \simeq (\texttt{3.83} \pm \texttt{0.53})\,\text{pixels}$$

From these results, we derive a range of reasonable values for the interface width of a Rab5 domain by considering the smaller and the higher values corresponding to our estimations, that is:

$$w_{\text{min}} \simeq (\texttt{3.46} - \texttt{0.69})\,\text{pixels}$$

$$w_{\text{max}} \simeq (\texttt{3.83 + 0.53})\,\text{pixels}$$

which corresponds to the following range of values for w:

$$w \in (\texttt{2.77}, \texttt{4.36})\,\text{pixels}$$

Finally, given our estimation of a pixel in μm:

$$w \in (0.24, 0.38)\mu m \tag{5.23}$$

Now, we have to compute $\alpha^{\pm}$ coefficients. As we know, they depend on several quantities (see eq. (3.22)). We take as reference the work from *Bezeljak et al.* [3]. The authors of this article combine experiments with computational modeling to shed light on the stochastic nature of the collective activation of Rab5 on membranes of early endosomes (EEs). In particular, as physicists, we will focus on the computational aspects of this paper. They consider a model of Rab5 activation (Figure 5.1) and solve it using the Gillespie algorithm to account for biochemical noise (stochasticity) in the reactions. Notice that they consider a reinforcing feedback loop only for the E^+ enzyme; they also take into account the inactivation of Rab5 molecules but without specifying the enzyme which mediates this conversion and, most importantly, without considering a reinforcing feedback loop.

However, as we argue in Section 5.1 we also introduce the negative feedback loop in our model because it is necessary for domain formation. Hence, in order to estimate the $\alpha^{\pm}$ coefficients, we use the rate values of our reference paper (see Figure 5.2) through the parametrization discussed in Section 5.1. We do the following identifications:

- $k_c^+ = k_7 \approx k_7^B \cdot K_M \quad \rightarrow$ rate of the catalytic activity of E^+ on membrane
- $k_c^- = k_3^B \frac{(K_d^{\text{GAP}} + [b.s.])K_M}{\left[\text{GAP}^{\text{tot}}\right][b.s.]} \quad \rightarrow$ rate of the catalytic activity of E^- on membrane
- $E^+_{\text{cyto}} = [\text{GEF}] \quad \rightarrow$ volume concentration of enzyme E^+_{cyto} in cytosol
- $E^-_{\text{cyto}} = [\text{GAP}] \quad \rightarrow$ volume concentration of enzyme E^-_{cyto} in cytosol
- $K_d^+ = k_d^+ / k_a^+ = k_2 / k_5 \quad \rightarrow$ dissociation constant of enzyme E^+
- $K_d^- = k_d^- / k_a^- = k_3 / k_6 \quad \rightarrow$ dissociation constant of enzyme E^-

Keeping in mind these identifications, the $\alpha^{\pm}$ coefficients are easily obtained. Of course, they will depend on the particular parameter set employed for the simulation. Since we perform several simulations for different parameter sets, we report a plot of the diffusion coefficient as a function of the factor $\frac{\alpha^+ + \alpha^-}{2}$, i.e. the sum of the catalytic ratios (see Figure 5.5). In fact, if we invert the relation (3.21), we obtain:

$$D = \frac{w^2}{(2\kappa + 1)^2} \cdot \frac{\alpha^+ + \alpha^-}{2} \tag{5.24}$$

where w is fixed to a value within the range (5.23). In particular, we consider the central value $w = 0.31\,\mu m$ and the two extrema of the range (represented by the three straight lines of Figure 5.5). The uncertainty related to the estimation of w results in the uncertainty of the diffusion coefficient of the membrane-bound species. Hence, for each parameter set considered in our simulations we set a value for the diffusivity D according to this relation in order to be as close as possible to the experimental set-up of *Cezanne et al.* .

Finally, it is important to note that this discussion and its implications rely heavily on our theoretical understanding of the phenomenon and on the assumptions leading to our reference equation (3.21) (see Section 3.4).

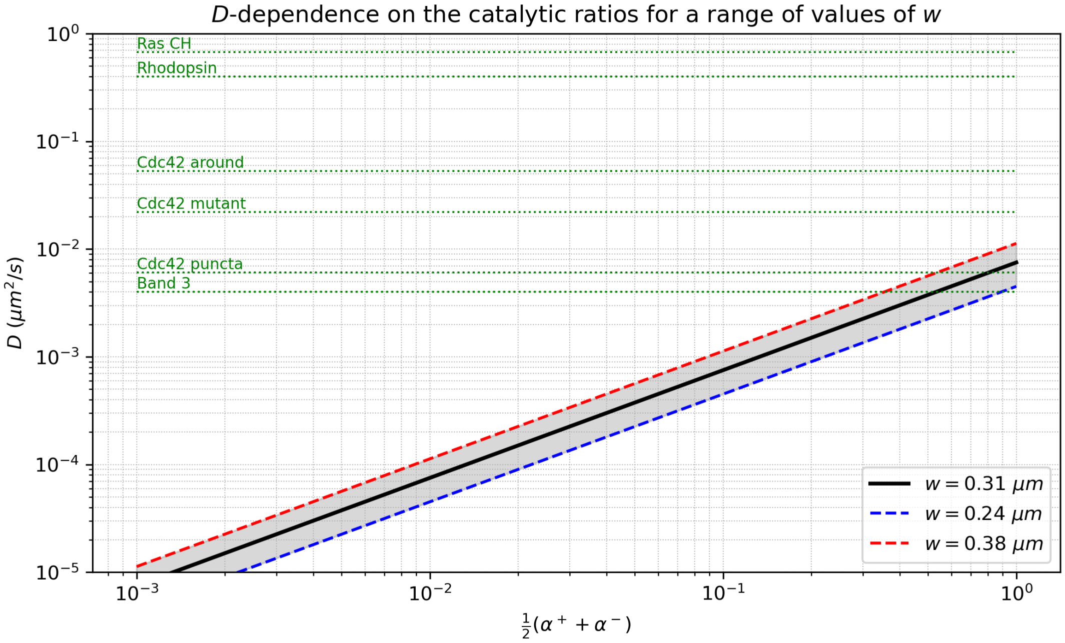


Figure 5.5. Plot of the Diffusion coefficient D as a function of the sum of the catalytic ratios $\alpha^{\pm}$ for the estimated range of values of the interface width w (with $K_M = 200\text{nM}$ and $c = [\text{Rab5: GDP}] + [\text{Rab5: GTP}] = 33.92\text{nM}$ for $\kappa = \frac{K_M}{c}$). We draw a line for the central value $w = 0.31\,\mu m$ (in **black**) and two lines for the extreme values of the range, i.e. for $w = w_{\min} = 0.24\,\mu m$ (in **blue**) and for $w = w_{\max} = 0.38\,\mu m$ (in **red**). We consider a logarithmic scale for both axis since we consider a range of 3 orders of magnitude for the values assumed by the catalytic ratios (as we do in the Results' Chapter 6). The dotted lines in **green** represent some of the experimental measurements depicted in Table 5.3.

5.4 Estimation of the lateral diffusion coefficient for membrane-bound proteins

In Figure 5.5 we pointed out several experimental measurements of the diffusion coefficient for a wide range of proteins and membrane environments. The aim of this section is to assign biologically meaningful values to the diffusion coefficient D for the membrane-bound proteins involved in our model: Rab5 (both active and inactive states), the Rab5:RR complex and the GAP. However, direct measurements of D for these specific proteins in early endosomal (EE) membranes are currently unavailable. To address this, we have to use results from the literature for comparable proteins and membrane environments, combined with our biological and physical knowledge.

The selected experimental measurements are also reported in Table 5.3. These include small GTPases (Rab5, Ras, Cdc42), transmembrane receptors, and lectin- or antigen-binding proteins, measured through the widely used technique called FRAP (Fluorescence Recovery After Photobleaching).

The values span a broad range from approximately $0.003\,\mu m^2/s$ to over $9\,\mu m^2/s$, reflecting the combined influence of:

- membrane viscosity;
- protein crowding;
- interactions with the membrane;
- local lipid environment (e.g., presence of sphingomyelin and cholesterol).

The only measurements regarding directly Rab5 comes from the paper of *Münzberg et al.* (2019) ([24], *Table* 2) and they varies from 4.3 to 9.3 $\mu m^2/s$, which are significantly higher compared to our expectations in such a viscous environment like the EE's membrane. This is likely due to the extremely fluid composition of the artifical lipid bilayer (POPC membrane), which lacks lipid rafts (microdomains) and membrane-associated proteins.

An important reference in this context is provided by the Ras family of small GTPases (HRas, NRas, KRas), which are localized predominantly to the cytosolic face of the plasma membrane, to which they are anchored via lipid modifications. Ras proteins are known to diffuse extremely rapidly compared to other cell surface proteins; their lateral diffusion coefficients range from 0.65 to 1.25 $\mu m^2/s$ under control conditions in COS-7 cells, and from 0.35 to 1 $\mu m^2/s$ when the plasma membrane is enriched with cholesterol ([16], estimated from *Figure* 4(A)). These values exhibit a reduction of $\sim 30 \div 50\%$ in D upon cholesterol loading, which is particularly relevant for us since the EE membrane is also rich in cholesterol. Moreover, they are considerably lower than those measured for Rab5 in artificial POPC bilayers, even though we expect the Ras proteins to have a larger mobility. This suggests us that measurement of Rab5 in more physiological membranes would likely yield significantly slower values.

Another relevant comparison is Cdc42, a small GTPase that shares structural and membrane-anchoring features with Rab5. Measurements in *Saccharomyces cerevisiae* plasma membranes show a slow diffusion, particularly in regions enriched of PS (phospholipid phosphatidylserin) where D drops to 0.0061 $\mu m^2/s$. By contrast, the decay profile outside the puncta (the name assigned to the PS-enriched membrane domain) is consistent with a diffusion coefficient of 0.053 $\mu m^2/s$. The reduction of D is attributed to local lipid order (PS microdomains) and protein clustering; conditions that may be partially replicated in early endosomes. To further investigate the role of PS they repeat the experiment with the Cdc42^{S185D} mutation, which exhibits a minor affinity for PS, and they measure a D at the puncta of 0.022 $\mu m^2/s$, more than three times faster than for wild-type Cdc42 ([27], *Results - Non-uniform membrane diffusion of Cdc42*) . These results support the idea that lipid specificity and microdomain formation can strongly regulate diffusion.

Further insight comes from the classic review by *Richard J., Cherry*(1979), which reports and collects several FRAP measurements of the lateral diffusion coefficient for membrane proteins, in both cellular and artificial systems ([9], *Table 1*, section *IVB(3)*). Here, we report some of his considerations. The diffusion of concanavalin A receptors in different cells is in the range of $10^{-2} \div 10^{-3}\,\mu m^2/s$, with evidences of a decreasing value with increasing Concanavalin A doses, probably due, partially, to aggregation phenomena. Measurements of the diffusion of surface antigens for mast cells and fibroblasts are of the order of $10^{-2}\,\mu m^2/s$; higher, compared to lectin receptors, supporting the idea that mobility is slowed down by aggregation.

There is also a measurement of D for Band 3 in erythrocyte ghosts of 0.004 $\mu m^2/s$ (section *IID(3)*, [9]). Band 3 is an integral protein with molecular weight of about 90 kDa, responsible for anion transport; such a low value is attributed to anchoring to the spectrin cytoskeleton. Furthermore, a value of 0.005 $\mu m^2/s$ is reported for acetylcholine receptors ($\sim 50 \div 60$ kDa [29]) in myotubes; and a value of $0.2 \div 0.6$ $\mu m^2/s$ for Rhodopsin (~ 40 kDa [29]) in photoreceptor membranes of rod outer segments, which is relatively high, but still slower than lipid-anchored proteins in artificial bilayers.

These examples illustrate a clear dependence of the protein lateral mobility on the membrane structure, lipid composition and anchoring mechanism; and they offer experimentally validated lower and upper bounds for the diffusion coefficient of the membrane-bound proteins. For example, the measurements for Rab5 obtained in artificial membranes represent an upper limit of mobility, achievable only in highly fluid, homogeneous bilayers.

As discussed in Section 2.4, the early endosomal membrane is enriched in sphingomyelin and cholesterol, both known to increase membrane order and reduce protein lateral mobility [6],[12],[13]. Moreover, the Rab5 domain formation induced by positive-feedback probably contributes to slow down the protein mobility on the membrane, as also suggested by *Cherry* in his review.

In conclusion, taking into account all the experimental evidence discussed here and the lipid contributions to the membrane's mobility outlined in Section 2.4 strongly support the use of low diffusion coefficients D ($10^{-2} \div 10^{-4}\,\mu m^2/s$) for membrane-bound proteins on the early endosomal surface. These values reflect the expected reduction in lateral mobility in particularly viscous environments such as EE membranes, enriched in sphingomyelin and cholesterol. Furthermore, the two enzymes involved in the activation/inactivation processes are significantly heavier than the proteins depicted in Table 5.3 (see Table 5.1, Section 5.2). This is another reason that leads us to consider an even smaller lower limit for D w.r.t. the reported measurements.

Beyond biological accuracy, however, the values of D used in our simulations must also be consistent, at least approximately, with our theoretical predictions. In particular, they have to respect the expected relation between the interface width w and the catalytic ratios $\alpha^{\pm}$ (see Figure 5.5). Our choice of D therefore balances both empirical plausibility and analytical constraints discussed in Section 3.4.

Protein	**Environment/Membrane type**	$D\,(\mu m^2/s)$	**Reference**
Rab5 (di-GG)	POPC membrane (model lipid bilayer)	$4.3 \div 9.3$	[24]
Ras (HRas, Nras, KRas)	COS-7 plasma membrane, CH-loaded	0.35÷1.00	[16]
Ras (HRas, Nras, KRas)	COS-7 plasma membrane, control	0.65÷1.25	[16]
Cdc42	Yeast plasma membrane, PS-enriched puncta	0.0061	[27]
Cdc42	Yeast plasma membrane, surrounding region	0.053	[27]
Cdc42 S185D	Mutant in yeast membrane	0.022	[27]
Rhodopsin	Disc membrane of rod outer segments	0.2÷0.6	[9]
Band 3 (integral protein)	Erythrocyte ghosts (red blood cell membranes)	0.004	[9]
Acetylcholine receptors	Myotubes	0.005	[9]
Concanavalin A receptors	Fibroblasts (3T3)	0.0039	[9]

Table 5.3. Diffusion coefficient measurements of various different proteins in different membrane systems.

5.5 Reference values for the Michaelis-Menten constant

In this section, we report several experimental measurements of the Michaelis-Menten constant. These are often performed for the ratio k_{cat}/K_M, which represents the catalytic efficiency. This is done especially with low substrate concentrations, when it is the most appropriate and direct parameter to describe the global efficiency of the enzyme. From Table 5.4 we can see that the activity of the Rab5 GEF, Rabex5, in its wild-type form is approximately $10^4 \div 10^5\,M^{-1}s^{-1}$ when the substrate is Rab5, or Rab5A [4],[22]. In the same Table there are also some separate measurements of k_{cat} and K_M for the full-length Rabex and for the HB-Vps9 tandem with Rab5(wt) as substrate [11]. By taking the ratio between the two parameters to compare with the other reference values, we see that the catalytic efficiency is still approximately $10^4\,M^{-1}s^{-1}$.

In contrast, our reference rate value k_7^B (k_7 in Table 5.2) is $10^8\,M^{-1}s^{-1}$. As described in Section 5.1, we consider $k_7 = k_7^B \cdot K_M$ as catalytic rate for the GEF activity, hence for us the catalytic efficiency k_{cat}/K_M is constant and equal to k_7^B. This is much larger than the reported values. We know that the GEF is three times more active when coupled with its effector, Rabaptin5, but this is not enough to justify such a discrepancy [23]. Of course, the positive-feedback will further increase the GEF activity, but we didn't find reference for the catalytic activity of the whole complex Rab5:Rabex5:Rabaptin5 in the literature. Finally, we note that the value employed by *Bezeljak et al.* is chosen starting from a reference value that is similar to the those reported in Table 5.4.

In Table 5.5, instead, we collect various measurements of the activity of different GAP proteins. Some of them are expressed as k_{cat}/K_M and others are performed for the two parameters alone, i.e. k_{cat} and K_M. Since there are no direct measurements for Rab GAPs, we report some measurements for the GAPs of Ypt7p and Ypt51p, the corresponding proteins of Rab7 and Rab5, respectively, in *Saccharomyces cerevisiae* [1]. They also belong to the family of Rab GTPases, so they are good candidates for an estimation of the Rab5 GAP. From the measured values of k_{cat}, K_M we get a catalytic efficiency of the order of $10^4\, M^{-1}s^{-1}$, further confirmed by another reference value [25]. The measurements of k_{cat} and K_M for the Rab3 GAP lead to a bit smaller catalytic efficiency but still of the same order of magnitude [10]. Finally, higher GAP activities are measured for the Shark GAP and the Sus GAP with active Rab11 as substrate, reaching a catalytic efficiency of $10^5\, M^{-1}s^{-1}$.

Similarly to the GEF case, we can't modify the catalytic efficiency k_{cat}/K_M because of our parametrization of the inactivation catalytic rate (5.16) described in Section 5.1. It's a more complex expression compared to the GEF catalytic rate, but we still have a direct proportionality between k_{cat} and K_M, even if in this case it is not constant. Indeed, by modifying the GAP concentration we can tune this ratio, although not changing the catalytic ratio α^-, as described in the introduction of Chapter 6.

Enzyme	Substrate	Measured Parameter	Value	Reference
Rabex5(wt)	Rab5A(wt)	k_{cat}/K_M	$9 \cdot 10^5\, M^{-1}s^{-1}$	[4]
Rabex-5(D313A/Y345A)	Rab5A(wt)	k_{cat}/K_M	$7.7 \cdot 10^3\, M^{-1}s^{-1}$	[4]
Rabex-5(wt)	Rab5(wt)	k_{cat}/K_M	$2.5 \cdot 10^4\, M^{-1}s^{-1}$	[22]
Rabex-5(full length)	Rab5(wt)	k_{cat}	$0.007 s^{-1}$	[11]
Rabex-5(full length)	Rab5(wt)	K_M	270 nM	[11]
Rabex5(HB-Vps9 tandem)	Rab5(wt)	k_{cat}	$\geqslant 0.1 s^{-1}$	[11]
Rabex5(HB-Vps9 tandem)	Rab5(wt)	K_M	$\geqslant 2\,\mu M$	[11]

Table 5.4. GEF catalytic activity measurements. The values are taken from the references on the right, the exact location within the article is specified here: [4], paragraph "Impaired Rabex-5 GEF activity affects targeting of Rab5A", Results Section; [22], paragraph "Distinct roles for switch II residues in Rab activation", Results Section; [11], Discussion Section.

Enzyme	Substrate	Measured Parameter	Value	Reference
Gyp7-His$_6$	Ypt7p	K_M	$400\,\mu M$	[1]
Gyp7-His$_6$	Ypt7p	k_{cat}	$7.5 s^{-1}$	[1]
Gyp1-46p	Ypt51p	K_M	$143\,\mu M$	[1]
Gyp1-46p	Ypt51p	k_{cat}	$3.9 s^{-1}$	[1]
Rab3-GAP	Rab3	K_M	$75\,\mu M$	[10]
Rab3-GAP	Rab3	k_{cat}	$1 s^{-1}$	[10]
Gyp7	Ypt7	k_{cat}/K_M	$1.0 \div 1.5 \cdot 10^4\, M^{-1}s^{-1}$	[25]
Shark GAP	Rab11a·GTP	k_{cat}/K_M	$1.19 \cdot 10^5 M^{-1}s^{-1}$	[7]
Sus GAP	Rab11a·GTP	k_{cat}/K_M	$1.86 \cdot 10^5 M^{-1}s^{-1}$	[7]

Table 5.5. GAP catalytic activity measurements. The values are taken from the references on the right, the exact location within the article is specified here: [1], Table III: "Comparison of the catalytic properties of several GTPase activating proteins"; [10], paragraph "Catalytic Properties of Rab3-GAP", Results Section; [25], estimated from Figure 7(C) in the list of figures after the References Section; [7], paragraph "GAP assay of shark GAP and Sus GAP", Results Section.

Chapter 6

Parameter dependence

In the previous chapters, we illustrated the theoretical framework, assumptions, and goals of our study. In this chapter, we present the results of several simulations, performed with different initial conditions and parameter values, to analyze the contribution of the various reactants to domain formation and to discuss the possible biological and physical implications of these findings.

In particular, we focus on how variations in the concentration of key components affect Rab5 activation and the emergence of phase-separated domains.

One important observation concerns the catalytic activity of the inactivation process. Our parametrization of the reaction rates (5.18) relates the catalytic rate $k_c^- = k_8$ to the Michaelis-Menten constant K_M and to the total concentration of the GAP through equation (5.16), namely:

$$k_8 = k_3^B \frac{(K_d^{\mathrm{GAP}} + [b.s.])K_M}{\left[\mathrm{GAP}^{\mathrm{tot}}\right][b.s.]}$$

If we substitute this into the expression of α^-:

$$\alpha^- = \frac{k_8 \cdot [\mathrm{GAP}^{\mathrm{cyto}}]}{K_d^{\mathrm{GAP}}} \backsimeq \frac{k_3^B K_M}{[b.s.]} \tag{6.1}$$

where we approximate $[\mathrm{GAP}^{\mathrm{cyto}}] \approx [\mathrm{GAP}^{\mathrm{tot}}]$ because we assume that most GAP enzymes are in the cytosol, which is confirmed by our simulations (see Figure 4.4(B)). Moreover, we observe that $K_d^{\mathrm{GAP}} \gg [b.s.]$, indeed $K_d^{\mathrm{GAP}} = \frac{k_3}{k_6} = 2782.6\mathrm{nM}$ and $[b.s.] = 8.48\mathrm{nM}$ in all the simulations performed here.

Hence, the catalytic ratio of the GAP is determined by the reference rate k_3^B. The fact that increasing the GAP concentration does not increase the catalytic activity confirms that our parametrization succesfully reproduces the dynamics described by *Bezeljak et al.* [3]. From equation (6.1), we see that α^- can be tuned by varying our free parameter K_M, but it is important to note that this does not result in a different catalytic efficiency. In fact, the catalytic efficiency is generally measured through the parameter k_{cat}/K_M, but in our case $k_{\mathrm{cat}} \propto K_M$, meaning that the ratio remains constant.

In conclusion, we can modify the balance between the two catalytic processes only by acting on the GEF activity, which is not constrained by these limitations. This is due to the fact that we want to respect the reference's experimental set-up, but since they did not explicitly consider the GAP, we have to introduce it and meanwhile respect the catalytic power imposed by k_3^B.

The discussion above is important because the $\alpha^{\pm}$ coefficients determine the value of the diffusion coefficient D through the relation (5.24) and the fact that we can't vary significantly α^- (as modifying K_M does not affect the catalytic efficiency) results in a limitation of the accessible values that D can assume.

Figure 5.5 shows a plot of this relation for $w = 0.31\,\mu m, K_M = 200\,\mathrm{nM}$ and $c = 33.92\,\mathrm{nM}$, corresponding to the concentrations considered in Table 6.1. From this plot one can immediately notice that to get larger orders of magnitudes in the diffusion coefficient D one has to increase the value of the $\alpha^{\pm}$ coefficients, but we know that we can only act on the value of α^{+} coefficient. In conclusion, we can only increase the GEF concentration to obtain higher values of D. In the situation described in Table 6.1 we see that $D \sim 10^{-4}\,\mu m^2/s$ and, although higher GEF concentrations would allow for slightly larger diffusivities, this is not enough to reach values on the order of $10^{-3} \div 10^{-2}\,\mu m^2/s$ without compromising phase separation.

In fact, if we only increase the catalytic activity of the activation process, at some point there will be an excessive unbalance between the two catalytic processes, leading the system out of the bistability region. This means that we cannot explore the full estimated range of the diffusion coefficients (see Section 5.4) without losing the theoretical consistency provided by equation (5.24). For these reasons, in all simulations described in this chapter, we will consider $D \sim 10^{-4}\,\mu m^2/s$. This is not a problem, as we expect D to be very small on early endosomal membranes (see the discussion of Section 5.4).

6.1 Phase separation

The main objective of this thesis is to show that our computational model can exhibit phase separation between the two conformational states of Rab5, i.e., the domain formation of one species in a "sea" of the other. In particular, as described at the beginning of Chapter 5, we want to show that, by considering the reaction scheme proposed by *Bezeljak et al.*(2020) [3] and the reaction rates that they employed, we are able to obtain domain formation through our computational framework. This would represent an extension of their work, since they only investigated the collective activation of Rab5 on membranes and the associated stochasticity.

Furthermore, since they did not take into account the spatial component in their study, we also refer to the work of *Cezanne et al.*(2020) [6] to estimate the interface width w of a Rab5 domain formed in a reconstituted system. In this way, we can impose some constraints on the diffusion coefficient D, based on our theoretical relationship between D and w (see Section 5.3).

Here, we propose a set of parameters that leads to phase separation:

Parameter	**Value**
[Rab5: GDP]	$16.96\,\mathrm{nM}$
[Rab5: GTP]	$16.96\,\mathrm{nM}$
$[b.s.]$	$8.48\,\mathrm{nM}$
[GEF(c)]	$100\,\mathrm{nM}$
[GAP(c)]	$50\,\mathrm{nM}$
D	$2.15 \cdot 10^{-4}\,\mu m^2/s$
[GDI]	$300\,\mathrm{nM}$
r_{endo}	$5\,\mu m$
r_{cell}	$17\,\mu m$

Parameter	**Value**
k_1	$0.0675\,s^{-1}$
k_2	$0.08\,s^{-1}$
k_3	$0.08\,s^{-1}$
k_4	$0.0029\,s^{-1}\mathrm{nM}^{-1}$
k_5	$0.00002875\,s^{-1}\mathrm{nM}^{-1}$
k_6	$0.00002875\,s^{-1}\mathrm{nM}^{-1}$
k_7	$20\,s^{-1}$
k_8	$0.69\,s^{-1}$
K_M	$200\,\mathrm{nM}$

Table 6.1. (left) Reactant concentrations, diffusion coefficients and radius employed in this simulation. (right) Reaction rates and Michaelis-Menten constants (equal for both catalytic reactions) used for this simulation. The GEF(c) was called RR in Section 5.1, but we use it to indicate the full complex as well.

Each site of the membrane lattice is populated with 20 Rab5:GDP molecules, 20 Rab5:GTP molecules, and 10 (free) binding sites. One could argue that this represents a low number of binding sites with respect to the other species concentrations, but the simulation results indicate that the number of binding sites is not a limiting factor. The reason is that Rab5 can dissociate from the membrane, thus creating new binding sites.

The reaction rates are obtained according to the parametrization (5.18) discussed in Section 5.1. For example, the value of k_1 is determined by the GDI concentration, which is not an explicit reactant in our scheme. The Michaelis-Menten constant K_M sets the value of the two catalytic rates (k_7, k_8). Since it is not considered in either of our reference papers, it's a free parameter for us. Given the fact that we cannot reproduce a catalytic efficiency comparable to those found in the literature for the reasons described in Section 5.5, we set it to the value $K_M = 200\text{nM}$. This value represents a compromise between the fact that we assume $K_M \gg c$, where $c = [\text{Rab5: GDP}] + [\text{Rab5: GTP}]$, in both our parametrization and to obtain the relation (3.21) that we used to constrain the values of the diffusivity, and the fact that we don't want to take a too much high $\kappa = K_M / c$ because it would result in even smaller values of D.

Moreover, by computing the $\alpha^{\pm}$ coefficients we determine the value of the diffusion coefficient D through the relation (5.24) for the membrane-bound species. We chose the central value of the estimated interface width $w = 0.31\,\mu m$ to compute D (see Section 5.3). In order to be consistent with this procedure, we have to set $r_{\text{endo}} = 5\,\mu m$, corresponding to the radius of the MCBs used by *Cezanne et al.* in their experiments. Finally, the cell radius is obtained through the procedure described in Section 5.2, i.e., $r_{\text{cell}} = r_{\text{endo}} + \sqrt{D \cdot \tau}$ where $D = 10\,\mu m^2/s$ and $\tau = k_1^{-1}$, since k_1 is the smallest dissociation rate.

The parameter values reported in Table 6.1 lead to the following temporal evolution:

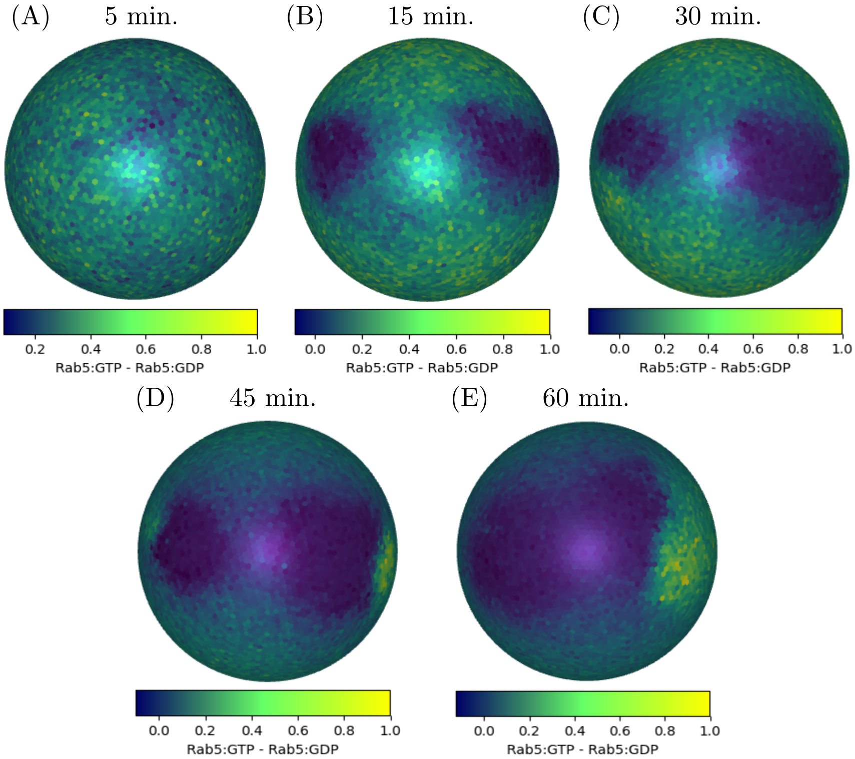


Figure 6.1. Pictures of the system at different times of its evolution. The order parameter visualized here is the difference between active and inactive populations, i.e., Rab5: GTP − Rab5: GDP. The color scale should be interpreted according to the individual colorbar shown for each frame (see Section 4.3 for further details on the output visualization, and also Appendix A for an insight of the algorithm).

In Figure 6.1(E) we can easily identify a domain enriched in Rab5:GDP (in dark blue), surrounded by Rab5:GTP proteins, corresponding to the steady state reached after approximately 1 hour of simulation. The dynamics leading to the formation of this single domain are widely described in the literature: the beginning of the process is characterized by the nucleation of a germ of the more stable phase (according to the reaction kinetics used); then, this germ can grow, competing with other growing germs (coarsening), and eventually fuses with other domains into a larger and more stable domain (coalescence) [14].

The fact that the majority of Rab5 proteins on the membrane are active (see Figure 4.4(B)) is probably due to the fact that the reaction rates employed by *Bezeljak et al.* were specifically designed to favor the activation process, since their aim was to observe the collective activation of Rab5 on membranes. However, we show that the chosen parameter set corresponds to a region of bistability in the phase diagram, according to our model.

6.2 The role of GDI in domain formation

Taking the parameter set presented in Table 6.1 as a reference, we want to investigate the effect of GDI on the process of domain formation. To do this, we vary its concentration over a range of values and analyze the simulation results.

It is important to recall that GDI concentration modulates the parametrization of the dissociation rate of Rab5:GDP from the membrane, since it is not treated as an explicit reactant in our model (see Section 5.1). Hence, we must update the rate constant accordingly in each simulation. Moreover, we also estimate the effective cytosolic volume by means of the expected timescale for the dissociation process to take place, following the approach described in detail in Section 5.2. This leads to the expression:

$$r_{\text{cell}} = r_{\text{endo}} + \sqrt{D \cdot \tau}$$

where $\tau = k_{\text{diss}}^{-1}$ and k_{diss} is the slowest dissociation rate in our system.

In the reference parameter set, the smallest rate is k_1, i.e. the dissociation rate parametrized by GDI, while the other two are both set to $0.08\,s^{-1}$. This means that when we consider GDI concentrations such that k_1 becomes larger than the other two, we have to use those rates to set r_{cell} and, as a consequence, the effective volume. However, since the radius obtained from $k_{\text{diss}} = 0.08\,s^{-1}$ is $r_{\text{cell}} = 16\,\mu m$, very similar to the one set by k_1 in the reference case, we can ignore this modification, especially because we prefer to overestimate the effective volume, as explained in Section 5.2.

When instead we consider smaller GDI concentrations, which result in an even smaller k_1, we should update the effective volume through the described procedure. However, we decide to keep $r_{\text{cell}} = 17\,\mu m$ for all the simulations for two reasons: because our algorithm runs very slowly with large volumes and because considering such a large volume when the GDI concentration is small appears to be an artifact of our parametrization, rather than a biologically realistic effect. In fact, we do not expect that a small concentration of GDI would result in such a large expected time for the association/dissociation process to and from the membrane. Furthermore, it's reasonable to work with a volume set by dissociation rates that are comparable to each other.

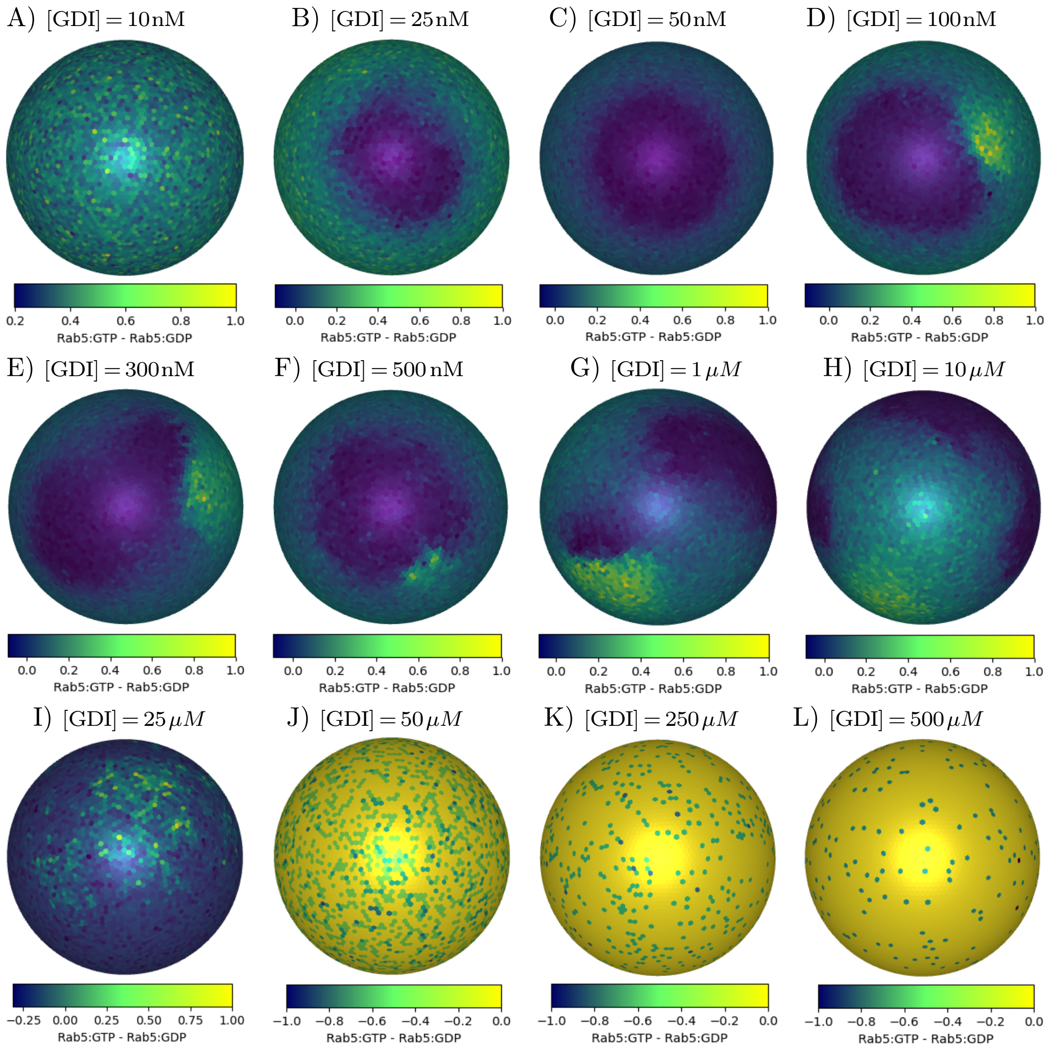
In summary, all simulations share the same parameter set presented in Table 6.1, except for k_1, which varies as a function of GDI concentration according to our parametrization:

$$k_1 = k_2^B \cdot [\text{GDI}]$$

The concentrations considered for GDI range from 10 nM to 500 µM and the simulation results, at steady state, are shown in Figure 6.2. We observe that when GDI concentration is not enough, the system remains homogeneous and fails to break symmetry (Figure 6.2(A)). Above a certain threshold, phase separation is observed, further confirming the results from *Cezanne et al.* [6], who demonstrated that GDI is necessary for domain formation (Figure 6.4).

Its main role is to guarantee an efficient flux between cytosolic and membrane-bound Rab5, allowing for continuous delivery and extraction of inactive Rab5 to and from the membrane. This dynamic exchange is essential for domain formation, as it supports the spatial redistribution of Rab5 on membrane.

Similarly, the system does not exhibit phase separation even for large concentrations, thus providing evidence for a non-monotonic dependence on GDI concentration: not only the system does not exhibit phase separation if there is not enough GDI, the symmetry breaking is also suppressed by an excess of GDI (Figure 6.2(I-L)). This behavior further supports the importance of a proper balance between the constitutent elements of the system for the emergence of this kind of phenomena.



A) $[\mathrm{GDI}] = 10\mathrm{nM}$ B) $[\mathrm{GDI}] = 25\mathrm{nM}$ C) $[\mathrm{GDI}] = 50\mathrm{nM}$ D) $[\mathrm{GDI}] = 100\mathrm{nM}$

E) $[\mathrm{GDI}] = 300\mathrm{nM}$ F) $[\mathrm{GDI}] = 500\,\mathrm{nM}$ G) $[\mathrm{GDI}] = 1\,\mu M$ H) $[\mathrm{GDI}] = 10\,\mu M$

I) $[\mathrm{GDI}] = 25\,\mu M$ J) $[\mathrm{GDI}] = 50\,\mu M$ K) $[\mathrm{GDI}] = 250\,\mu M$ L) $[\mathrm{GDI}] = 500\,\mu M$

Figure 6.2. The images shown in this figure represent the system at steady state, for different GDI concentrations. The intensity of the order parameter ([Rab5: GTP] − [Rab5: GDP]) is shown, representing one of our two options for visualization, as described in Section 4.3. The time needed to reach steady state is not always the same; we waited until it was reached and then took a snapshot of the system. Here we report the times corresponding to the depicted states of the system, for each simulation: A) 20 min; B) 130 min; C) 135 min; D) 60 min; E) 60 min; F) 75 min; G) 45 min; H) 30 min; I) 40 min; J) 15 min; K) 5 min; L) 5 min.

We also report the simulation results by showing the total amount of Rab5 on the membrane at steady state, instead of the order parameter. These are shown in Figure 6.3 and are very similar to those depicted in Figure 6.2. The differences in colour in the images are due to the choice of the visualized parameter, but as we can see, the Rab5 distribution on the membrane is qualitatively the same. We use these images to compare with the experimental observations by *Cezanne et al.*, reported in Figure 6.4, since they label all Rab5 molecules, both active and inactive, with GFP.

In summary, despite the underlying simplifications of our model, in which GDI is an implicit reactant, our simulations succeed in reproducing the essential features experimentally observed. We confirm not only that GDI is necessary for domain formation, but also that there exists an optimal range of its concentration in which the system is able to exhibit these patterns. The emergence of domains requires a balance between membrane recruitment and extraction.

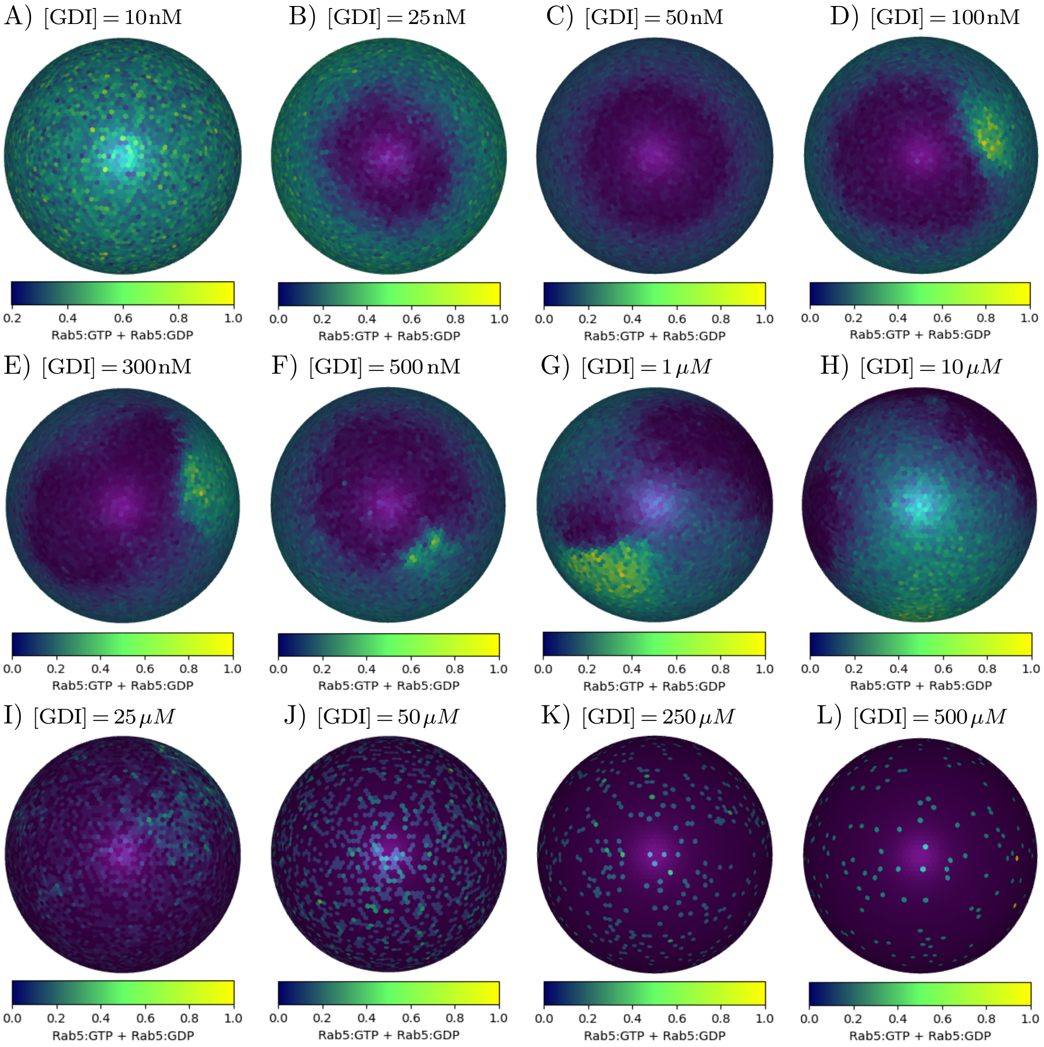


Figure 6.3. Steady-states of the system for different GDI concentrations. The total amount of Rab5 for each site is displayed here. The timescales needed to reach steady-states are reported in Figure 6.2.

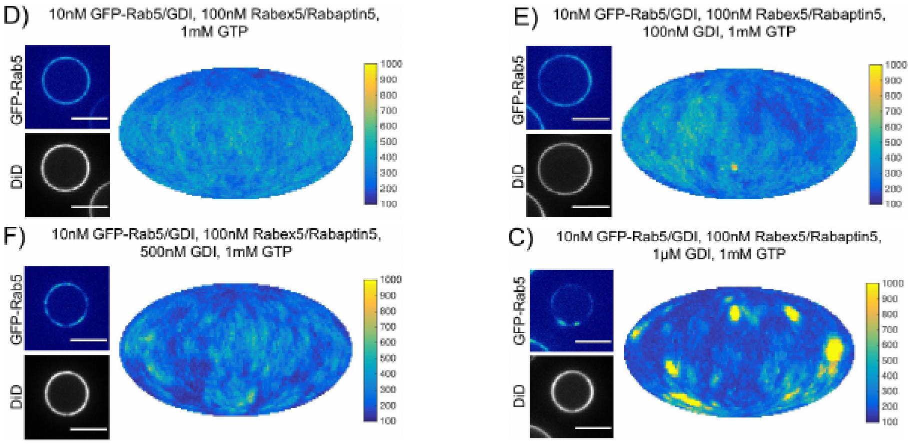

D)
10nM GFP-Rab5/GDI, 100nM Rabex5/Rabaptin5, 1mM GTP
E)
10nM GFP-Rab5/GDI, 100nM Rabex5/Rabaptin5, 100nM GDI, 1mM GTP
GFP-Rab5
DiD
1000
900
800
700
600
500
400
300
200
100
F)
10nM GFP-Rab5/GDI, 100nM Rabex5/Rabaptin5, 500nM GDI, 1mM GTP
C)
10nM GFP-Rab5/GDI, 100nM Rabex5/Rabaptin5, 1μM GDI, 1mM GTP

Figure 6.4. The images are taken from the paper of *Cezanne et al.* [6], Figure 2 C),D),E),F). They reconstituted Rab5 domain formation in vitro by incubating EE MCBs for 15 min at 23°C with 10 nM GFP-Rab5/GDI, 100 nM Rabex5/Rabaptin5, 1 mM GTP and D) 0 nM GDI; E) 100 nM GDI; F) 500 nM GDI; C) 1 μM GDI. The experiments showed that GDI is necessary for domain formation.

6.3 The dependence on GEF concentration

Cezanne et al. also demonstrated that the Rabex5/Rabaptin5 complex is essential for Rab5 domain formation in vitro [6]. Figure 6.7 shows their experimental results: they observed that the emergence of Rab5 patterns requires the GEF/effector complex in a concentration-dependent manner. Furthermore, as already mentioned in Section 2.3, they also proved that the full complex, and not one of its components, is necessary for domain formation.

In this section we show the results of simulations performed with varying GEF concentrations and compare them with the experimental results discussed above. As with the analysis of the role of GDI, the reference will be the parameter set in Table 6.1. For clarity, recall that the term 'GEF' refers to the full GEF/effector complex Rabex5/Rabaptin5, as illustrated in Section 5.1.

To do this, we recall that the GEF concentration is involved in the definition of the catalytic ratio α^+, which sets the value of the membrane diffusivity D through equation (5.24), as described in detail in Section 5.3. This means that we have to update the value of D in each simulation. As already fully explained at the beginning of Section 6, we cannot work with orders of magnitude higher than $10^{-4}\ \mu m^2/s$ for D, because of limitations in the tunability of the catalytic ratio associated with the inactivating process, i.e., α^-.

The simulation results are presented in Figure 6.5. The GEF concentrations employed for the simulations range from 1 to 350 nM. Similarly to GDI, a non-monotonic dependence on GEF concentration is observed, as the system breaks the symmetry with a sufficient, but not excessively high, amount of GEF. Indeed, with 1 nM GEF, the inactivating power is completely dominant and we have zero Rab5:GTP at steady state (Figure 6.5(A)); by slightly increasing the GEF concentration we observe a finite population of active Rab5 at equilibrium, but the system cannot fully develop patterns. At steady state, the percentage associated with the first twenty spherical harmonics remains around 48% without increasing further, indicating that some spatial organization has been reached (Figure 6.5(B)). With 10 nM, however, we observe phase separation, and 80% of the total weight is associated with the first twenty spherical harmonics of the spatial organization (Figure 6.5(C)). Symmetry breaking occurs at GEF concentrations between 10 nM and 300 nM (Figure 6.5(C-G)) but is no longer observed with 350 nM (Figure 6.5(H)).

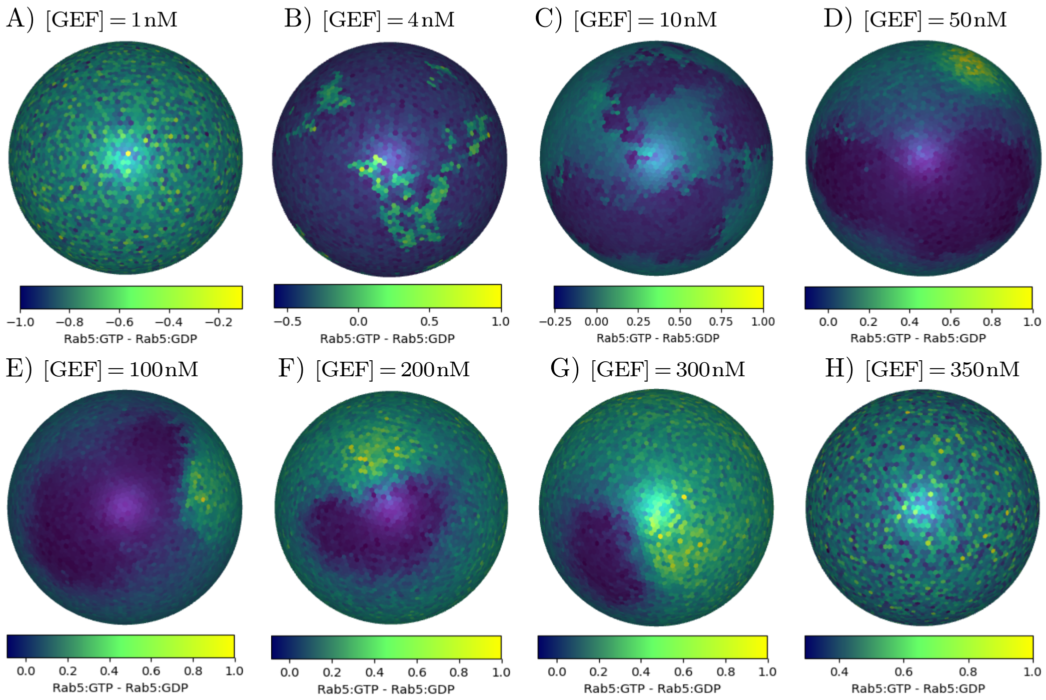


Figure 6.5. The simulation results are visualized on a sphere, as described in Section 4.3. Here, we report the steady states of the simulations run at several different GEF concentrations. In these simulations, we visualize the order parameter [Rab5: GTP] − [Rab5: GDP]. The snapshots of the system were taken at the following times: A) 10 min; B) 100 min; C) 80 min; D) 60 min; E) 60 min; F) 60 min; G) 130 min; H) 15 min. We waited until the system reached the steady state; therefore, these times are good indicators of the timescale required for the system to reach equilibrium.

We also report the values of the diffusion coefficients used in the simulations, which are determined by the α^+ coefficients, according to equation (5.24): A) $D = 6 \cdot 10^{-6}\,\mu m^2/s$; B) $D = 1.2 \cdot 10^{-5}\,\mu m^2/s$; C) $D = 2.5 \cdot 10^{-5}\,\mu m^2/s$; D) $D = 1.09 \cdot 10^{-4}\,\mu m^2/s$; E) $D = 2.15 \cdot 10^{-4}\,\mu m^2/s$; F) $D = 4.26 \cdot 10^{-4}\,\mu m^2/s$; G) $D = 6.37 \cdot 10^{-4}\,\mu m^2/s$; H) $D = 7.43 \cdot 10^{-4}\,\mu m^2/s$.

The same simulation results are visualized using the total amount of Rab5 at each site (see Figure 6.6). As in the GDI case, the images are very similar to those generated by visualizing the order parameter at each site. However, they are important because we still use the work by *Cezanne et al.* as a reference and, as already said in the previous section, they track the fluorescence intensity of the total Rab5 density, since all Rab5 molecules are GFP-labeled. By looking at Figure 6.7, we see that our results confirm the experimental results, i.e. the complex Rabex5/Rabaptin5 is essential for domain formation and the phenomenon depends on GEF concentration.

Furthermore, as outlined above, we also observe that the system does not exhibit phase separation when the GEF concentration is too large. This further confirms the importance of a proper balance between the system components for the emergence of such phenomena. In particular, in this case, it is a finely tuned interplay between activation and inactivation rates that is crucial for domain formation.

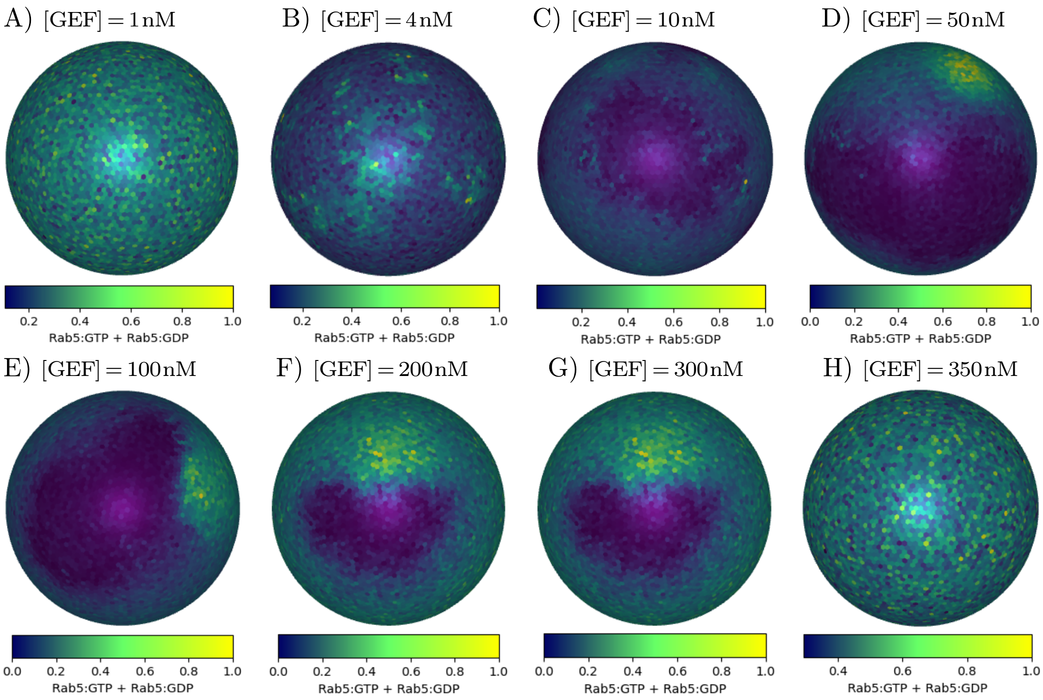


Figure 6.6. Steady-state solutions for different GEF concentrations. In this Figure we visualize the total amount of Rab5 for each site, regardless of its activation state. The considered times and diffusion coefficients for each simulation correspond to those in Figure 6.5.

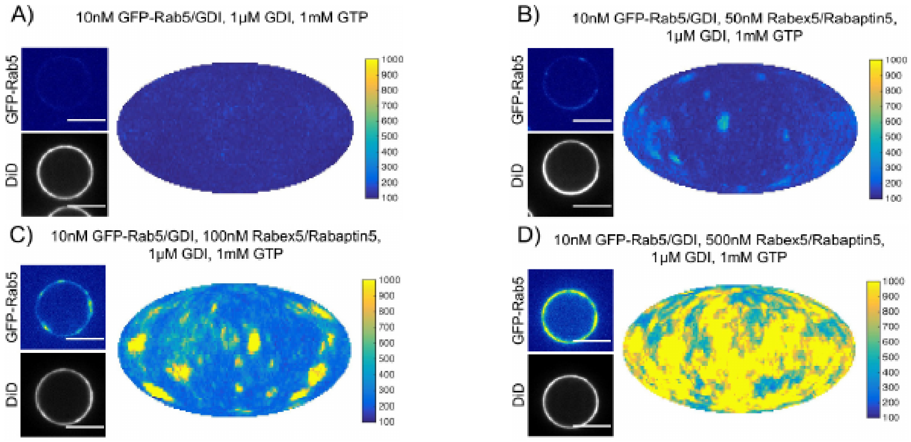

A) 10nM GFP-Rab5/GDI, 1µM GDI, 1mM GTP
GFP-Rab5
DiD
1000 900 800 700 600 500 400 300 200 100
B) 10nM GFP-Rab5/GDI, 50nM Rabex5/Rabaptin5, 1µM GDI, 1mM GTP
GFP-Rab5
DiD
1000 900 800 700 600 500 400 300 200 100
C) 10nM GFP-Rab5/GDI, 100nM Rabex5/Rabaptin5, 1µM GDI, 1mM GTP
GFP-Rab5
DiD
1000 900 800 700 600 500 400 300 200 100
D) 10nM GFP-Rab5/GDI, 500nM Rabex5/Rabaptin5, 1µM GDI, 1mM GTP
GFP-Rab5
DiD
1000 900 800 700 600 500 400 300 200 100

Figure 6.7. The images are taken from the paper of *Cezanne et al.* [6], Figure 3 A),B),C),D). They show that domain formation depends on concentration of Rabex5/Rabaptin5 and that the complex is essential for the process. As for GDI results, they incubated EE MCBs for 15 min at 23°C with 10 nM GFP-Rab5/GDI, 100 nM Rabex5/Rabaptin5 (RR), 1 mM GTP and A) 0 nM RR; B) 50 nM RR; C) 100 nM RR; D) 500 nM RR.

Chapter 7
Conclusions

In this thesis, we investigated the mechanisms underlying domain formation on cell membranes, a complex process relevant to *self-organization* and intracellular signaling, having several analogies with *phase separation*. To this aim, we adopted a lattice-gas model to simulate reaction-diffusion dynamics on a spherical membrane in contact with a cytosolic protein reservoir. This approach allowed us to study the conditions under which spatiotemporal patterns of Rab5 emerge and are sustained.

Our simulations revealed that within a specific range of parameters, the system spontaneously breaks symmetry and exhibits stable domains, in which one conformational state of Rab5 (either active or inactive) accumulates in a background dominated by the opposite state. These domains arise through a combination of positive feedback loops, diffusion on the membrane and continous exchange with the cytosol. Importantly, we also observed that the phenomenon occurs over a broad range of GEF and GDI concentrations, thus showing that it is robust to fluctuations.

An important aspect of this work is that both the reaction scheme and the associated rates were derived from *Bezeljak et al.* [3] through a systematic parametrization. This approach ensures that our simulation results reflect biologically plausible regimes, thereby giving credibility to the observed results. Furthermore, we provide a potential extension of their findings by obtaining domain formation starting from their reaction rates. While their study focused on the collective activation of Rab5 on the membrane without considering spatial distribution, we show that domain formation can be obtained by including the contribution of diffusivity in their biochemical network.

To incorporate diffusion in a biologically meaningful way, we used the experimental observations from *Cezanne et al.* [6] as a reference to estimate the typical interface width of a Rab5 domain. From this, we derived a range of values for the diffusion coefficient through a relation obtained within our analytical model. Beyond parameter estimation, we used the same reference to qualitatively compare our simulation results with their fluorescence microscopy images, focusing in particular on the dependence of the phenomenon on the concentration of some of its components, such as GDI and GEF. Consistently with their observations, our model shows that the formation of Rab5 domains emerge beyond a certain threshold concentration of these regulators. Furthermore, it also predicts that domain formation is suppressed at very high concentrations, revealing a non-monotonic behavior.

One limitation of our work is that, due to constraints imposed by our parametrization, some parameters such as the catalytic ratio α^- of the inactivation process, are not freely tunable, as their defining rates are interdependent. As a result, we cannot explore the behavior of the system for high diffusivity without losing theoretical consistency. Moreover, large D values lead to significantly slower simulations, so that we could not investigate the system in the regime of high diffusivity even if we relax our theoretical constraints.

The minimal nature of our model reveals that complex spatial organization can arise from fundamental reaction-diffusion principles. Importantly, this approach can be extended to study other self-organizing systems beyond Rab5, highlighting the universality of the underlying mechanism. The simulation algorithm is also easily generalizable, making it suitable for studying other systems exhibiting similar biochemical dynamics.

Bibliography

[1] Štefan Albert, Elke Will, and Dieter Gallwitz. Identification of the catalytic domains and their functionally critical arginine residues of two yeast GTPase-activating proteins specific for Ypt/Rab transport GTPases. *The EMBO journal*, 1999.

[2] Martine Arrio-Dupont, Georges Foucault, Monique Vacher, Philippe F Devaux, and Sophie Cribier. Translational diffusion of globular proteins in the cytoplasm of cultured muscle cells. *Biophysical journal*, 78(2):901–907, 2000.

[3] Urban Bezeljak, Hrushikesh Loya, Beata Kaczmarek, Timothy E Saunders, and Martin Loose. Stochastic activation and bistability in a Rab GTPase regulatory network. *Proceedings of the National Academy of Sciences*, 117(12):6540–6549, 2020.

[4] Julia Blümer, Juliana Rey, Leif Dehmelt, Tomáš Mazel, Yao-Wen Wu, Philippe Bastiaens, Roger S Goody, and Aymelt Itzen. RabGEFs are a major determinant for specific Rab membrane targeting. *Journal of Cell Biology*, 200(3):287–300, 2013.

[5] Cecilia Bucci, Robert G Parton, Ian H Mather, Henk Stunnenberg, Kai Simons, Bernard Hoflack, and Marino Zerial. The small GTPase rab5 functions as a regulatory factor in the early endocytic pathway. *Cell*, 70(5):715–728, 1992.

[6] Alice Cezanne, Janelle Lauer, Anastasia Solomatina, Ivo F Sbalzarini, and Marino Zerial. A non-linear system patterns Rab5 GTPase on the membrane. *Elife*, 9:0, jun 2020.

[7] Yan-Na Chen, Xin Gu, X Edward Zhou, Weidong Wang, Dandan Cheng, Yinghua Ge, Fei Ye, H Eric Xu, and Zhengbing Lv. Crystal structure of TBC1D15 GTPase-activating protein (GAP) domain and its activity on Rab GTPases. *Protein Science*, 26(4):834–846, 2017.

[8] Jacqueline Cherfils and Mahel Zeghouf. Regulation of small gtpases by gefs, gaps, and gdis. *Physiological reviews*, 93(1):269–309, 2013.

[9] Richard J Cherry. Rotational and lateral diffusion of membrane proteins. *Biochimica et Biophysica Acta (BBA)-Reviews on Biomembranes*, 559(4):289–327, 1979.

[10] Aude Clabecq, Jean-Pierre Henry, and Franc ois Darchen. Biochemical characterization of Rab3-GTPase-activating protein reveals a mechanism similar to that of Ras-GAP. *Journal of Biological Chemistry*, 275(41):31786–31791, 2000.

[11] Anna Delprato, Eric Merithew, and David G Lambright. Structure, exchange determinants, and family-wide rab specificity of the tandem helical bundle and Vps9 domains of Rabex-5. *Cell*, 118(5):607–617, 2004.

[12] Eileen Edler, Eric Schulze, and Matthias Stein. Membrane localization and dynamics of geranylgeranylated Rab5 hypervariable region. *Biochimica et Biophysica Acta (BBA)-Biomembranes*, 1859(8):1335–1349, 2017.

[13] AV Filippov, MA Rudakova, and BV Munavirov. Lateral diffusion in sphingomyelin bilayers. *Magnetic Resonance in Chemistry*, 48(12):945–950, 2010.

[14] Elisa Floris, Andrea Piras, Luca Dall'Asta, Andrea Gamba, Emilio Hirsch, and Carlo C Campa. Physics of compartmentalization: How phase separation and signaling shape membrane and organelle identity. *Computational and Structural Biotechnology Journal*, 19:3225–3233, 2021.

[15] Andrea Gamba, Antonio De Candia, Stefano Di Talia, Antonio Coniglio, Federico Bussolino, and Guido Serini. Diffusion-limited phase separation in eukaryotic chemotaxis. *Proceedings of the National Academy of Sciences*, 102(47):16927–16932, 2005.

[16] J Shawn Goodwin, Kimberly R Drake, Catha L Remmert, and Anne K Kenworthy. Ras diffusion is sensitive to plasma membrane viscosity. *Biophysical journal*, 89(2):1398–1410, 2005.

[17] Jacob Halatek, Fridtjof Brauns, and Erwin Frey. Self-organization principles of intracellular pattern formation. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 373(1747):20170107, 2018.

[18] Hisanori Horiuchi, Roger Lippé, Heidi M McBride, Mariantonietta Rubino, Philip Woodman, Harald Stenmark, Vladimir Rybin, Matthias Wilm, Keith Ashman, Matthias Mann et al. A novel Rab5 GDP/GTP exchange factor complexed to Rabaptin-5 links nucleotide exchange to effector recruitment and function. *Cell*, 90(6):1149–1159, 1997.

[19] Anthony A Hyman, Christoph A Weber, and Frank Jülicher. Liquid-liquid phase separation in biology. *Annual review of cell and developmental biology*, 30(1):39–58, 2014.

[20] Ken Jacobson and John Wojcieszyn. The translational mobility of substances within the cytoplasmic matrix. *Proceedings of the National Academy of Sciences*, 81(21):6747–6751, 1984.

[21] Karl Kerényi. *Dioniso: archetipo della vita indistruttibile*. Adelphi, Milano, 1992.

[22] Lars Langemeyer, Ricardo Nunes Bastos, Yiying Cai, Aymelt Itzen, Karin M Reinisch, and Francis A Barr. Diversity and plasticity in Rab GTPase nucleotide release mechanism has consequences for Rab activation and inactivation. *Elife*, 3:0, 2014.

[23] Roger Lippé, Marta Miaczynska, Vladimir Rybin, Anja Runge, and Marino Zerial. Functional synergy between Rab5 effector Rabaptin-5 and exchange factor Rabex-5 when physically associated in a complex. *Molecular biology of the cell*, 12(7):2219–2228, 2001.

[24] Eileen Münzberg and Matthias Stein. Structure and dynamics of mono-vs. doubly lipidated Rab5 in membranes. *International journal of molecular sciences*, 20(19):4773, 2019.

[25] Daniel P Nickerson, Matthew RG Russell, Shing-Yeng Lo, Hannah C Chapin, Joshua Milnes, and Alexey J Merz. Msb3/Gyp3 GAP selectively opposes Rab5 signaling at endolysosomal organelles. *Traffic (Copenhagen, Denmark)*, 13(10):1411, 2012.

[26] Erwin Schrödinger. *What is Life?* Cambridge University Press, Cambridge, 1944.

[27] Brian D Slaughter, Jay R Unruh, Arupratan Das, Sarah E Smith, Boris Rubinstein, and Rong Li. Non-uniform membrane diffusion enables steady-state cell polarization via vesicular trafficking. *Nature communications*, 4(1):1380, 2013.

[28] J Peter Slotte. Biological functions of sphingomyelins. *Progress in lipid research*, 52(4):424–437, 2013.

[29] URL: https://www.uniprot.org.

[30] Marino Zerial and Heidi McBride. Rab proteins as membrane organizers. *Nature reviews Molecular cell biology*, 2(2):107–117, 2001.

Appendix A

Here, we report the details of the algorithm used for the simulations. The following is the main code, it is written in c and it calls several functions that are defined in other files contained in the same folder. We report only one of them, the one that saves the state of the system in textual files, in order not to have such a large Appendix. Moreover, the missing ones are not so essential to understand what the algorithm does. The most important functions called by the code are "readreactions" and "printreactions"; their aim is to read an external file containing all the necessary information of our system and print them on screen.

After the evolution code and the function saving the simulation results, we also show the Jupyter Notebook code, written using Python language, whose aim is to reproduce the textual informations, generated by the simulation's algorithm, into 3D images (see Section 4.3).

- Algorithm that performs simulations:

```
#define END           -1000000
#define DELTAT       800
#define TMAX         100000
#define TSTART       0
#define PRINTHIST    1000
#define LMAX         20
#define N_SITES         10242
#define MAX_REACTIONS   35
#define MAX_REACTANTS   20
#define MAX_CHEMICALS   4
#define H               6
#define  CREATION    1
#define  DESTRUCTION 2
#define  ADSORPTION  3
#define  DESORPTION  4
#define  CATALYSIS   5
#define  DIFFUSION   6
#define  MUTATION    7
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

extern int     *ivector    (int imin,int imax);
extern double  *dvector    (int imin,int imax);
extern int     **imatrix   (int imin,int imax,int jmin,int jmax);
extern double  **dmatrix   (int imin,int imax,int jmin,int jmax);
extern FILE    *fopenf     (char *format,char *mode,...);
extern void    error       (char *format,...);
extern double  Xrandom     (void);
extern void    winit       (int n,int max);
extern void    wzero       (int i);
extern void    wpush       (int i,int x,double wnew);
extern void    wdelta      (int i,int x,double del);
extern int     wpop        (int i);
extern double  wread       (int i,int x);
extern int     cpuint      (void);
extern int     cputerm     (void);
extern void    harmonics   (int nx,double **X,int **N,int *S,int lmax,double **alm);
extern void    printeps(int **n,int *dim,double count,double t,int nfil);
void readreactions  (int *dim, int **dimrR, int *global, int *catalytic, double *kappa,double *Km, int **Kd,
int *incond, int **change, int **fz0, int **z0, int **r,int **R,int **n, double **X, double *r_cell, double
*r_endo);
void printreactions (int *dim,  int **dimrR, int *global, int *catalytic, double *kappa,double *Km, int **Kd,
int *incond, int **change, int **fz0, int **z0, int **r,int **R, int **n, double **X, double r_cell, double
r_endo);

int readhex(int **a){
   FILE *stream;
   int i,j,b;
   if ((stream=fopen("hex.nn.10242","r"))==NULL) {
      printf("Can’t␣load␣file␣"); printf("hex.nn.10242"); printf("\n"); exit(1);
   }
   for(j=1;j<N_SITES+1;j++){
   for(i=0;i<H;i++){
    fscanf(stream,"%d",&b);
    a[j][i]=b;
   }
   }
    fclose(stream);
    return 0;
}

double weight(int i,int x,int **r,int **n, double *Km,int *global) {
   double tmp;
   int itmp;
   switch( r[i][0] ) {
      case DESORPTION:     // (1) → (2) + (3)(cyto)
         if( x == 0 )   return 1.;
         else           return (double)n[x][ r[i][1] ];
         break;
      case ADSORPTION:      // (1)(cyto) + (2) → (3)
         if( x == 0 )   return (double)n[0][ r[i][1] ];
         else           return (double)n[x][ r[i][2] ];
         break;
      case CATALYSIS:      // (1) + (3) → (2) + (3)
         if(global[ itmp=r[i][1] ]==1)
            {
               if( x == 0 )   return 1.;
               else
                       {
                          tmp=n[x][itmp];
                          return n[x][ r[i][3] ]*tmp / (Km[i]+tmp);   //M.M
                       }
            }
         if(global[ itmp=r[i][1] ]==0)
            {
               tmp = n[0][ itmp ]/(double)N_SITES;
               if( x == 0 )   return tmp;
               else return  n[x][ r[i][3] ] / (Km[i]+tmp);
                 }
         break;
      case DIFFUSION: // (1)(x) → (1)(y)
         if( x == 0 )   return 1.;
         else           return (double)n[x][ r[i][1] ];
         break;
      case MUTATION: // (1) → (2)
         if( x == 0 )   return 1.;
         else           return (double)n[x][ r[i][1] ];
         break;
      default:
         printf("Error␣%d␣%d\n",r[i][0],i);
         break;
   }
  return 0.;
}

void wsav(int x, int j, double *tmp, int **r, int **n,int **R,double *Km,int *global) {
   int l;
   for( l=1 ; l <= R[0][j] ; l++ ) tmp[l] = -weight(R[l][j],x,r,n,Km,global);
}

void wmod(int x, int j, double *tmp,int **r,int **n,double *w,int **R,double *Km,int *global) {
   int l,m;
   for( l=1 ; l <= R[0][j] ; l++ ) {
      tmp[l] += weight(m=R[l][j],x,r,n,Km,global);//era tmp[l], uccidere wsav
      w[m] += tmp[l];
      wdelta(m,x,tmp[l]);
      }
   }

int main(int argc,char **argv) {

   int i,j,l,x,y,m,nx, nr, nR,count,nfil,flag,z,lmax;
   int *global, *dim, *ds, *catalytic, **Kd,**dimrR, *incond, **change,**n,**r,**R,**a,**N, **z0, **fz0,**nE,
*count_reactions;
   double xtmp,ytmp,ztmp,*tmp,delta,deltat,t,tmax,volume,surface,na,r_cell,r_endo;
   double *kappa, *Km, *w,*f,**X,**alm,**xE,**yE,**zE;
```

```
char filname[40];
FILE *stream,*stream1;

r_cell=0.0;
r_endo=0.0;
count_reactions=ivector(0,MAX_REACTIONS+1);
dim=ivector(0,4);
dimrR=imatrix(0,MAX_REACTIONS+MAX_REACTANTS,0,MAX_CHEMICALS+1);
global=ivector(0,MAX_REACTANTS+1);
catalytic=ivector(0,MAX_REACTIONS+1);
kappa=dvector(0,MAX_REACTIONS+MAX_REACTANTS+1);
Km=dvector(0,MAX_REACTIONS+1);
Kd=imatrix(0,1,0,MAX_REACTANTS+1);
w=dvector(0,MAX_REACTIONS+1);
f=dvector(0,MAX_REACTIONS+1);
incond=ivector(0,MAX_REACTANTS+1);
change=imatrix(0,MAX_REACTIONS,0,MAX_REACTANTS+1);
r=imatrix(0,MAX_REACTIONS+MAX_REACTANTS,0,MAX_CHEMICALS+1);
R=imatrix(0,MAX_REACTIONS+MAX_REACTANTS,0,MAX_REACTANTS+1);
n=imatrix(0,N_SITES+1,0,MAX_REACTANTS+1);
fz0=imatrix(0,MAX_REACTIONS,0,MAX_REACTANTS+1);
z0=imatrix(0,MAX_REACTIONS,0,MAX_REACTANTS+1);
ds=ivector(0,N_SITES+1);
X=dmatrix(0,2,0,N_SITES-1);
N=imatrix(0,5,0,N_SITES-1);
a=imatrix(0,N_SITES+1,0,H);
xE=dmatrix(0,MAX_REACTANTS,0,3*N_SITES);
yE=dmatrix(0,MAX_REACTANTS,0,3*N_SITES);
zE=dmatrix(0,MAX_REACTANTS,0,3*N_SITES);
nE=imatrix(0,MAX_REACTANTS,0,3*N_SITES);
tmp=dvector(0,MAX_REACTIONS+1);
alm=malloc((LMAX+1)*sizeof(double*));
init_random(150000,5);
for (l=0;l<=LMAX;l++) alm[l]=(double*)malloc((2*l+1)*sizeof(double))+l;

nx=N_SITES;

if ((stream=fopenf("hex.centers.%i","r",nx))==NULL) error("Need␣file␣'hex.centers.%i'",nx);
for (x=0;x<nx;x++)
  {
   for (i=0;i<3;i++) fscanf(stream,"%le",&X[i][x]);
  }
fclose(stream);


if ((stream1=fopenf("hex.nn.%i","r",nx))==NULL) error("Need␣file␣'hex.faces.%i'",nx);
for (x=0;x<nx;x++)
  {
   for (i=0;i<6;i++) fscanf(stream1,"%i",&N[i][x]);
  }
fclose(stream1);

readhex(a);

for(i=1;i<19;i++){
for(j=0;j<H;j++){
 printf("%6d",a[i][j]);
 }
printf("\n");
 }

readreactions(dim,dimrR,global,catalytic,kappa,Km,Kd,incond,change,fz0,z0,r,R,n,X,&r_cell,&r_endo);
printreactions(dim,dimrR,global,catalytic,kappa,Km,Kd,incond,change,fz0,z0,r,R,n,X,r_cell,r_endo);

nR=dim[0]+dim[3]-1;
nr=dim[1];
volume=(4.0/3.0)*M_PI*(pow(r_cell,3)-pow(r_endo,3));
surface=4*M_PI*pow(r_endo,2);
na=6.0221;
printf("\nvolume␣della␣cellula␣=␣%f␣µm\u00B3\n",volume);
printf("superficie␣dell'endosoma␣=␣%f␣µm\u00B2\n\n",surface);
printf("\n␣#␣of␣reactions␣+␣diffusions␣nR␣=␣%d\n␣#␣of␣reactants␣nr␣=␣%d\n␣#␣of␣sites␣nx␣=␣%d\n\n",nR,nr,nx);
deltat=DELTAT;
tmax=TMAX;
lmax=LMAX;
count=nfil=flag=0;
t=0.;

for( i=1 ; i <= nR ; i++ )
```

```
{
switch( r[i][0] ) {
   case DESORPTION:     // (1) → (2) + (3)(cyto)
      break;
   case ADSORPTION:     // (1)(cyto) + (2) → (3)
      kappa[i]/=(.1*volume*na);
      break;
   case CATALYSIS:      // (1) + (3) → (2) + (3)
      Km[i]*=(.1*na*volume/(double)N_SITES);
      break;
   case DIFFUSION: // (1)(x) → (1)(y)
      kappa[i]/=(sqrt(3)*surface/nx);
      break;
   case MUTATION: // (1) → (2)
      break;
   default:
      printf("Error␣%d␣%d\n",r[i][0],i);
      break;
}
}

winit(nR,nx);

for( i=1 ; i <= nR ; i++) {
   wzero(i);
   w[i] = 0;
   count_reactions[i] = 0;
}

for( i=1 ; i <= nR ; i++ ) {
   w[i] = 0;
   for( z=1 ; z <= nx ; z++ ) {
      xtmp = weight(i,z,r,n,Km,global);
      wpush(i,z,xtmp);
      w[i] += xtmp;
   }
}

while(1) {
   f[0]=0;
   for( i=1 ; i <= nR ; i++ ) {
      f[i] = kappa[i]*w[i]*weight(i,0,r,n,Km,global);
      f[0] += f[i];
   }
   xtmp=f[0]*Xrandom();
   i=1;
   while( (xtmp -= f[i]) > 0 ) i++;
   count_reactions[i] += 1;
   x=wpop(i);
   switch( r[i][0] ) {
      case DESORPTION:     // (1) → (2) + (3)(cyto)
      j=r[i][1];
         wsav(x, j,tmp, r, n,R,Km,global);
         n[0][j]--; n[x][j]--;
         wmod(x,j,tmp,r,n,w,R,Km,global);
      j=r[i][2];
         wsav(x, j,tmp, r, n,R,Km,global);
         n[0][j]++; n[x][j]++;
         wmod(x,j,tmp,r,n,w,R,Km,global);
      j=r[i][3];
         wsav(x, j,tmp, r, n,R,Km,global);
         n[0][j]++;
         wmod(x,j,tmp,r,n,w,R,Km,global);
      break;
      case ADSORPTION:     // (1)(cyto) + (2) → (3)
      j=r[i][1];
         wsav(x, j,tmp, r, n,R,Km,global);
         n[0][j]--;
         wmod(x,j,tmp,r,n,w,R,Km,global);
      j=r[i][2];
         wsav(x, j,tmp, r, n,R,Km,global);
         n[0][j]--; n[x][j]--;
         wmod(x,j,tmp,r,n,w,R,Km,global);
      j=r[i][3];
         wsav(x, j,tmp, r, n,R,Km,global);
         n[0][j]++; n[x][j]++;
         wmod(x,j,tmp,r,n,w,R,Km,global);
      break;
      case CATALYSIS:      // (1) + (3) → (2) + (3)
```

```
         j=r[i][1];
            wsav(x, j,tmp, r, n,R,Km,global);
            n[0][j]--; n[x][j]--;
            wmod(x,j,tmp,r,n,w,R,Km,global);
            if(n[0][j]<0)
            printf("**********************%d␣%d␣%f␣%f\n",j,n[0][j],w[i],weight(i,0,r,n,Km,global));
         j=r[i][2];
            wsav(x, j,tmp, r, n,R,Km,global);
            n[0][j]++; n[x][j]++;
            wmod(x,j,tmp,r,n,w,R,Km,global);
         break;
         case DIFFUSION:      // (1)(x) → (1)(y)
            l=H*Xrandom();
            y=a[x][l];
            while ( a[x][l] == 0 ){
               l=H*Xrandom();
               y=a[x][l];
            }
         j=r[i][1];
            wsav(x, j,tmp, r, n,R,Km,global);
            n[x][j]--;
            wmod(x,j,tmp,r,n,w,R,Km,global);
            wsav(y, j,tmp, r, n,R,Km,global);
            n[y][j]++;
            wmod(y,j,tmp,r,n,w,R,Km,global);
         break;
         case MUTATION: // (1) → (2)
            j=r[i][1];
            wsav(x, j,tmp, r, n,R,Km,global);
            n[0][j]--; n[x][j]--;
            wmod(x,j,tmp,r,n,w,R,Km,global);
         j=r[i][2];
            wsav(x, j,tmp, r, n,R,Km,global);
            n[0][j]++; n[x][j]++;
            wmod(x,j,tmp,r,n,w,R,Km,global);
         break;
      }
      delta=-log(Xrandom())/f[0];
      t+=delta;
      count++;
      if (cputerm()) break;
      if (cpuint()) flag=1;
      if (flag || t>TSTART+deltat) {
         deltat+=DELTAT;
         printeps(n,dim,count,t,nfil);
         for(x=0;x<nx;x++) ds[x]=n[x+1][2]-n[x+1][1];
         harmonics(nx,X,N,ds,lmax,alm);
         sprintf(filname,"%05d_harm",nfil); stream=fopen(filname,"a");
         fprintf(stream,"%10f␣",t);
         ztmp=0;
         for (l=0;l<=lmax;l++)
            {
            ytmp=0;
            for (m=-l;m<=l;m++)
               {
               xtmp=(m==0?1:2)*alm[l][m]*alm[l][m]/(4*M_PI);
               ytmp+=xtmp;
              }
              fprintf(stream,"%10f␣",ytmp);
              ztmp+=ytmp; // sum of the all C_l
           }
        fprintf(stream,"%10f\n",100*ztmp);
        printf("---␣%.2f%%␣weight␣on␣first␣%d␣harmonic␣components␣---␣\n",100*ztmp,lmax);
        fclose(stream);

        flag = 0;
      }
   }
   printf("\nNumero␣di␣esecuzioni␣per␣ogni␣reazione:\n");
   for (i=1; i <= MAX_REACTIONS; i++) {
      printf("Reazione␣%d:␣%d␣volte\n", i, count_reactions[i]);
   }
   return 0:
}
```

- Algorithm containing the function `printeps` called by the above algorithm:

```
#define MAXN 10242
#define MAXM 81920

#include <stdio.h>
#include <stdlib.h>
#include "writeeps.h"

void printeps(int **n,int *dim, double count,double t,int nfil)
  {
   int i,j,ntriangles,m,nr;
   int max=MAXN;
   int **N;
   FILE *strim, *strim1 ;
   char name[100], nami[100];

   nr=dim[1];

    if(count==0) {
            sprintf(name,"./State/%05d_Global_State.txt",nfil);
            sprintf(nami,"./State/%05d_Detailed_State.txt",nfil);
    }
       else {
               sprintf(name,"./State/%05d_Global_State_%02d:%02d:%02d.txt",
                            nfil,(int)(t/3600),((int)(t/60.)%60),((int)t)%60);
               sprintf(nami,"./State/%05d_Detailed_State_%02d:%02d:%02d.txt",
                            nfil,(int)(t/3600),((int)(t/60.)%60),((int)t)%60);
       }
     printf("Writing␣the␣state␣of␣the␣system␣on␣%s␣and␣%s\n",name, nami);

   // Check opening file of detailed_state
   if ((strim=fopen(nami,"w"))==NULL) {
      printf("error␣opening␣%s\n",nami);
      exit(1);
   }
   // writing the detailed state
   fprintf(strim,"%02d:%02d:%02d\t\n",(int)(t/3600),((int)(t/60.)%60),((int)t)%60);
   for(i=1;i<MAXN+1;i++) {
       for(j=1;j<nr+1;j++) {
        fprintf(strim,"%d\t",n[i][j]);
       }
       fprintf(strim,"\n");
   }
   fclose(strim);

   // Check opening file of global_state
   if ((strim1=fopen(name,"w"))==NULL) {
      printf("error␣opening␣%s\n",name);
      exit(1);
   }
   // writing the global state
   fprintf(strim1,"%02d:%02d:%02d\t\n",(int)(t/3600),((int)(t/60.)%60),((int)t)%60);
   for(m=1;m<nr+1;m++) fprintf(strim1,"%d\t",n[0][m]);
   fclose(strim1);

}
```

- Algorithm that produces images from the outputs of the simulation:

```
import igl
import numpy as np
import matplotlib.pyplot as plt
import pylab as pl
from meshplot import plot, subplot, interact
from mpl_toolkits import mplot3d
import pythreejs
from pythreejs import *
import ipywidgets
from IPython.display import display

path='/home/matteo/Documents/Materiale_Tesi/simula_bindsites/simula_bindsites_MATTE/State/'

from glob import glob
Glob_state_list=glob(path+'*00000_G*.txt*')
Glob_state_list.sort()
Glob_state=np.array(pl.loadtxt(Glob_state_list[0],skiprows=1)) # contains the global initial conditions
```

```
import re

def estrai_tempo(filename):
    match = re.search(r’(\d+):(\d+):(\d+)’, filename)
    if match:
        return tuple(map(int, match.groups()))
    else:
        return (0, 0, 0)

Glob_state_list.sort(key=estrai_tempo)

# Visualization of the global state for many times
# initialize the matrix of reactants R[i,j]=value of the reactant j at time i
R=np.zeros((len(Glob_state_list),len(Glob_state)))
time=np.linspace(1,len(Glob_state_list),num=len(Glob_state_list))
for i in range(len(Glob_state_list)):
    Glob_state=np.array(pl.loadtxt(Glob_state_list[i],skiprows=1)) # import the file at time i
    for j in range(len(Glob_state)):
        R[i,:]= Glob_state # storing the reactants

# List of vertices
V = np.array(pl.loadtxt(’hex.vertices.10242’))
# List of faces: face 0 (first row) has the elements in that row as vertices
F = np.array(pl.loadtxt(’hex.faces.10242’))
# List of centers
C=np.array(pl.loadtxt(’hex.centers.10242’))

F1=np.copy(F.astype(’int32’)-1)
# Adjust the pentagons
F1[:12,5]=F1[:12,4]

# Function to convert hexagons in triangles
def hexagons_to_triangles(vertices, faces):
    triangles = []
    for face in faces:

        for i in range(1, len(face) - 1):
            triangles.append([face[0], face[i], face[i + 1]])
    return np.array(triangles)

# Convert the mesh composed by hexagons in a triangular mesh
triangles = hexagons_to_triangles(V, F1)

from glob import glob
Det_state_list=glob(path+’*00000_D*.txt’)
Det_state_list.sort()
Det_state_list=Det_state_list[::10] # slicing, I keep an element every 10 of the list
# Order the list based on the extracted time
Det_state_list.sort(key=estrai_tempo)

# VISUALIZATION METHOD n.1

# Order parameter: Rab5:GTP - Rab5:GDP

# Colours composing the colorbar: dark blue → green/light blue → yellow
colors = [(0, 0, 0.4),  # dark blue
          (0.4, 1.0, 0.4),    # green/light blue
          (1, 1, 0)]    # yellow
custom_cmap = LinearSegmentedColormap.from_list("my_cmap", colors)

last_mes = len(Det_state_list)
reactant1 = 1
reactant2 = 2

for i in range(last_mes):
    All = np.array(np.loadtxt(Det_state_list[i],skiprows=1)) # import the file at time i
    diff = All[:,reactant1-1] - All[:,reactant2-1] # storing the reactant at time i
    max_diff = np.max(np.abs(diff))
    if max_diff == 0:
        normalized_diff = diff
    else:
        normalized_diff = diff / max_diff

    # setting colors for Conf at time i
    color = np.zeros(4*diff.shape[0])
    color[::4] = normalized_diff
    color[1::4] = normalized_diff
    color[2::4] = normalized_diff
    color[3::4] = normalized_diff
    print("%s" % Det_state_list[i])
    canvas = plot(V,triangles,color)

    # Creating customized colorbar for each frame
    fig, ax = plt.subplots(figsize=(3, 1))
    vmin = normalized_diff.min()
    vmax = normalized_diff.max()
    norm = plt.Normalize(vmin=vmin, vmax=vmax)
    sm = plt.cm.ScalarMappable(cmap=custom_cmap, norm=norm)
    sm.set_array([])
    cbar = plt.colorbar(sm, cax=ax, orientation='horizontal')
    cbar.set_label('Rab5:GTP␣-␣Rab5:GDP', fontsize=10)
    plt.tight_layout()
    plt.show()

# VISUALIZATION METHOD n.2

# Total Rab: Rab5:GTP + Rab5:GDP

# Colours composing the colorbar: dark blue → green/light blue → yellow
colors = [(0, 0, 0.4),  # dark blue
          (0.4, 1.0, 0.4),    # green/light blue
          (1, 1, 0)]    # yellow
custom_cmap = LinearSegmentedColormap.from_list("my_cmap", colors)

last_mes = len(Det_state_list)
reactant1 = 1
reactant2 = 2

for i in range(last_mes):
    All = np.array(np.loadtxt(Det_state_list[i],skiprows=1)) # import the file at time i
    Rab = All[:,reactant1-1] + All[:,reactant2-1] # storing the reactant at time i
    max_Rab = np.max(np.abs(Rab))
    if max_Rab == 0:
        normalized_Rab = Rab
    else:
        normalized_Rab = Rab / max_Rab

    # setting colors for Conf at time i
    color = np.zeros(4*Rab.shape[0])
    color[::4] = normalized_Rab
    color[1::4] = normalized_Rab
    color[2::4] = normalized_Rab
    color[3::4] = normalized_Rab

    print("%s" % Det_state_list[i])
    canvas = plot(V,triangles,color)

    # Creating customized colorbar for each frame
    fig, ax = plt.subplots(figsize=(3, 1))
    vmin = normalized_Rab.min()
    vmax = normalized_Rab.max()
    norm = plt.Normalize(vmin=vmin, vmax=vmax)
    sm = plt.cm.ScalarMappable(cmap=custom_cmap, norm=norm)
    sm.set_array([])
    cbar = plt.colorbar(sm, cax=ax, orientation='horizontal')
    cbar.set_label('Rab5:GTP␣+␣Rab5:GDP', fontsize=10)
    plt.tight_layout()
    plt.show()
```