Modelling at the Mesoscale: a Novel Approach to Protein-Protein Interaction and Multicomplex Formation

Benedetta Frida Baldi
Homerton College
University of Cambridge

A thesis submitted for the degree of

Doctor of Philosophy

May 28, 2014

Declaration

MODELLING AT THE MESOSCALE: A NOVEL APPROACH TO PROTEIN-PROTEIN INTERACTION AND MULTICOMPLEX FORMATION

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text.

This thesis does not exceed the specified length limit of 60.000 words as defined by the Biology Degree Committee.

This thesis has been typeset in 12pt font using Latex according to the specifications defined by the Board of Graduate Studies and the Biology Degree Committee.

Acknowledgments

I would like to thank my supervisor Nicolas Le Novère, who gave me the extraordinary opportunity to be part of the EBI and to give me the freedom to pursue the topic in this thesis, and my TAC members Julio Saez-Rodriguez, Janet Thornton and Ole Paulsen for the yearly discussions on my results. I would like to thank the welcoming CompNeur especially Lu Lee, Michele Mattioni, Massimo Lai, and Christine Seeliger, for the laughs, the compelling scientific conversations and the flying animals.

When I first started this PhD, I never imagined I would find such an amazing community of fellow students. Jean-Baptiste Pettit, Felix Kruger, Nils Kolling, Michele, Jorge Soares, Sander Timmer, Myrto Kostadima, Nenad Bartonicek, and Christine, you guys really made this experience worth the pain. If there were an award for the nicest person it would go to Nenad Bartonicek. He made me feel like I belonged from the very first day and I can't thank him enough for all the help and for making Croatia happen.

During this PhD I had countless ups and downs, and some of the most intense moments of my life. From day one, this crazily amusing lab mate of mine shared them all. We started this adventure together at the same time and we grew closer at every turn. It has been a hell of a ride Christine. Thank you for Zagreb, thank you for pushing the boundaries of my comfort zone one nanometer at the time, and thank you for always being there for me.

I also would like to thank my friends from Italy Serena Corradini, Sara Montanari, Barbara Testoni, and Marialuisa Caiazzo, that somehow managed to be so close to me despite the geographical distance. You showed me that friendship doesn't know any barrier, thank you. I would like to thank the patient proofreaders of this thesis: Benjamin Stauch, Andrew Ayres, Steven Wilder, Massimo, Nenad and of course Christine. I can't thank you enough for the time and effort, but I am sure a beer will be good start.

Lastly, I would like to thank my parents, who never stop cheering for me. Grazie mamma, grazie papà, you're the best.

Abstract

Modelling at the Mesoscale: a Novel Approach to Protein-Protein Interaction and Multicomplex Formation.

Baldi Bendetta Frida

The Post-Synaptic Density (PSD) is a proteinaceous organelle present on the membrane of excitatory spines that is at the core of signal processing in neuronal transmission. The major components of the PSD have been identified as well as its layered organisation, but so far there were no successful functional reconstruction.

A new methodology has been developed in order to study the PSD assembly formation, in which protein-protein interactions are studied with a level of detail that is between an all-atom representation of a protein and a primitive geometry representation. The proposed method takes advantage of the computer animation software Maya, to create a coarse grained representation of the protein surfaces starting from their PDB file structure. Subsequently Unity, a game engine, is used to simulate diffusion and reaction of the protein models in the simulation environment, using the developed Unity extension T.A.R.S.I.D. T.A.R.S.I.D utilises an agent-based, event driven approach to encode the protein behaviour inside a simulator container of virtually any geometry. Diffusion in 2/3D was implemented as mass driven, with explicit calculations of translation and rotation, and the 2D diffusion was implemented to be independent of the shape of the container, its concave nature, and its tessellation. The protein binding strategy was implemented based on collision theory. The method was fully validated for the 2D and 3D diffusion and for complex formation, and applied to the study of the PSD assembly formation.

The model reproduced experimental evidence and showed that a PSD-like structure is an emerging property of the protein geometries and their binding network. Moreover, the model helped gaining new insights into the structural importance of key components of the PSD such as Homer, Shank3 and PSD-95.

Contents

List of Figures

List of Tables

Nomenclature

Roman Symbols

AFP	Insect antifreeze protein
$AMPAR$	alpha-Amino-3-hydroxy-5-Methyl-4-isoxazolePropionic Acid Receptor
BD	Brownian Dynamics
BTM	Bound To Membrane, protein state
CA	Cellular Automata
$CaMKII$	calcium /calmodulin-dependent protein kinase II
D_r	Rotational Diffusion coefficient
D_t	Translational Diffusion coeffient
$ePMV$	embedded Phython Molecular Viewer
$ERK2$	Extracellular signal-Regulated Kinases 2
$FabH$	Beta-Ketoacyl-acyl carrier protein synthase III
FBX	FilmBoX, file format
FVM	Finite Volume Method
$GAPDH$	Glyceraldehyde 3-Phosphate Dehydrogenase
LTD	Long Term Depression

LTP Long Term Potentiation

MD Molecular Dynamics

$mMaya$ molecular Maya toolkit

MSD Mean Squared Displacement

$NMDAR$ N-Methyl-D-Aspartate Receptor

NMR Nuclear Magnetic Resonance

ODE Ordinary Differential Equation

PDB Protein Data Bank

PDE Partial Differential Equation

PDZ acronym combining the first letters of three proteins post synaptic density protein (PSD95), Drosophila disc large tumor suppressor (Dlg1), and zonula occludens-1 protein (zo-1)

$PI3K$ Phospho-Inositide 3-Kinase

$PI3K\gamma$ PhosphatidylInositol-4,5-bisphosphate 3-Kinase gamma

PKA cyclic adenosine monophosphate-dependent protein kinase

RBD Ras Binding Domain

$RMSD$ Root Mean Squared Displacement

SAM Sterile alpha motif, protein domain

$SHANK3$ SH3 and multiple ankyrin repeat domains 3

SSA Stochastic Simulation Algorithm

Chapter 1

Introduction

> For some people, small, beautiful events are what life is all about.
>
> *The Doctor*
> Doctor Who

From before we are born our brain has quite remarkable abilities; it reacts to the physical environment and it is continually changed by our experiences. The brain not only responds to the changes in our environment, but it also records them while creating memories. However, memories are more than a mere collections of events. Memories are used to solve problems. They are mandatory for a functional language. As the Nobel prize laureate in physiology and medicine in 2000, Eric Kandel, said " Memory is the glue that binds our life together, it allows you to have continuity in your life". In order to understand how memory works it is essential to understand how the brain functions. The brain is probably the last big frontier in mammalian physiology, one of the last big mysteries to solve.

In the XIX century, it was still debated whether the cell theory applied to the nervous system or if every cell in our brain was connected like a fish net. The first descriptions of nerve cells are attributed to Christian Gottfried Ehrenberg (Ehrenberg, 1836) who, in 1836, studied the nervous system of a leech, and to Purkinje (Purkinje, 1837) who, in 1837, described large cells in a mammalian cerebellum, that nowadays are known by his name (Figure 1.1 C). One of his students, Valentin, was supposedly the first to publish a drawing with details of the nucleus and nucleolus of a nerve cell (Figure 1.1 A). The first complete image of a neuron with details of axons and dendrites is attributed to Deiters, in posthumous publication from 1865 (Deiters and Schultze, 1865) (Figure 1.1 B).

Another milestone for neuroscience is the work of Camillo Golgi with his histological method of staining that revolutionised the study of the nervous system, called the silver-chromate technique (Golgi, 1891). Firstly, the brain tissue was fixed with a mixture of potassium bichromate and osmic acid for several days, then stained through immersion in a silver nitrate solution. The stain created the now famous *black reaction* in which a black precipitate of silver chromate is produced, making nerve cells detectable under a microscope. Between the years 1883 and 1886, Golgi published a series of works in which he proposed that nerve cells are fused by the terminal arborization much like the intricate net of blood vessels and capillaries, a theory called *rete nervosa diffusa* or

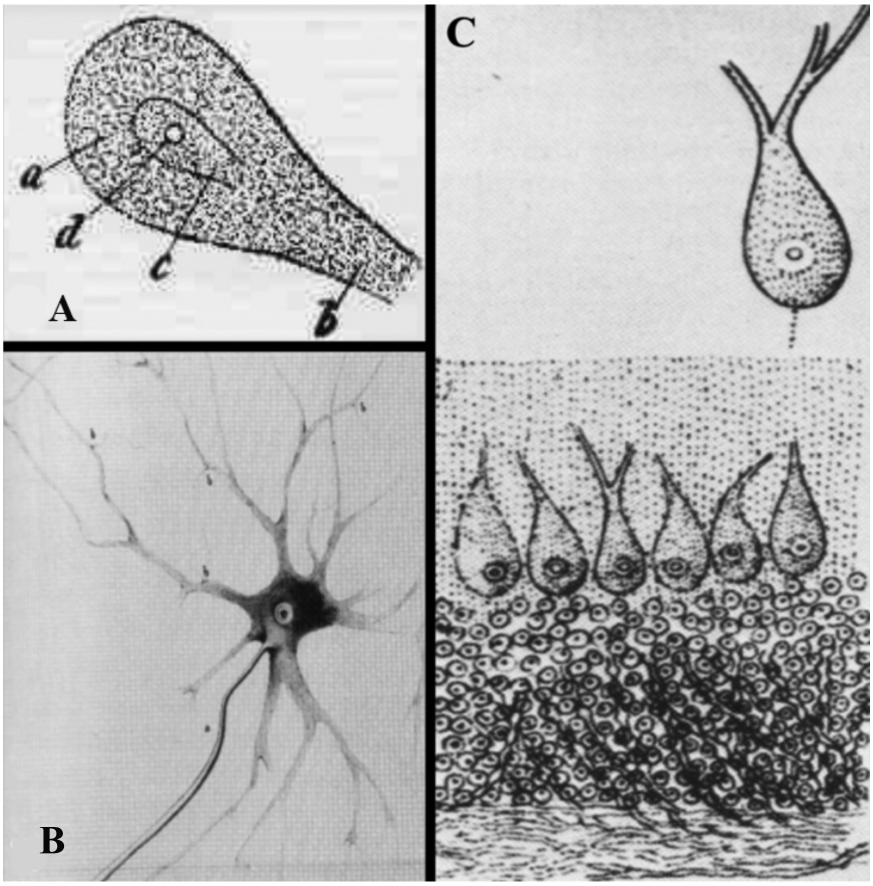


Figure 1.1: First illustrations of nerve cells: (A) First drawing of a nerve cell from the human cerebellum with details on nucleus and nucleolus by Valentin in 1836. (B) Detailed drawing of a spinal cord nerve cell by Deiteres from 1865 showing the nucleus and the cell body as well as axons and dendrites (Deiters and Schultze, 1865). (C) Drawing of neuronal cells from the cerebellum by Purkinje for the Congress of Physician and Scientists Conference in Prague 1837 (Purkinje, 1837).

reticular theory (Golgi, 1883; Golgi, 1885).

In 1888, just few years after these publications, Santiago Ramon y Cajal reported a complete opposite observation. Using the double impregnation technique that resulted from the improvement of Golgi's staining, he observed discontinuity between nerve cells, reported as empty space between the digitiform arborization of the dendrites and axonic fibres (Cajal, 1889). He also majorly contributed to the validation of the neuron theory, stating that the nerve cells are independent elements, never anastomoses (unlike in the reticular theory) and that the nervous propagation is allowed by contacts at the level of specific apparatuses (Cajal, 1889).

A convinced defender of the neuron theory was Wilhelm von Waldeyer that coined the term neuron to define a nerve cell. In his work from 1891, Waldeyer stated: " The nervous system is constituted by numerous nervous units (neurons) without anatomical or genetic connection. Each nervous unit comprise of three parts: the nerve cell, the nerve fibre and the terminal arborization." (Waldeyer-Hartz, 1891).

The subsequent year, Cajal made another crucial discovery in the history of neuroscience, known as *the law of dynamic polarisation*, in which he proposed that the information was carried between neurons like in an electrical circuit from the dendrites to the soma and from there to the axons (Cajal, 1892).

Due to the very limited power of the optic microscope, it was impossible to actually see the contacts between neurons but in 1897 the concept of synapses emerged in a chapter of the *Textbook of Physiology* written by Charles Scott Sherrington (Foster, 1897):

> "So far as our present knowledge goes, we are led to think that the tip of a twig of the arborescence is not continuous with but merely in contact with the substance of the dendrite or cell-body on which it impinges. Such a special connection of one nerve cell with another might be called a synapse. The lack of continuity between the material of the arborization of one cell and that of the dendrite (or body) of the other offers the opportunity for some change in the nature of the nervous impulse as it passes from one cell to the other."

The definitive confirmation of the concept of synapses and therefore of the neuron theory came in the mid–1950s with the advent of electron microscopy. In 1954, a group formed by Sanford Louis Palay and George Emil Palade, and in 1955 by Eduardo De Robertis and Henry Stanley Bennet published electron microscopy images that demonstrated the individuality of neurons and the synaptic discontinuity (Palade, 1954; De Robertis and Bennet, 1955).

They described a local swelling of the neuronal membrane at the synaptic level, an accumulation of small vesicles close to the broadening of the presynaptic element of 20–60 nm in diameter, and 20 nm of extracellular space between the two swollen membrane (figure 1.2).

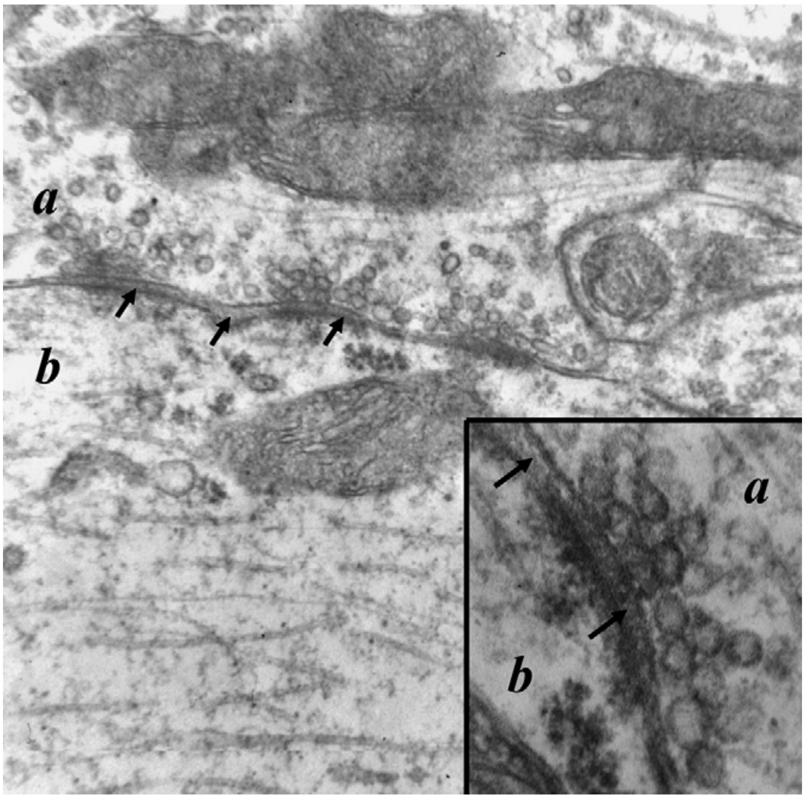


Figure 1.2: First electron micrograph of an axo-somatic synapse: the image shows a synapse from the anterior horn of the human spinal cord. Labeled as a) is the axon which constitutes the presynaptic element with the characteristic vesicles near the presynaptic membrane and some in contact with it. Labeled as b) is the postsynaptic element. The arrows point to a clear synaptic space between the two elements.

1.1 Hippocampus: a special place for our memories

Although techniques including imaging and electrophysiology highly improved since 1955, the specific functions of dendritic spines are still elusive, especially regarding their plasticity. It has been proposed that dendritic spine changes are associated with learning and memory storage, highlighting the importance of fully understanding their functions.

The first theory on how neurons adapt during learning was formulated by Donald Olding Hebb in 1949 (Hebb, 1949). He suggested that "When an axon of cell A is near enough to excite cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased"

The Hebb theory proved to be quite difficult to verify with experimental evidence, mostly due to the lack of good experimental models. Studies with patients with damage to the medial temporal lobes helped to categorise memory into two distinct systems: procedural and declarative (Scoville and Milner, 1957; Squire, 1992; Schacter and Tulving, 1994). Procedural (or implicit) memory is the memory associated with perceptual and motor skills, while declarative (or explicit) memory is the memory for facts, events, places and people which is particularly linked to the hippocampus and the medial temporal lobe.

Procedural memory is a type of memory that is shared even with invertebrates and insects, which are intrinsically simpler systems. Basic reflexes like the gill-withdrawal of Aplysia (Kandel and Tauc, 1963) or the escape reflex of Tritonia (Willows and Hoyle, 1969) fall into the category of procedural memory.

Even though these findings helped to link specific types of memory with anatomical regions of the brain, they did not provide any inside at a cellular level. The next step for understanding the biology of memory was made in 1971 when O'Keefe and Dostrovsky published their findings on single unit recordings firing in the hippocampus of awake and freely moving rats (O'Keefe and Dostrovsky, 1971).

The study revealed that some hippocampal neurons register information about the animal environment based on sensory inputs. These cells were called *place cells* and they selectively fired when the animal entered a specific area of its surrounding. It was later postulated that these hippocampal cells make a cognitive map of the environment that animals use for navigation purposes (O'Keefe and Nadel, 1978).

Since then, the study of memory has been closely linked to the investigation of mammalian hippocampus in animals performing different navigation or place recognition tasks. One example of these experiments is the Morris water maze, in which animals need to find a platform hidden below the water surface using only visual clues (Morris et al., 1986). Interestingly, a recent work using structural MRI technique confirmed the importance of the hippocampus for place memory in humans (Maguire et al., 2000). In this study MRI scans from 16 licensed London taxi drivers, who underwent extensive training (roughly 2 years) and examinations for their licence, were compared to a control group. The scan of the subjects showed a significant increase in size of the posterior hippocampus compared to control subjects. This study confirmed not only that the hippocampal area stores a spatial representation of the environment, but also suggested that the adult brain possesses the necessary plasticity to accommodate a large amount of new information.

1.2 Anatomy of a dendritic spine

It is estimated that the human brain contains about 86 billions of neurons (Herculano-Houzel, 2012). On the most principal neurons[1], little protrusions known as dendritic spines populate the dendritic arborization. As previously mentioned, dendritic spines are the receiving end of individual synapses, and they play a key role in the processing of the synaptic information. In the hippocampus, dendritic spines are predominantly excitatory (Megias et al., 2001) and mostly contacted by a single presynaptic bouton, a specialised structure at the end of an axonic terminal (Andersen, 1990). The estimated density of dendritic spines varies from 2 to 4 per μm of dendrites from hippocampal granule cells (Trommald and Hulleberg, 1997) and CA1 pyramidal cells (Harris and Stevens, 1989; Harris et al., 1992; Goldman-Rakic et al., 1989). The spines are localised throughout the dendrites but their specific spatial distribution seems to play a role in the transmission efficacy of the information to the cell body, the soma (Whitford et al., 2002).

The information is passed to a dendritic spine in the form of a neurotransmitter release. When an action potential reaches the presynaptic bouton it triggers the fusion of vesicles containing neurotransmitters with the presynaptic membrane (figure 1.3).

[1] including the pyramidal neurons of the neocortex, the medium spiny neurons of the striatum, and the Purkinje cells of the cerebellum.

The neurotransmitters are then released into the synaptic cleft. The diffusing molecules of neurotransmitter reach the ligand gated ion channels expressed in the postsynaptic membrane and activate them. Subsequently, an ion influx through the channels triggers two major events: activation of specific signalling pathways and depolarisation of the postsynaptic membrane. After the neurotransmitters dissociate from the channels various re-uptake mechanisms recycle the neurotransmitters back into the presynaptic site.

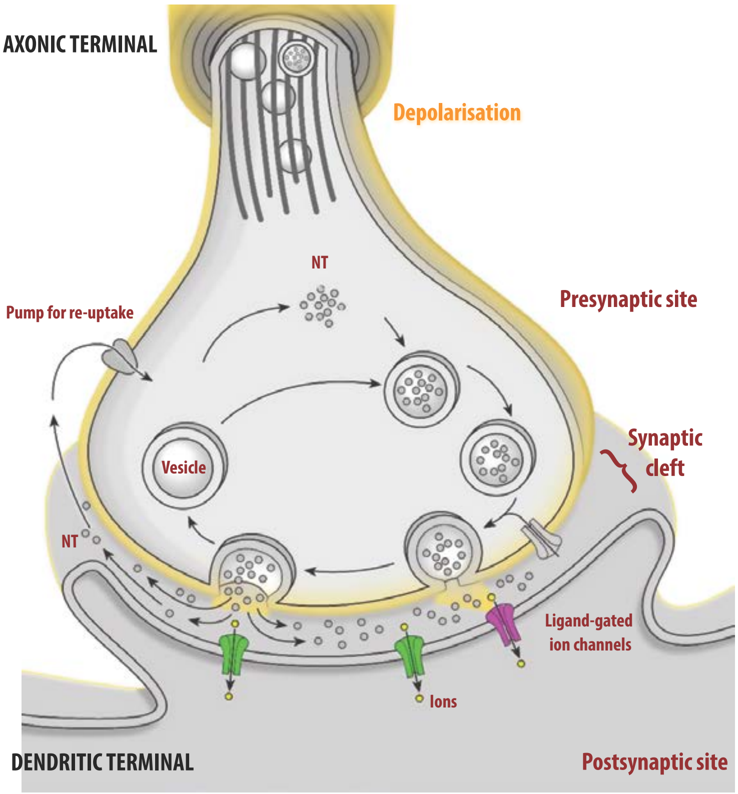


Figure 1.3: Schematic representation of the key steps of synaptic transmission: upon depolarisation caused by an action potential running across the axonic terminal of a neuron, the vesicles in which the neurotransmitters are stored fuse with the presynaptic membrane. The neurotransmitters (NTs) are then released into the synaptic cleft in which they get in contact with the extracellular domain of the ligand-gated ion channels present on the postsynaptic site. Upon binding of the NTs, the channels are activated and ions enter the postsynaptic site, triggering signalling pathways and the depolarisation of the membrane. The NTs are then recycled into the presynaptic site.

Dendritic spines have different shapes and sizes (figure 1.4). The most common mature spines are the mushroom spines (figure 1.4 b), in which a thin neck is surmounted by a head of about 0.6 microns in diameter (Harris et al., 1992). Other spines show a stubby protrusion from the neck and multiple synapses. The larger the spine, the bigger the protein apparatus for signal transmission known as the Post-Synaptic Density (PSD) will be. Having bigger PSDs increase not only the amount of proteins present in it therefore increasing the strength of processed signals, but also their interactions resulting in a more complex assembly. For excitatory glutamatergic synapses commonly found in the hippocampus, larger spines contain a higher number of glutamate receptors, the alpha-amino–3-hydroxy–5-methyl–4-isoxazolepropionic acid receptor AMPAR and the N-methyl-D-aspartate receptor NMDAR. More receptors on the membrane translate into a stronger response to glutamate stimuli, resulting in various effects from a higher amount of intracellular calcium, to a different regulation of protein translation and degradation. The accepted theory behind this morphological variability is that smaller spines, as in case of stubby spines, are more prone to changes and are therefore implicated in the process of learning. On the other hand, bigger spines such as the mushroom type, are in a more advanced state of maturation and are implicated in the process of memory (Bourne and Harris, 2007).

The morphological changes of dendritic spines are referred to as structural plasticity which is branch of the more generic term synaptic plasticity. The first documented observation of synaptic plasticity was made in 1966 and in 1973 by Lomo and Bliss (Andersen et al., 1966; Bliss and Lomo, 1973). Upon high frequency stimulation of the perforant path input of the hippocampus[2], the stimulated synapses displayed an increase in strength. This phenomenon is known as *long term potentiation* or LTP. It was more than ten years later, thanks to the work of Wingstrom and collegues, that LTP was associated to Hebb's theory (see section 1.1) due to the postulated properties of associability and specificity. In particular they demonstrated that only synapses that are in an active state when the postsynaptic cell is strongly depolarised were potentiated, while neighbouring inactive synapses were not (Wigstrom et al., 1986). Hebbian LTP also showed cooperativity since multiple inputs must be activated in order to produce a sufficient depolarisation in the postsynaptic site to induce LTP. This implies that

[2] The perforant path input of the hippocampus is a connectional route from the entorhinal cortex to all fields of the hippocampal formation, dentate gyrus, CA and subiculum.

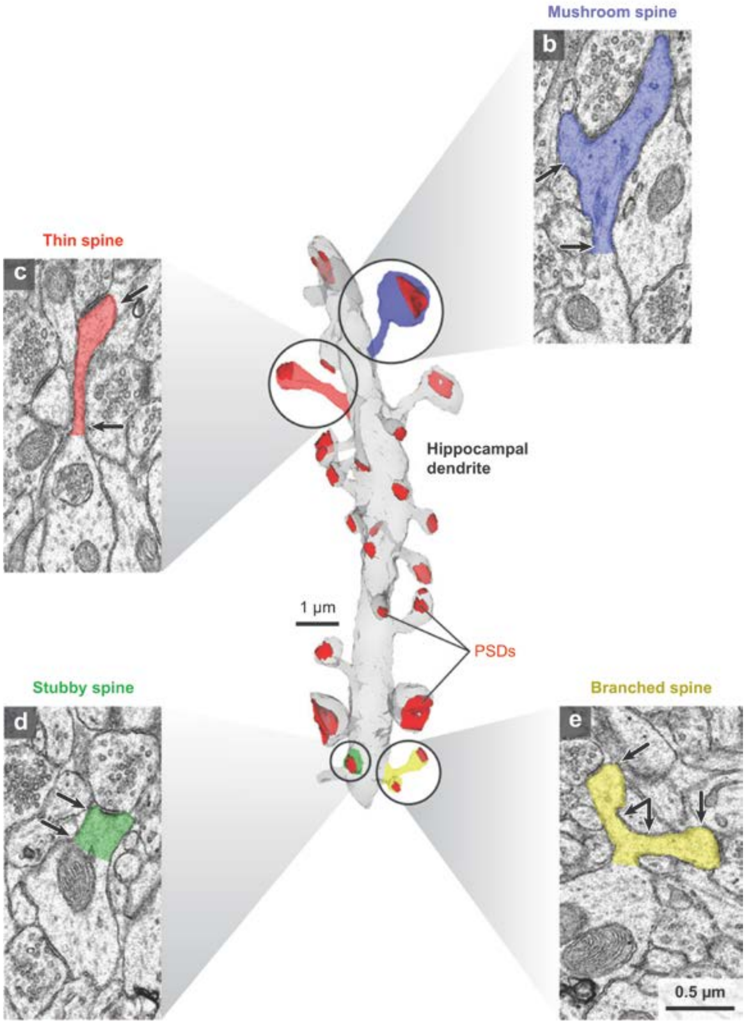


Figure 1.4: Variability in shape and size of dendritic spines: a 3D reconstruction of a hippocampal dendrite (light grey) showing spines of different shapes and sizes including mushroom (blue), thin (red), stubby (green) and branched (yellow). The PSDs are depicted in red and also show different shapes and sizes. To illustrate the relationship between the reconstructed 3D image and the EM image stacks, for each highlighted spine, an EM image was chosen and colour coded accordingly. (B) Example of a mushroom shaped spine (blue), in which the head diameter exceeds of 0.6 microns the neck. (C) Example of a thin spine (red), with a small head and narrow neck. (D) Example of a stubby spine (green) with a similar head and neck diameter. (E) Example of a branched spine (yellow) in which a single spine divided in two branches each of which developed into a thin spine. Scale bar set to 0.5 μm and the arrows pinpoint where the head and neck diameters were measured for each spine. Image adapted from (Bourne and Harris, 2008a)

coordinated group of synapses contribute together to the firing of the postsynaptic neuron, increasing its strength.

The explanation of most of these features can be linked to the behaviour of NMDA receptors that are present on the postsynaptic site. Unlike most neurotransmitter receptors that activate solely by binding their associated neurotransmitter, the NMDAR is also sensitive to membrane potential. At resting conditions, even when glutamate is bound to the receptor, a magnesium ion blocks the channel pore, preventing the influx of calcium. However, when the postsynaptic membrane is depolarised, the magnesium block is released and calcium can flow through the channel (Banke and Traynelis, 2003). Since NDMAR activation requires both, glutamate and depolarisation of the postsynaptic membrane, LTP can only occur in synapses where the presynaptic neuron was already active (associativity) and on a postsynaptic neuron that was already depolarised at or near the time of the transmitter release (cooperativity).

Upon NMDAR receptor activation, the calcium signal activates a wide range of signalling pathways including the calcium buffering protein calmodulin, the calcium /calmodulin-dependent protein kinase II (CaMKII), cyclic adenosine monophosphate-dependent protein kinase (PKA), and the MAP kinase cascade (Kennedy et al., 2005). Each of these proteins is implicated in the induction of LTP. The activation of Calmodulin by calcium is required for the activation of CaMKII. CaMKII, once active, promotes the phosphorylation of AMPAR, which increases the channel conductance, and also promotes insertion of new AMPAR into the membrane. Both processes lead to an increase of the strength of the synapse. The number of AMPA receptors present in the postsynaptic membrane correlates with size and the maturity of the synapse, determining the relative strength of the synaptic response to glutamate stimulation (Bourne and Harris, 2008b). On the opposite spectrum of synaptic plasticity, there is the long term depression, LTD which also require NMDAR activation and calcium pathways (Mulkey and Malenka, 1992). This bidirectional control of synaptic strength supports the idea that memories are encoded in the distribution of synaptic weights in neural circuits, and not simply by LTP mechanisms.

Many other types of synaptic plasticity have been discovered in recent years, some of which are NMDAR independent, like the metabotropic glutamate receptor dependent LTD, and the endocannabinoid mediated LTD (Citri and Malenka, 2008). All these forms of plasticity, apart from presynaptic LTP, share the same basic principle: membrane receptors capture the signal from the presynaptic site triggering second messengers. A protein that responds to the second messenger will then activate other proteins in a cascade of interactions. In order for the second messenger and responsive proteins to be activated they need to be colocalised in the same microdomain of their respective activators.

There are three factors that affect the colocalisation of second messengers and their effectors: the amplitude of the signal, the duration of the increase in the messenger concentration, and the relative position of the messenger and its effector. Therefore, in order to better understand how synaptic plasticity functions and how it is linked to memory and learning, it is necessary to determine how second messengers and their effectors are localised in the postsynaptic site with respect to membrane receptors. In order to achieve this, it is necessary to create an accurate structural model of the area located underneath the membrane receptors.

1.3 The Post Synaptic Density: the functional core of a dendritic spine

In excitatory synapses, response and transduction of neurotransmitter signals takes place at the level of the postsynaptic density, PSD, a functional specialisation of the postsynaptic membrane and the adjacent cytoplasmic compartment. The PSD was firstly observed by Gray in 1959 as a fuzzy electron-dense thickening in an electron micrograph. It was only in 1980 that the first PSDs were isolated and measured as disk-like structures of 20–30 nm in diameter (Carlin et al., 1980). The PSD could be described as proteinaceous organelle attached to the postsynaptic membrane and held in place by actin filaments coming from the cytosol of the dendritic spine (Sheng and Kim, 2000). A typical PSD covers the tip of a dendritic spine directly opposite the release site of neurotransmitters. A typical spine has a circular shape but, especially for large spines, the PSD can often be irregular and discontinuos (perforated PSD) (Spacek and Hartmann, 1983). PSDs are heterogeneous in size: their diameter can range from 200 to 800 nm with a mean value of 300–400 nm and a thickness from 30 to 60 nm (Chen et al., 2008b; Bourne and Harris, 2008a; Dosemeci et al., 2000). The PSD contains the glutamate receptors and their associated signalling molecules as well scaffolding proteins. The PSD size correlates with the size of the spine in which it is located and with the abundance of glutamate receptors, another evidence that bigger synapses are also stronger (Kasai et al., 2010).

The starting point in understanding the PSD assembly was to establish its protein composition. Estimations of the PSD mass was made by Chen and colleagues in 2005, using scanning transmission electron microscopy (STEM) images from purified and freeze-dried PSDs. They revealed that the molecular mass of an average PSD is about 1 GDa (Chen et al., 2005). The first proteins isolated from the PSDs were identified by SDS-PAGE of purified PSD as CaMKII and the membrane associated guanylate kinase MAGUK protein PSD–95 (Banker et al., 1974; Kennedy et al., 1983; Cho et al., 1992). A fruitful approach to identify PSD proteins were yeast two-hybrid screens using known postsynaptic proteins as *bait*. Yeast two-hybrid screens with cytoplasmic tails of NMDA receptors and potassium channels revealed that these membrane proteins bind directly to proteins of the PSD–95 family (Kim et al., 1995a; Niethammer et al., 1996). Another technique used to identity PSD components is the *tandem affinity purification* or TAP which involves the creations of a fusion protein with a *tag*. The tag is used to separate the fusion protein, and everything that is bound to it, through affinity selection (Rigaut et al., 1999). When this technique was used on PSD–95 a multiprotein complex was recovered containing 118 proteins including glutamate receptors, potassium channels, scaffolding as well as signalling proteins (Fernández et al., 2009).

In recent years, with the advancements of mass spectrometry and peptide fingerprinting, precise identification of the PSD composition became possible (Walikonis et al., 2000). A recent experiment carried out by the group of Seth Grant with human neocortex tissue, identified an astonishing number of 1461 different proteins in the PSD using peptide fingerprinting techniques (Bayes et al., 2010).

Various experimental approaches and PSD purification protocols showed very different results in the composition of the PSD assembly (Chen et al., 2005; Cheng et al., 2006; Chang et al., 2007). Despite this diversity, a set of more than 400 proteins was found to be common to all the experiments (Collins et al., 2006). It is possible that this list of common proteins contains some false positives due to impurity of the PSD preparations, such as contamination by mitochondrial proteins and material from other organelles (Sheng and Kim, 2011). Conversely, mass spectrometry approaches may have missed true PSD proteins that are present at low concentrations or are loosely associated with other PSD components. Membrane receptors and ion channels, protein kinases and phosphatases involved in signal transmission, as well as scaffolding and anchoring proteins are highly represented in the PSD proteome. Mass spectrometry analysis of proteins associated with NMDAR and PSD–95 led to the identification of two sets of respectively 77 and 288 proteins which are largely overlapping and likely to be at the core of the PSD (Husi et al., 2000; Dosemeci et al., 2007). Quantitative mass spectrometry and imaging methods allowed the measurement of absolute concentration of PSD components (figure 1.5). Kinases and phosphatases are highly localised in the PSD constituting 11% of the purified fraction. Two other major PSD fractions are cytoskeleton and scaffolding proteins that account for 12% and 6% respectively (Peng et al., 2004). For an average PSD, the most expressed proteins in terms of copy numbers are CaMKII (∼5600 subunit resulting in approximately ∼466 holoenzymes), and MAGUK proteins (PSD–95, PSD–93, SAP97 and SAP102 with ∼400 copies, 300 of which are PSD–95 alone) (figure 1.5, panel B). SynGAP, a Ras GTPase-activating protein that binds PSD–95 is also highly represented with ∼360 instances in a medium PSD. Regarding scaffolding proteins other than the MAGUK family, the family of GKAP/SAPAP proteins and Shank/ProSAP are the most prevalent in the PSD with around 150 copies per family (Peng et al., 2004; Sugiyama et al., 2005; Cheng, 2006). It has to be noted that the protein composition of the PSD varies between different brain regions (and probably between cell types) reflecting regional variation in the molecular mechanisms underlying synaptic plasticity (Zhang et al., 1999; Vullhorst et al., 2009).

Using electron microscopy techniques, it was possible to reveal the general three dimensional organisation of the PSD from rats homogenised brain tissue (Petersen et al., 2003). Granular particles of 5–13 nm in diameter and membrane patches of 50–100 nm in diameter were found in the postsynaptic membrane. Furthermore, irregular protrusions formed in large parts by CaMKII were found on the cytoplasmic side of the PSD. Several studies found that proteins were not uniformly distributed in the PSD but particular proteins were positioned in specific locations in the PSD (figure 1.6). PSD–95 molecules were found close to the postsynaptic membrane surface, in agreement with the known interactions between PSD–95 and membrane receptors. Proteins like SAPAP (also known as GKAP), Shank and CaMKII were found located closer to the cytosol in an area about 24–26 nm away from the membrane (Valtschanoff and Weinberg, 2001; Petersen et al., 2003; Rostaing et al., 2006; Dani et al., 2010).

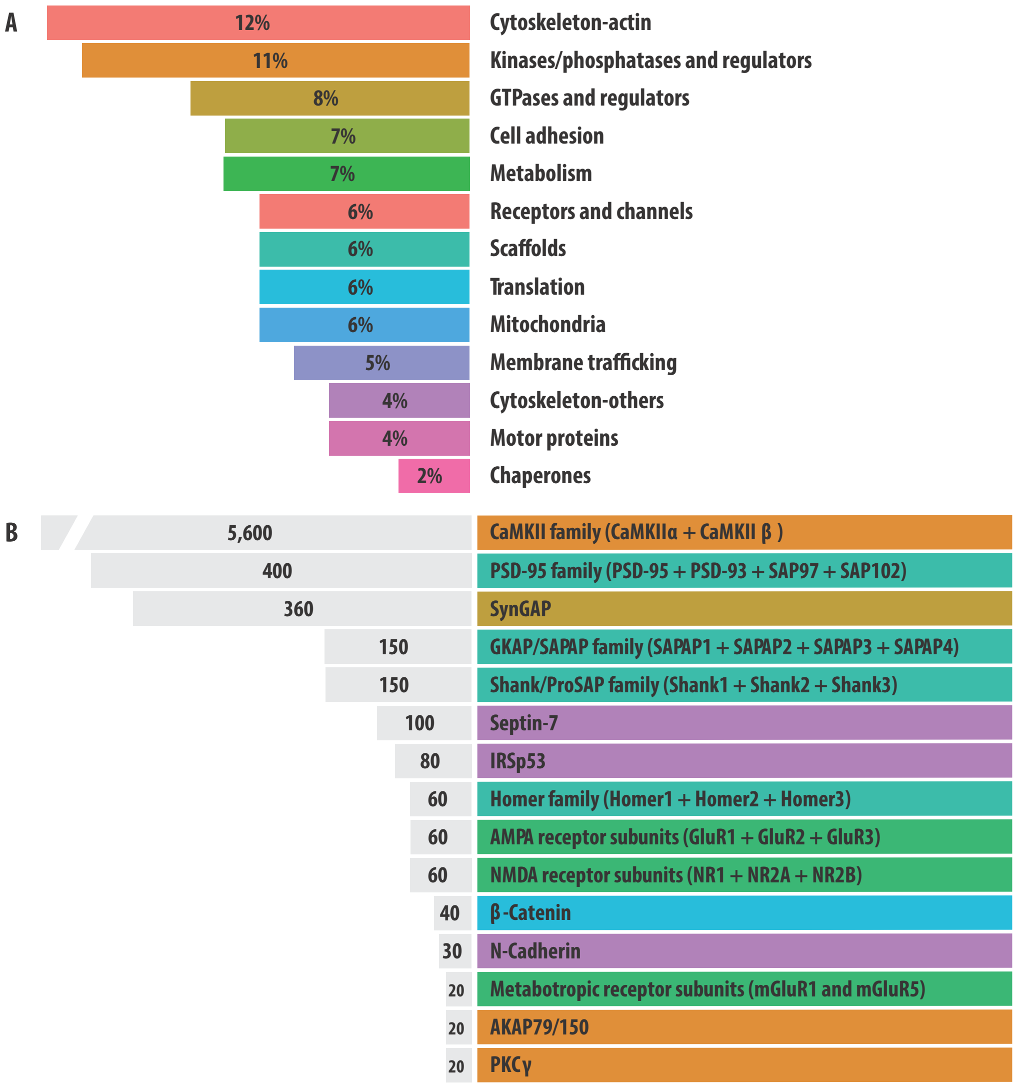


Figure 1.5: Categorisation of proteins of the PSD of excitatory synapses: (A) Relative percentage of proteins present in purified PSDs divided into functional categories. Proteins with miscellaneous function cover the remaining 15%. (B) Copy number of selected proteins in an average PSD, coloured by the protein function as in (A). Adapted from (Collins et al., 2006).

NMDA receptors appear to be located in the central area of the PSD while AMPA receptors are more freely distributed. This is in agreement with the higher stability of NMDAR compared to AMPAR: AMPAR are more mobile compared to NMDAR (higher diffusion coefficients). Moreover AMPAR possess a higher exchange rate reflecting the changes of AMPAR numbers in response to synaptic activity (Kharazia and Weinberg, 1997; Racca et al., 2001; Triller and Choquet, 2008).

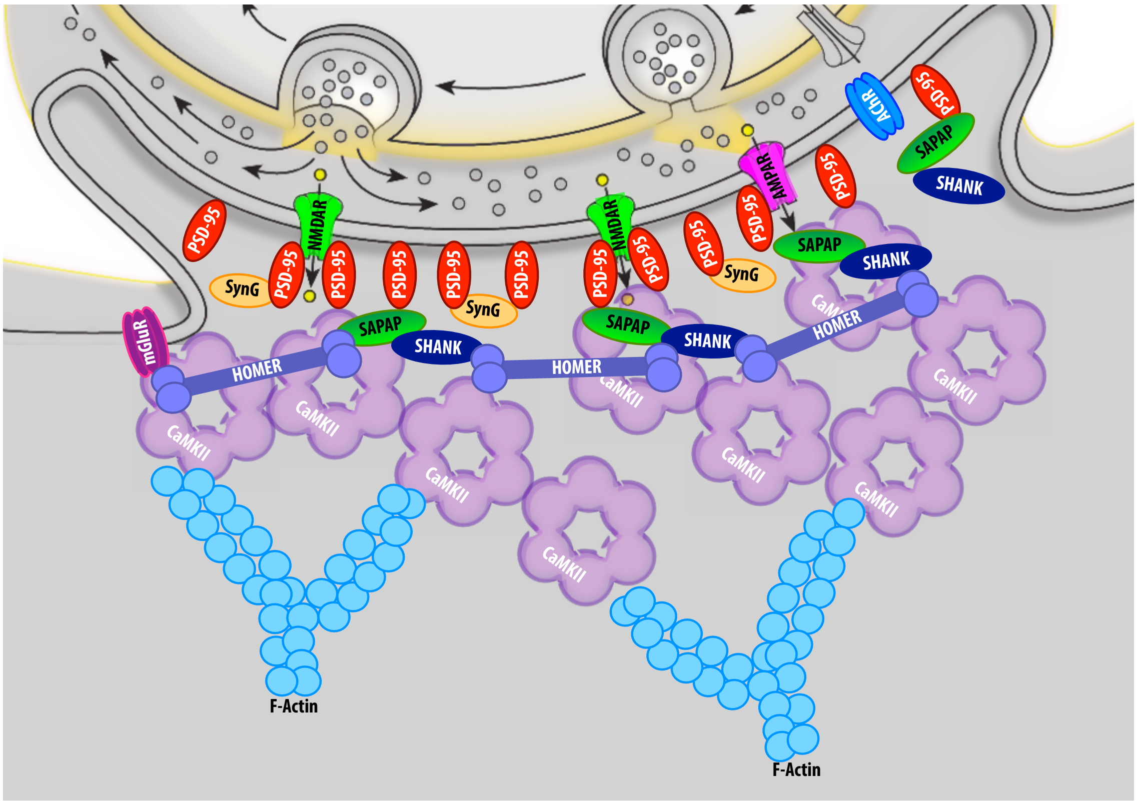


Figure 1.6: Schematic representation of the molecular interactions in the PSD assembly in excitatory synapses: The major molecular components of the PSD are represented and protein interactions are indicated as direct contact between the proteins. mGluR: metabotropic Glutamate receptors; AChR: acetylcholine receptor.

The laminar nature of the PSD has been revealed in 2008 thanks to the work of Chen and colleagues (Chen et al., 2008b; Chen et al., 2008a). In his study, Chen segmented and analysed a part of the protein in PSDs of rat hippocampal neurons. He found that

the juxtamembrane part of the assembly is composed of verticals molecules characterised as PSD–95 bound to the membrane in an extended conformation and to NMDAR and AMPAR cytoplasmic tails. Moreover, two different types of horizontal filaments were found to be organised in two layers. The filaments were characterised by their size and location in respect to the membrane. One pool was found closer to the membrane (10–20 nm) consisting of 20 nm long filaments with 4–5 nm in diameter (depicted in purple in figure 1.7 H and I panel). A second pool of filaments lies further away at a distance of 15–20 nm from the membrane. Those are 30–35 nm long and 5–6 nm in diameter (depicted in white in figure 1.7 H and I panel).
The reconstruction also showed NMDAR and AMPAR cytoplasmic tails and extracellular domains. The tails of both receptors are in contact with the vertical filaments of PSD–95 with a stoichiometry of two PSD–95 per NMDAR tail and one per AMPAR. The horizontal filaments were not definitively associated with any protein of the PSD but it is easy to speculate that the shorter filaments (depicted in purple in figure 1.7) are possibly formed by Shank and or SAPAP proteins and the the longer filaments by Homer, which are all known to interact with each other and with PSD–95 (Kim et al., 1997; Naisbitt et al., 1999).

The complete structure of the PSD is far more complex than presented in the work of Chen, which was focused on the juxtamembrane part of the PSD. Dendritic spines are particularly rich in actin, which is at the very core of the spine's cytoskeleton (Fifková, 1985). A recent study using high resolution electron microscopy and metal shadowing highlights the complexity of the actin network in dendritic spines (Alber et al., 2007) but its functional organisation still remains unclear.

A lot of additional work still has to be done to gain a better understanding of the PSD from a structural and functional perspective. A clear definition of the PSD components and their respective interactions is a necessary step forward as well as a construction of a more dynamic picture of its function. Complex systems like the PSD are unlikely to be fully understood by experimental approaches only. Computational techniques can give good insights into complex systems especially when simulating experiments that are still not feasible to perform in the lab.

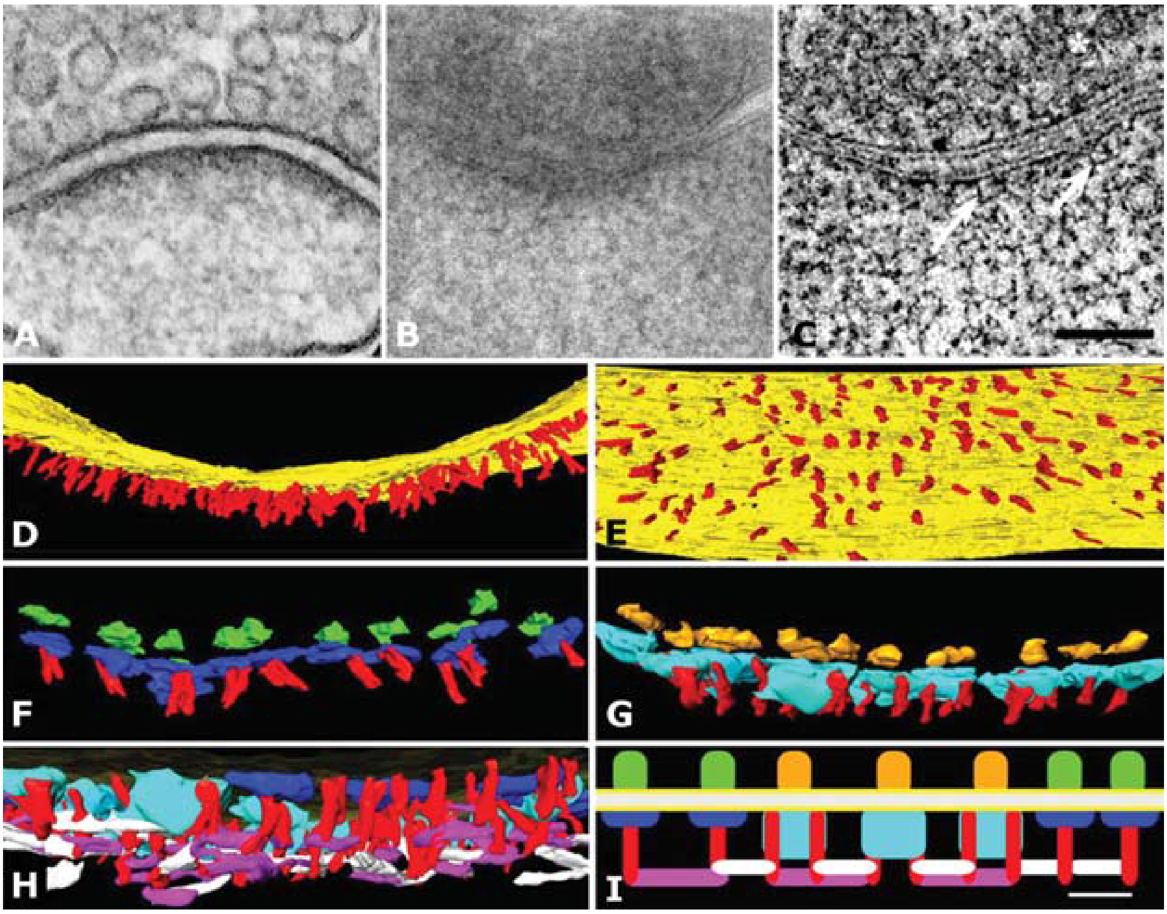


Figure 1.7: Thee dimensional organisation of the PSD derived by EM tomography: (A) EM image of an excitatory synapse of a hippocampal neuron. (B) A typical section of 200 nm in thickness of a hippocampal neuron. (C) A virtual section of 1.4 nm in thickness derived from the reconstructed tomogram of the synapse shown in panel B. (D) Rendered view after the segmentation of the tomogram; The vertical filaments in red are composed of PSD-95, which are in contact with the postsynaptic membrane, represented in yellow. Other components of the PSD were removed for the image for clarity. (E) Cytoplasmic view of the postsynaptic membrane as in panel D. (F) Magnification of some of the membrane bound complexes found at the postsynaptic membrane. The red vertical filaments are PSD-95 while the blue and green are the AMPAR cytoplasmic tail and extracellular domain respectively. The membrane was removed from the picture for clarity. (G) A second type of transmembrane complex, the light blue and yellow ones are composed of NMDAR cytoplasmic tails and extracellular domains respectively. (H) Cross section of the PSD reveal a complex laminar structure. The vertical and transmembrane complexes are in contact with two different horizontal filaments rendered in purple and white which show a slightly different distance from the membrane (10-20 nm for the purple filaments and 15-20 nm for the white) and different length (20 nm purple, 30-35 nm white). (I) Schematic representation of the distribution and interactions of the different structures. Adapted from (Chen et al., 2008b)

1.4 Computational neurobiology

Neurons show a dualistic nature: electrical and biochemical deeply intertwined. Historically, computational neuroscience focused on the electrical behaviour of neurons, which was more accessible to the experimental and computational technology available at the time. The first milestone of computational neuroscience is the mathematical model made by Alan L. Hodgkin and Andrew F. Huxley, that led to the *cable theory* (Hodgkin and Huxley, 1952). The model based on the giant squid axon potential, approximated the electrical behaviour of firing neurons with an electrical circuit, resulting in a surprisingly accurate and quantitative description of the shape of axon potentials. The study of the model made two important contributions: first it revealed that the current generated by sodium and potassium ions is sufficient to generate an action potential; second, by the subsequent confirmation with experimental single-channel recording, it showed that the channel gating mechanisms predicted by the model were indeed correct.

Another major advancement based on the work of Hodgkin and Huxley was the application of their *cable equation* to dendrites (Rall, 1959). Rall was the first to introduce the spatial dimension to neuronal activity and showed that the neuronal membrane has properties of low-pass filters (a filter that reduces the amplitude of high frequency signals while leaving low frequencies untouched) and that the dendritic arborization strongly affects the processing of the synaptic input (Rall, 1962). Such models of neuronal electrical activity are very common in computational neuroscience, and even more surprisingly, the cable equation is still widely used in its original form (Cannon and D'Alessandro, 2006; Rubin and Wechselberger, 2007).

Signalling pathways are not included in electrical models, but they are crucial for the understanding of the functional aspects of synaptic plasticity. Models of synaptic plasticity can be divided into two main categories: phenomenological models and mechanistic models. Phenomenological models accurately describe the relationship between in vitro neural activity and changes in the synaptic responses (Shouval et al., 2002; Song et al., 2000; Morrison et al., 2008). These models are valuable for the investigation of changes in the network involved in the plasticity process but lack a description of the underlying mechanisms. Mechanistic models on the other hand, tend to focus on the role of calcium in synaptic plasticity by combination of the more traditional models of electrical activity with equations that describe the calcium dynamics (Holmes and Levy, 1990; Schiegg et al., 1995; Gamble and Koch, 1987). These models were crucial for the discovery that the amplitude of calcium influx increases depending on the frequency of the stimulation due to the voltage dependent activation of NMDA receptors. However, despite the greater completeness of the mechanistic models, they still do not include all the signalling pathways that influence synaptic plasticity.

The lack of an exhaustive model of synaptic plasticity reflects the complexity of the topic and underlines the need for a systemic approach. Systems level modelling is exceedingly difficult, mainly due to the amount of information required on the quantity and subcellular location of the main protein components. Nonetheless, thanks to high-throughput technology and new computational methods, integrative approaches are more common and more successful.

1.4.1 A connected picture: modelling networks with graph theory

With the advent of the genome era as well as proteomics, metabolomics and lipidomics, more and more large scale datasets are becoming available. One approach to predict and analyse relationships between genes, proteins or molecules is to use graph theory, the study of the interconnectedness of related items. Graph theory was initiated by the work of Leonhard Euler in 1736 (Euler, 1736) and subsequently by the developments of Paul Erdös and Alfred Rényi on random graphs (Erdős and Rényi, 1959).

Instead of viewing reaction pathways as enzymes that require certain substrate to form products, biochemical interactions can be abstracted as nodes and links forming a graph (Eisenberg et al., 2000). Each node in the network represents a molecular species and the link between species, an edge, represents the interaction between them.
The rationale behind this approach is, that the topology of a network alone can lead to new insights on the function and malfunction of a system. Moreover, this approach allows for predictions without any knowledge of the dynamics of the system itself.

A successful example of the graph theory approach was the study of the core metabolic networks of 43 different organisms performed by Jeong and colleagues (Jeong et al., 2000). The study showed that, in metabolic pathways, species do not connect uniformly but that the connectivity follows a Poisson distribution. Moreover, they also show that metabolic networks are scale-free, meaning that they are extremely heterogeneous with their topology being dominated by a few highly connected nodes. These properties imply that only few proteins are highly connected and conserved during evolution, and are therefore more likely to be essential to the functionality of the network. A study of the phosphoprotein network in the synapse, which consisted of more than 1500 proteins, suggested that the network could be a starting point for linking molecules to behaviours (Collins et al., 2005).

The analysis of network topology showed that several structural modules, or *network motifs*, are commonly found in biological systems. Feedback loops and bow-tie motifs, like the one depicted in figure 1.8 panel A, are common examples. The study of these motifs may let us gain insight on the function of larger networks that are more difficult to analyse (Wolf and Arkin, 2003). One limitation of such studies is that the topology alone cannot capture important dynamics such as the time-dependent variation in membrane potential or in protein phosphorylation, which are dependent on the synaptic inputs and reaction kinetics.

1.4.2 A snapshot is not enough: modelling with the time variable

The dynamic behaviour of a system, or its kinetics, is essential to determine and quantify its output. Simple but efficient examples are positive and negative feedback loops (Figure 1.8 panel B). In a positive feedback loop in which protein A activates protein B and protein B additionally activates A, the system often shows bistability: a system in which two stable states can exist for the same input. For such a system, usually referred to as a *switch*, a sufficiently high input can change the output from a low activation to a high activation state, which is maintained even after the original input has been reduced. However if the reaction kinetics are slow, the switch will not get activated, meaning that there will be only a gradual increase of the input before the system returns to the previous stable state. Another change in the behaviour of a loop is observed for negative feedback loops. In a system in which protein A activates protein B, and protein B inhibits A, if the inhibition is characterised by a small delay it will produce a gradual decrease in the output. Longer delays, however, will produce a transient or oscillatory response (Sauro and Kholodenko, 2004; Novák and Tyson, 2008).

Probably the simplest type of dynamic model that does not require kinetic data is

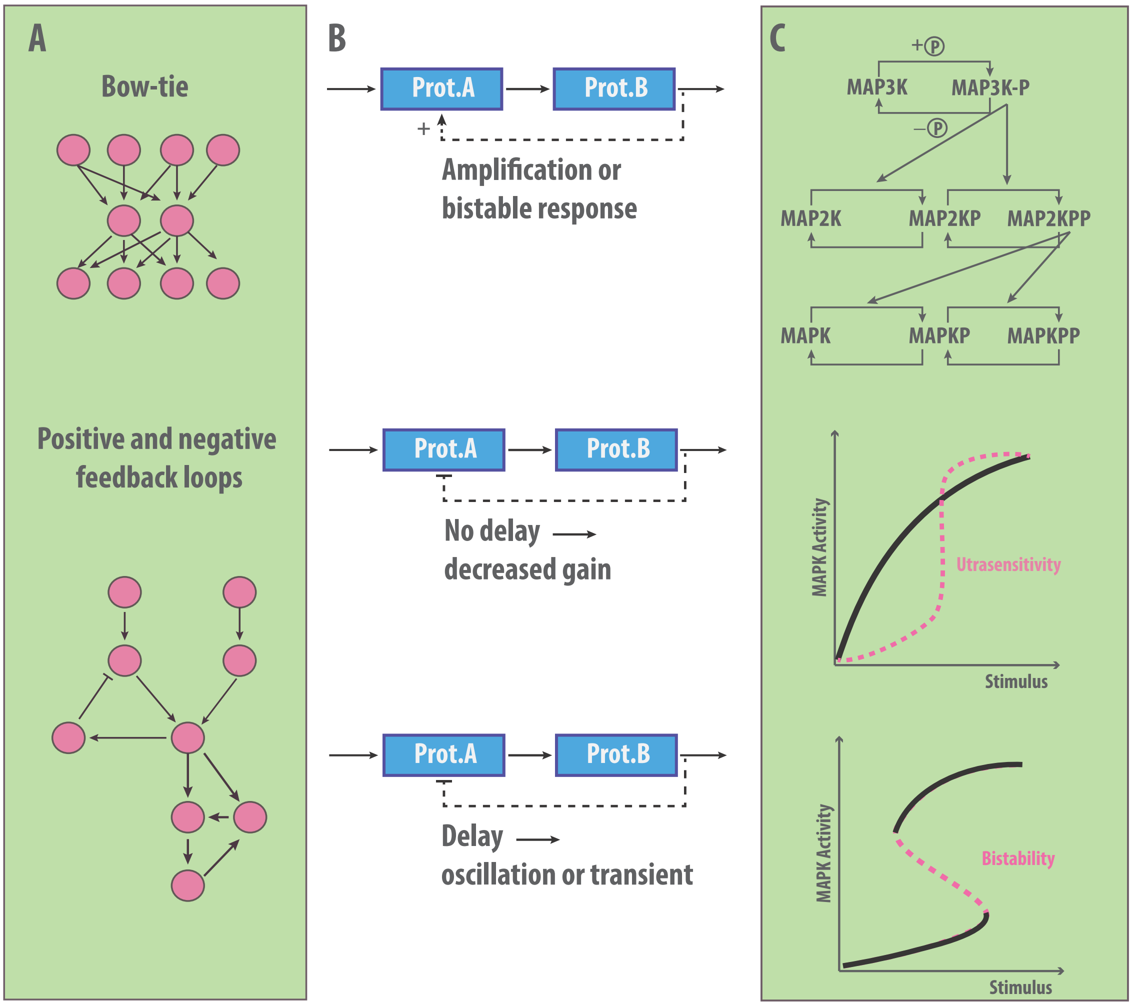


Figure 1.8: Network motifs, feedback loops and the MAPK cascade: examples of pathways motifs and computational units. Panel A: Two examples of common topology found in networks. At the top: bow-tie motif in which the signal coming from many proteins (e.g. receptors) converges onto few targets (e.g. second massengers) which can regulate many effectors (e.g. downstream pathways, transcription factors). At the bottom: positive and negative feedback loops motifs. Panel B: examples of positive and negative feedback loops and their response. Panel C: starting from the top, schematic representation of the MAPK cascade followed by examples of a deterministically simulated cascade response. The two plots differ only in their kinetic parameters. Adapted from Kotaleski and Blackwell (2010)

the boolean logic network. In a boolean network each node can assume only two states: present or absent, or ON and OFF. The dynamics of the network is simulated by using logical operators (for example protein A and protein B are both present at the same time) coupled with explicit time delays (for example protein C is present later). This type of modelling can be of use to study gene activation patters, or for large network for which quantitative data are not available.

A more mechanistic approach to dynamic modelling is to encode reactions using algebraic and ordinary differential equations (ODEs). The majority of the parameters present in an ODE system represent biochemical properties like association and dissociation constants, phosphorylation rates, enzyme turnover rates and so on, most of which are experimentally derived. Some other parameters might be more abstract, usually to represent observed behaviour that lacks a precise mechanistic description or to simplify such a mechanism (Bhalla and Iyengar, 1999). Many models have been made using ODEs, and an exhaustive collection of them can be found in the BioModels Database (Le Novère et al., 2006). Such models can be simulated using a numerical computing environment like MATLAB, or with biochemical simulators such as COPASI (Hoops et al., 2006). An interesting example is a model of the MAP kinases cascade shown in figure 1.8 panel C. The MAPK pathway is characterised by cycles of phosphorylation and dephosphorylation. Several iterations of the cycles, on different phosphorylation sites, amplify the signal and can result in *ultrasensitivity*, whereby a small change in the input near the threshold point results in a sharp change in the output. Depending on the kinetic encoded in the model, the cascade can show *bistability* instead of ultrasensitivity.

1.4.3 Sometimes it is one, sometimes it is three: stochasticity

A common misconception in modelling is to assume that the number of molecules in the observed system is always large enough to be expressed as a concentration, which is a continuous variable. For systems in which the molecule numbers are very low, the minimum variation that molecules can undergo (one molecule difference) become visible and a discrete approach in which molecule number can change by one integer at the time is preferable.

Many reactions that occur in neurons as well as in other cells, take place in small subcellular compartments or microdomains in which the number of molecules is very limited. One of such cases is the concentration of calcium inside a dendritic spine. At resting conditions, only few calcium ions are present in the spine. Moreover, during stimulation the amount of calcium that enters per spike can vary due to the stochasticity of all the processes involved, such as the release of neurotransmitters and the gating of channels.

The stochastic fluctuations in molecule numbers can largely affect the outcome of the system, as shown in figure 1.9 in which the same model was simulated using an ODE solver and with the Gillespie stochastic algorithm (Rao et al., 2002). In the case reported, the positive feedback loop was bistable in deterministic simulations but, when simulated stochastically showed a gradual increase in the concentration of the activator, instead of bistability. Stochasticity becomes especially relevant for low concentration systems. It has been postulated that the intrinsic noise of a system is proportional to the inverse square root of the mean concentration (Van Kampen, 1992), implying that low concentration systems are more influenced by noise.

The Gillespie algorithm is one of the most well known and used stochastic algorithm available (Gillespie, 1977). This algorithm is the first of a class of exact stochastic simulation algorithm (SSA), a set of algorithms that specifically attempts to describe the time evolution of a well-stirred chemically reacting system in a way that takes into account the discrete and stochastic nature of the system.
SSA is implemented in several simulation packages like COPASI, Cain[3] and eCell (Tomita et al., 1999).

1.4.4 Where do we go now? Spatial modelling

Another frequent but incorrect assumption is that biological systems are well-mixed, meaning that molecules are homogeneously distributed in the studied system. As mentioned before, cells are highly complex structures with many different subcompartments. As seen in the MAPK cascade example (subsection 1.4.2), the signal is transmitted from the membrane with the activation of receptors (such as NMDAR) to its final targets in the nucleus via a cascade of transcription factors. On the long way from the membrane to the nucleus the signal passes through different subcompartments with various effectors that participate in the final outcome of the pathway. Such compartmentalisation is a very

[3] http://www.cacr.caltech.edu/∼sean/cain/Welcome.htm

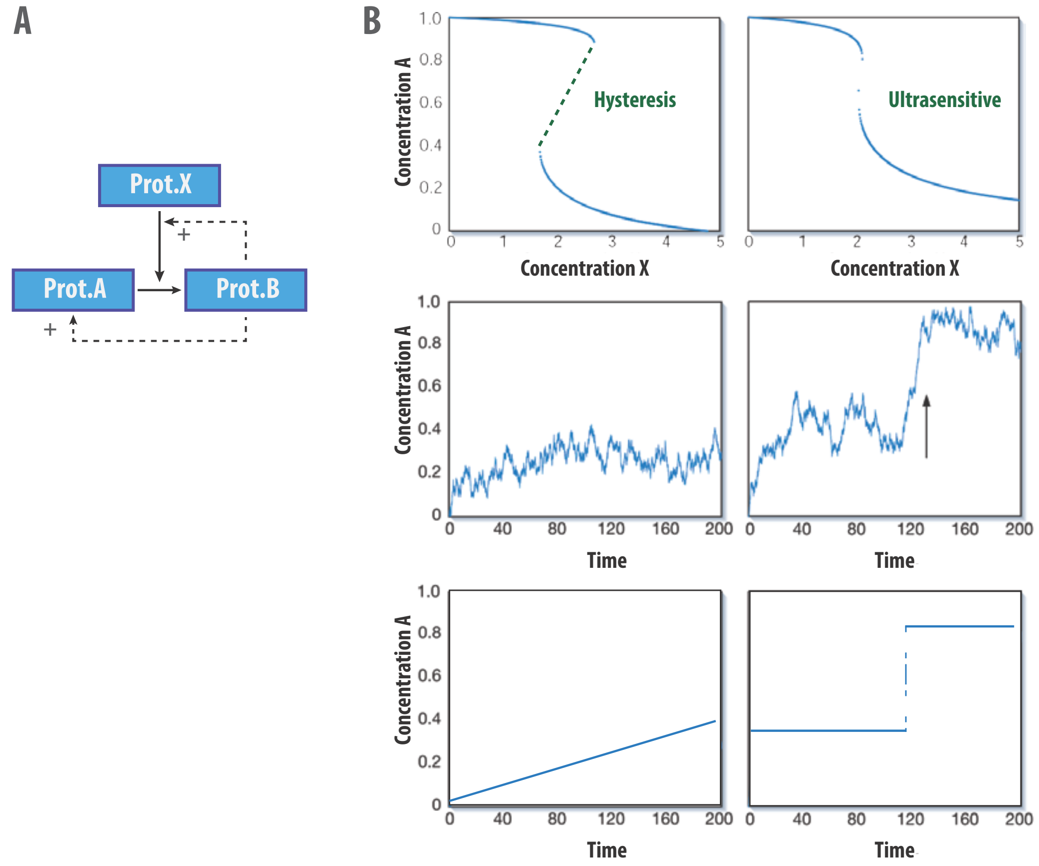


Figure 1.9: Comparison between deterministic and stochastic simulation of the same positive feedback loop: (A) Schematic representation of the feedback loop. (B) Top row: deterministic simulation of the system: on the left, a fast kinetic of the system resulted in hysteresis while on the right, with slower kinetics, resulted in an ultrasensitive response. Middle row: stochastic simulation of the system using the Gillespie algorithm: on the left the system does not show a bistable response, but a slow and gradual increase of the concentration of X; on the right, the ultrasensitivity of the system is conserved with the stochastic simulation, the arrow points at the activation of the switch. Bottom row: plots of the concentrations of A vs Time from the deterministic simulation, as comparison with the stochastic one. Adapted from Rao et al. (2002).

successful strategy to create microdomains that differ from the surrounding environment. For example, in certain microdomains the local concentration of certain molecules can be higher compared to the overall cellular one, which can allow the modulation of a particular reaction kinetics.
The interaction between molecules that come from two different compartments can depend on, or be affected by, the diffusion of molecules between those compartments. These are *diffusion limited* reactions in which the diffusion of the molecules is slower than their reaction kinetics. Segregating proteins in different compartments or forcing them together in a microdomain is common strategy used by biological systems to fine tune reaction kinetics (Nooren and Thornton, 2003).

Space can be integrated in modelling with different methods. The deterministic approach calculates the molecular concentrations as function of space using partial differential equations (PDEs) that contain unknown multivariable functions and their partial derivatives. The PDEs approach, due to its deterministic nature, cannot represent the fluctuations derived from the noise of biological systems. On the other hand, it can be a powerful tool to study large systems thanks to its low computational costs. A simulator that uses this approach is V-Cell (Schaff et al., 1997), which applies the finite-volume method[4] to calculate diffusion and reaction rates of molecules present in the system.

Another spatial simulation method are Cellular Automata (CA), which are a lattice-based methods in which each lattice cell can assume a finite number of states. The behaviour of each cell is dictated by a set of rules (or functions) that are updated at each time step. Stochasticity can be introduced via the encoded rules to generate a variation of the method known as *stochastic cellular automaton*. The grid size of the lattice can be modified as well as its shape (e.g. square, hexagon or triangles). Historically, this approach was only used for two dimensional simulations, but in recent years the 2D grid was expanded to 3D (Broderick et al., 2005; Love et al., 2001). CA have been used to simulate mesoscopic and microscopic systems (Malek Mansour and Baras, 1992; Schnell and Turner, 2004) and have been very successful in pattern formation simulations or for growth predictions (Wolfram, 1984) but at the best of my knowledge, have not been applied in explicit simulations on protein behaviour in crowded environments.

[4] Finite-volume method or FVM is a method for representing and evaluating PDEs typically used in computational fluid dynamics.

Crowded environments are defined as environment in which the molecule numbers reach such a high concentration that the molecules cannot diffuse freely. These phenomena are more common than previously thought. In fact, the vast majority of biological environments are crowded, including the cell cytoplasm (Zimmerman and Trach, 1991; Ellis, 2001). Molecular crowding is more accurately termed *excluded volume effect* because of the mutual impenetrability that all solute molecules possess, which makes their volume excluded from the available volume in which they can diffuse. How much of the intracellular volume is unavailable to other molecules depends on the numbers, sizes and shapes of all the molecules present in the studied compartment. Molecular crowding affects reaction kinetics and diffusion, making it an interesting factor to take into account during simulations (Dix and Verkman, 2008; Zhou et al., 2008a; Ellis, 2001).

Stochastic-reaction diffusions can be solved using a voxel-based approach in which space is divided in subvolumes. In each subvolume, the behaviour of a group of molecules is computed using chemical kinetics laws, and their diffusions is simulated as the transition from one subvolume to another. This approach is used for the simulators STEPS (Hepburn et al., 2012) and NeuroRD (Oliveira et al., 2010), which are specifically designed for computational neuroscience. Although useful, these methods lack in the possibility of tracking the behaviour of every single molecule present in the system, and can not be applied to molecular crowding related studies.

Another approach to spatial simulation is the one taken by Smoldyn (Andrews and Bray, 2004; Andrews et al., 2010) a particle-based stochastic simulator, in which molecules are represented as point-like particles that can diffuse in a continuous three dimensional space over fixed time steps. Smoldyn is based on Smoluchowski reaction dynamics (Smoluchowski, 1916) for which kinetic rates of molecular reactions are transformed into the radii of each molecule's reaction or binding radii. The higher the reaction propensity of a molecular species to react, the larger its binding radius. This approach has the advantage of taking into account of the stochastic nature of biological environments in space, but is not suitable for explicit molecular crowding studies, because molecules have no excluded volume.

The Smoluchowski reaction dynamics are a simplified version of Brownian dynamics (BD), which is another particle-base stochastic approach to spatial simulations. In BD molecules are represented as particles with a finite volume which exhibit noise as they are propagated according to the Langevin equation (Langevin, 1908). The model mimics interactions between the diffusing molecules and the implicit solvent with random forces. This approach can be used to effectively simulate crowded environments since molecules are explicitly represented in the simulation space. Example of successful studies that are based on BD are the effect of electrostatic competition between substrates binding to an enzyme (Elcock, 2002) and a study of a bacteria cytoplasm (McGuffee and Elcock, 2010). In their study, McGueffee and Elcock used the 50 most abundant cytoplasmic molecules types from E. coli for a total of 1008 individual molecules, and simulated their diffusion using steric and electrostatic forces and protein-protein thermodynamic interactions. One of the more interesting findings of this study is the discovery that proteins diffusing in crowded environment such as the cytosol show anomalous diffusion, which was previously thought to be a diversion of the norm. BD is clearly a very powerful tool to study protein behaviours by mimicking more closely the diffusive environment *in vivo*, but the accuracy of the calculations behind it makes it a very expensive computational approach, which cannot be used yet to compute large networks or whole cells.

Another technique worth mentioning for spatial simulation with an even higher level of details considered, is molecular dynamics (MD). In MD molecules are represented with an all-atom representation and the simulations are driven by a force field that takes into account atomic properties like partial charges and van der Waalls radii. Molecular trajectories are calculated integrating the N-bodies potential with Newton's laws over very small time steps. MD is commonly used in structural biology and drug design for computation of small conformational changes of proteins or to study the interactions between proteins and a small chemicals (Durrant and McCammon, 2011; Karplus and Kuriyan, 2005). Between all the mentioned approaches MD is by far the most accurate and the most expensive, making it a very powerful tool for an in depth understanding of molecular mechanisms but, practically unfeasible for large scale analysis.

1.4.5 How to move a particle in space: mathematical theory of diffusion

Diffusion is the random migration of molecules or small particles, due to thermal motion in a system. Independent of the representation of the particle, either point-like or with explicit volume, its mathematical characterisation of three dimensional diffusion can be split into three single components, one for each dimension (Berg, 1993).

Considering a particle that starts a time $t = 0$ and at position $x = 0$ a random walk can be defined with the following characteristics:

1. Each particle that undergoes a random walk, will move on the x axis with a velocity $\pm v_x$, meaning that every τ seconds, it moves either to a negative (left to the origin) or to a positive (right to the origin) distance δ.

2. The probability of going to the left or to the right, for each step is 50% and each step is statistically independent from each other. This principle categorises the random walk as an unbiased Markov process.

3. Each particle in the system moves independently from every other particle and the system is diluted enough that the particles do not interact with one another.

It is interesting to notice that a particle that follows the previous rules will on average not move, since the probability of going left is equal to the one of going right for every given step. Considering an ensemble of N particles, the position of the particle i after the nth step will be:

$$x_i(n) = x_i(n-1) \pm \delta \tag{1.1}$$

The mean displacement of the particles after the nth step can be calculated as following:

$$< x(n) > \quad = \frac{1}{N}\sum_{i=1}^{N} x_i(n) \tag{1.2}$$

Expressing $x_i(n)$ in terms of 1.1 we have:

$$\begin{aligned} < x(n) > \quad &= \frac{1}{N}\sum_{i=1}^{N}[x_i(n-1) \pm \delta] = \\ &= \frac{1}{N}\sum_{i=1}^{N} x_i(n-1) = \\ &= < x(n-1) > \end{aligned} \tag{1.3}$$

Equation 1.3 implies that the mean position of a particle will not change from step to step and since all particles started at the x=0 and spreading is symmetrical in respect of the origin point. In order to calculate the spreading of the particles or how much they actually travelled since t=0, we consider the root of the mean squared displacement. Since negative numbers when squared result in a positive value, the sum of all the squared displacement cannot be equal to zero.

$$< x^2(n) > \quad = \frac{1}{N}\sum_{i=1}^{N} x_i^2(n) \tag{1.4}$$

Which gives:

$$\begin{aligned} < x^2(n) > \quad &= \frac{1}{N}\sum_{i=1}^{N}[x_i^2(n-1) \pm 2\delta x_i(n-1) + \delta^2] \\ &= \quad < x^2(n-1) > +\delta^2 \end{aligned} \tag{1.5}$$

Therefore, the mean square displacement increases with the number of steps n and root mean squared displacement with the root square of n. Since we established from rule 1 that the particles will make n steps in a time that is $t = n\tau$, this tell us that the mean squared displacement is proportional to the time t, and that the root mean squared displacement is proportional to the square root of t.

In order to fully characterise the random walk it is necessary to calculate the probabilities that a particle will travel either to the left or to the right at different distances. If p is the probability of taking a step to right and q to the left, then q = 1 - p since the particle only has this two choices. The probability that a particle will step k times to the right in n trials, follows a binomial distribution:

$$P(k;n;p) = \frac{n!}{k!\,(n-k)!}p^k q^{n-k} \tag{1.6}$$

The number of times that a molecule will step to the right or the left is very high, in the order of millions of times per microsecond. When n is so large, binomial distributions have two asymptotic limits. One limit occurs when the product np remains finite even if $n \to \infty$, implying that p is infinitesimally small. This limit generates a Possion distribution. The other limit occurs when the probability of success p is finite, in this case when $n \to \infty$, also the product np will tend to infinity. This limit generates a Gaussian distribution. To derive the Gaussian distribution it is necessary to approximate the factorials of the binomial coefficients with Stirling's approximation:

$$n! \simeq (2\pi n)^{1/2}(n/e)^n \tag{1.7}$$

In which e is the base of the natural logarithms. Using this approximation gives:

$$P(k;n;p) \to P(k;\mu;\sigma)dk = \frac{1}{(2\pi\sigma^2)^{1/2}} e^{-(k-\mu)^2/2\sigma^2} dk \tag{1.8}$$

Where:

$$\mu =< k > \text{ and } \sigma = (< k^2 > - < k >^2)^{1/2} = (npq)^{1/2}$$

$P(k;\mu;\sigma)dk$ is the probability that k will be found in the interval between k and $k+$ dk in which dk is infinitesimally large.

Substituting $x = (1k - n)\delta, dx = 2\delta dk, p = q = 1/2, t = n/\tau, andD = \delta^2/2\tau$ gives:

$$P(x)dx = \frac{1}{(4\pi Dt)^{1/2}} e^{-x^2/4Dt} dx \tag{1.9}$$

This equation describes the Gaussian distribution of the probability of finding a particle between the position x and x +dx, at the time t. The mean of this distribution is zero, and the variance $\sigma_x^2 = 2Dt$ and standard deviation $\sigma_x = (2Dt)^{1/2}$. The distribution, shown in figure 1.10, when applied to each direction (x,y,z), gives the description of a three dimensional random walk.

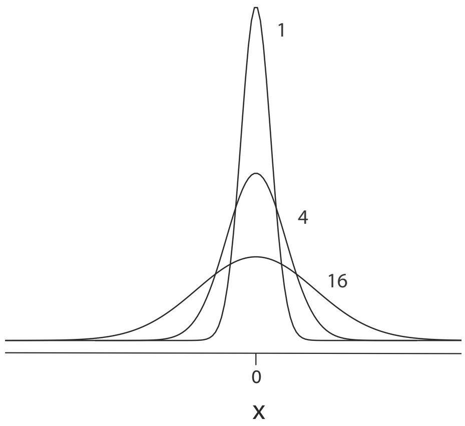


Figure 1.10: Relationship between time and particle position: probability of finding a particle at different x positions, at times $t = 1$, $t = 4$ and $t = 16$.

1.5 Not all that can be of use to neuroscience comes from science

The last few decades saw an impressive increase in the quality and complexity of video games and virtual reality simulations. Recently a new type of game started to populate the entertainment industry, the *scientific discovery games*. In these games the users help solve computationally challenging problems, in a crowdsourcing framework. In scientific discovery games scientific problems are turned into puzzles that even non-expert players are able to solve. The games are usually structured in several levels, with few of them dedicated to train and introduce the player to the scientific problem. One of the first examples of scientific discovery games is the Galaxy Zoo[5] (Lintott et al., 2011) in which users help to categorise galaxies from the vast amount of images stored in The Sloan Digital Sky Survey[6], a three dimensional map of the deep sky, containing more than 930,000 galaxies.

Another successful example in the category is Foldit[7] (Cooper et al., 2010), a game that aims at producing accurate structural models from protein sequences through gameplay. The idea behind it is that with the human innate spatial reasoning ability, the player can guide the search of stable protein folding in the vast protein conformational space. The player is asked to fold a protein sequence according to certain criteria. The score that the player is seeing is made by the Rosetta algorithm (Leaver-Fay et al., 2011) and the best models are then reported back to the research group for further analysis.

A similar strategy was used in the development of NanoDoc[8] a game in which the user helps to develop new anticancer nanoparticles. The user can change different parameters that characterise a nanoparticles (e.g. size, concentration, coating) and then simulate the efficacy of the virtual treatment inside the game. In this way the player guides the choice in a very large parameter space, and like for Foldit, the best models are fed back to the lab for further analysis.

Scientific discovery games are not the only category of games that has by far exceeded the entertainment label. Many games are made for high-end education, like the Virtual

[5] http://www.galaxyzoo.org

[6] http://www.sdss.org/

[7] https://fold.it/portal/

[8] http://nanodoc.org/

Operating Room[9], an online simulation platform for medicine students which focus on risk assessment in the operating room; or like the Laparoscopic Adjustable Gastric Band Simulator(LAGB), a virtual reality simulator for training in gastric laparoscopy. The LAGB simulator is based on the physic engine PhysX and is able to give the user a haptic feedback (use of force, vibration or motion to the user), which makes the virtual reality experience even more engaging and realistic (Maciel et al., 2009).

Another example worth mentioning of the use of game and virtual reality is in the rehabilitation of post-traumatic stress syndrome (PTSD) patients. Virtual reality is used to safely and gradually expose the patients to the traumatic environment and events as in exposure therapy. One of the pioneers of the field is Barbara Rothbaum head of the Trauma and Anxiety Recovery Program (TARP) at Emory University, which specialise in combat-related PTSD and sexual abuse traumas. In her studies she found that the combination of virtual reality and medications, is more effective than more traditional approaches (Cukor et al., 2009; Rizzo et al., 2014).

The main aim of the work presented in this thesis was to develop a new modelling approach to protein-protein interactions in the contest of neurobiology and more specifically in the post synaptic density environment. In the development of this new methodology, tools commonly used in game development and computer graphics were converted to simulation engine, and graphics and protein modelling software. In the next chapter, I will explain in details why I choose these tools and how I utilised them into the proposed new methodology, as well as a detailed explanation of the methodology pipeline and its simulation algorithm. In following chapters I will then present the methodology validation as well as its application to the study of the Post Synaptic Density assembly.

[9] http://3dvor.univ-jfc.fr/

Chapter 2

Novel approach to modelling protein-protein interactions in biological space

> If you can dream it, you can do it.
>
> Enzo Ferrari

The main aim of this work was to develop an accurate and detailed method to simulate protein-protein interactions and the formation of large complexes, and use it to study the properties of PSD formation in a dendritic spine. Dendritic spines are small specialised compartment of neurons (section 1.2), with a mean volume that can vary between 0.001 and 1 μm^3. With such a small volumes, the number of molecules present in the spine are too small to be represented with continuous values therefore a stochastic approach is required for accurate simulations of dendritic spines (section 1.4).
Moreover, it is known from experiments (section 1.3) that the dendritic spine environment is not well-mixed, but different proteins are enriched at different positions relative to the membrane. As mentioned in (subsection 1.4.4), when a system is not well-mixed space needs to be explicitly incorporated into the model, allowing micro-environments with different properties to emerge. Considering the low concentration of molecules and the need to incorporate the space variable, the best modelling approach for this system is a stochastic particle based spatial simulation. In such simulations each molecule is represented as a single distinct entity that moves stochastically in space, for which is possible to track every movement and consequently analyse it. Using a stochastic particle based spatial simulation will allow to study properties of the system such as co-localisation of molecules, local aggregates and so on, that could not be possible to study with other approaches.

In order to accurately study the formation of a complex multiprotein structure like the PSD, the monomers should be represented with an accurate shape. The use of accurate molecule-like shapes brings some advantages compared to the more classic approach of using dimensionless points as in Smoldyn. First of all, when three dimensional shapes are used, exclusion volume effects can be taken into account. These effects, such as crowding and confining effects, are known to influence dynamic properties of the system (Zhou et al., 2008b; White et al., 2010). Moreover, using accurate shapes results in a more accurate final geometry of the PSD that can be used for further modelling purposes, such as simulations of confinement effects of signalling proteins in the PSD.

There are already available methods to study protein-protein interactions based on protein geometries such as molecular dynamics or docking (Moreira et al., 2010; Karplus and Kuriyan, 2005). Both methods require an all-atoms representations of the proteins and although quite useful for binding conformation predictions and binding affinity estimations, they fail to simulate large assembly unless for time scale in the order of nanoseconds, or to study interaction kinetics. The main reason why these approaches are not applicable to the study of large multi-complex formation or binding kinetics is the very high computational costs of their simulations. To date, the fastest molecular dynamics simulation ever performed used a special-purpose supercomputer, called Anton, designed for molecular dynamics (Dror et al., 2012). Even with such architecture a one second simulation of only 5 instances of the tetrameric protein Homer, a common protein found in the PSD, would require two hundreds thousands days, since the performance of this system is $5\mu s$ of simulated time per day (Shaw et al., 2009).

Instead of using an all-atom representation of the proteins it is possible to approximate the protein geometries with simpler shapes. However, using molecule-like shapes is still computationally expensive and none of the available software allows for the use of such shapes in the context of multiprotein complex formation. When complex shapes are used in simulations involving binding processes, both translation and rotation needs to be taken into account when computing diffusion (subsection 2.6.6). The vast majority of the software that can simulate diffusion and binding of complex shapes does not take into account the rotational component of diffusion (Byrne et al., 2010; Tolle and Le Novère, 2010). Another essential feature of a suitable simulator is a sound binding strategy. As proteins are represented with their proper shape, the binding should be carried out not at random positions on the protein geometries, but on their specific binding sites. To achieve this, binding site areas should be defined as distinct surfaces on the 3D protein geometries with specific characteristics assigned to them. Lastly, in order to simulate protein-protein interactions, of relatively large scale, or big multi-protein complexes, a strategy is required to deal with the combinatorial explosion caused by all the possible interaction combinations and their intermediates.

In the present chapter, I discuss the method I developed to address this lack of methodology, taking advantage of tools already available in the computer graphics field. The methodology is divided in two main parts as represented in figure 2.1. The first, carried out in Autodesk Maya (section 2.1), consists of the preparation of the protein geometries. The second part, carried out in Unity (section 2.2), consists of converting those geometries into entities with the desired behaviour and then running the simulations.

Validation of the methodology described in this chapter will follow in chapter 3 and the application of the method for the study of the PSD assembly will follow in chapter 4.

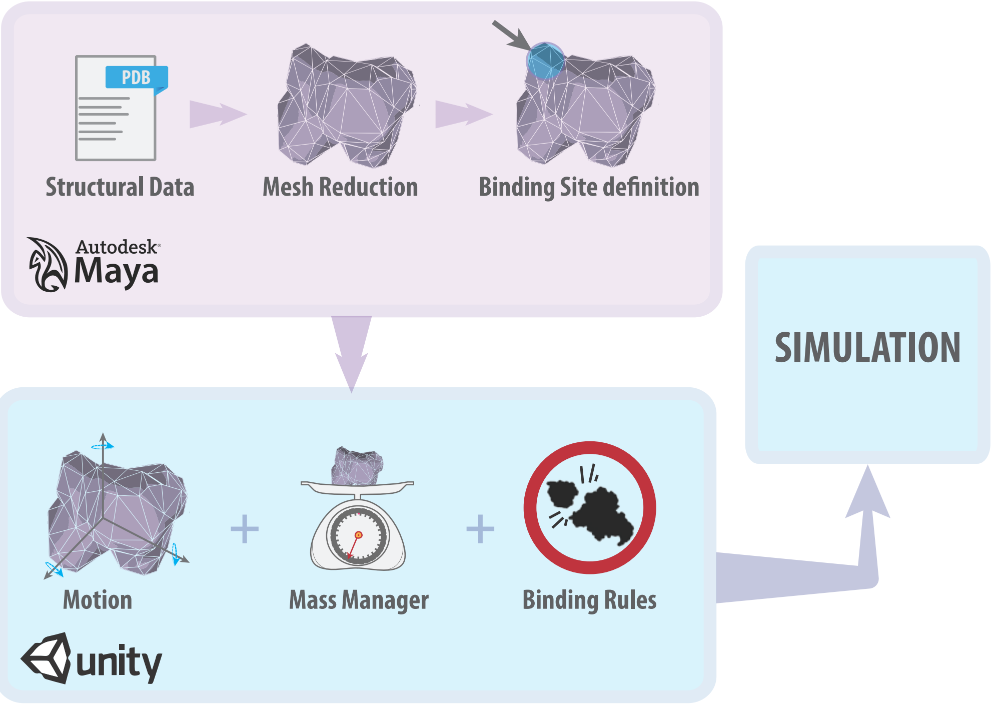


Figure 2.1: Pipeline Overview: The first part of the pipeline is carried out in Autodesk Maya (colour coded in light purple). Structural data of proteins are retrieved from the Protein Data Bank and imported into Maya. Protein surfaces are generated and reduced in their level of detail. Subsequently the binding sites are defined and mapped onto the protein surfaces. The resulting protein models are loaded into Unity where the modules for motion, mass managing and binding rules are added. Once all the proteins of interest are loaded and ready, the simulation is ready to start, directly in Unity (colour coded in light blue).

2.1 Introduction to Autodesk Maya

Maya is a 3D computer graphics software program from Autodesk[1], the company that made, and still develops, the famous 2D and 3D computer-aided design (CAD) and drafting program AutoCAD. Maya, which takes its name from the Hindu concept of illusion, is an extensible production platform for computer animation, modelling, simulation, rendering and compositing. It is mainly used for content creation for video games, in animated film and visual effects for films and television. The last 18 Academy Award winners for Best Visual Effects have used Autodesk's 3D, animation and/or visual effects software[2]. All five 2013 nominees for best animation used Autodesk's Maya.

Maya represents data and operations internally as *nodes*. A node is the building block of a Maya project: multiple nodes can be connected to each other so that the data from the first node is fed into the next and so forth, to the last one. This collection of connected nodes creates a workflow known as *Dependency Graph*. For example there is a time node, which keeps track of time during animation, and a deformation node, which deforms with certain modifiable parameters, the geometry which it is acting on, and so on. In Maya everything from geometry and its animations, to rendering settings are all linked together in the Dependency Graph. Having such a Dependency Graph is what makes Maya so flexible.

All the commands executed in Maya, even from the graphical interface, are in its cross-platform scripting language: MEL (Maya Embedded Language). Usually plugins and available toolkits are written in MEL, but Maya also supports the use of Python-style scripting. A complete, fully functional version of Maya with Educational license, for non-commercial use only, can be freely obtained from the educational centre of Autodesk[3]. A toolkit called Molecular Maya and a recently developed plugin called ePMV are available to aid protein structure manipulation with automatic PDB import features. These tools not only turned Maya into a molecular visualisation program, but provided a means for extensive manipulation and animation of protein structures, taking advantage of the inbuilt features of Maya.

Maya is not the only software available for 3D modelling, animation and rendering.

[1] http://www.autodesk.co.uk/products/autodesk-maya/

[2] http://www.forbes.com/sites/connieguglielmo/2013/02/22/autodesks-software-has-starring-role-in–2013-oscar-nominees/

[3] http://www.autodesk.com/education/student-software

Many other software products have been developed in the last decades, and probably the most famous are CINEMA 4D from MAXON[4], and Blender from Blender Foundation[5]. While Blender is a free and open source software, CINEMA 4D does not provide free licenses for educational purposes. The plugin ePMV is also available for Cinema 4D, while a different version of Blender for molecular visualisation and manipulation called BioBlender has recently become available (Andrei et al., 2012).

In recent years Maya has also been used for scientific and educational purposes. One of the main repositories of work on scientific visualisation, carried out in Maya and other software, is the MolecularMovies website[6]. Probably the most famous animations are from Drew Berry[7] in which topics like apoptosis, transcription, replication and the life cycle of the Malaria parasite are covered. Other prominent videos that had high visibility are: the kinesin mechanism from Graham Johnson[8], in which he shows a molecule of kinesin walking along a microtubule protofilament exchanging molecules of ATP and ADP; and the Inner Life of the Cell from the Biovision Lab at Harvard University[9], in which different types of cellular dynamics are explained.

Although the works listed above made use of PDB structures for the represented proteins, and the explained processes are biologically correct, they can only be classified as animations, since the interactions and processes showed are the result of manual positioning from the authors. To the best of my knowledge, only one study has been made using computer graphic software that is classified as simulation: the work of Jason Sharpe and Charles Lumsden (Sharpe et al., 2008). In their work, they simulated fibroblast cells moving in extracellular matrices of different densities. All the work was carried out in Maya and led to the publication of a unique book with a thorough description of the methods used.

I chose to carry out some of the work presented in this thesis, in Maya rather than another software program, due to its flexibility in 3D geometry manipulation, the large amount of freely available learning aids, the availability of a toolkit for PDB import and manipulation and lastly for the availability of a free fully functioning educational license.

[4] http://www.maxon.net/?id=1499

[5] http://www.blender.org/

[6] http://www.molecularmovies.com/showcase/

[7] http://www.molecularmovies.com/movies/viewanimatorstudio/Drew%20Berry/

[8] http://www.molecularmovies.com/movies/viewanimatorstudio/Graham%20Johnson/

[9] http://multimedia.mcb.harvard.edu/

The 2010, 2011 and 2012 versions were used at various stages of this work.

2.1.1 Importing PDB structures in Maya with Molecular Maya Toolkit

Molecular Maya (or mMaya) is a freely available software toolkit for Autodesk Maya developed by Gael McGill. The toolkit helps with the import, modification, and animation of proteins encoded as PDB files. mMaya is written in Python and MEL and its main purpose is to turn Maya into a protein visualisation program, taking full advantage of the features that makes Maya such a powerful CG software program.

Possibly the main feature of mMaya is in how the PDB files are imported. The molecules geometry in mMaya retains a live link to the underlying PDB dataset, and this is fundamental for switching representation types and for general modification of the structure.

A customised type of Maya node is automatically created appositely to store most of the informations in PDB files, from geometry to experimental data. mMaya, among other things, transforms each atom represented in the PDB into a particle maintaining its 3D coordinates. These particles are part of a particle system, which in computer graphics programs is used to create complex phenomena, like fire, water and smoke (Wei et al., 2002; Kruger et al., 2005). For this work, I mainly took advantage of the surface representation of mMaya. Mesh surfaces are generated using the pre-build feature of Maya to generate a mesh surface from a particle system. A polygon mesh is calculated from the particle system using the marching cubes method (Lorensen and Cline, 1987). The marching cubes algorithm is a widely used method for 3D representation in fluid and particle dynamics as well as for medical visualisation of magnetic resonance imaging and computed axial tomography scans. The algorithm converts the particle system into a scalar field, a function of space in which each point has a scalar quantity for value, into imaginary cubes. For each point in the scalar field, the algorithm processes the point and its 7 neighbour values, forming an imaginary cube (8 vertices per cube). Subsequently, the algorithm checks if and where the isosurface, a surface that represent points at the constant value such as density within a volume of space, intersects the 12 edges that describe the cube. Successively the algorithm calculates the polygon (or polygons) needed to represent that part of the isosurface that intersect the cube. Subsequently, all the individual polygons are merged together to obtain the final mesh.

The toolkit allows the user to tweak some parameters that influence the mesh generation. These parameters act directly on how the particle system is converted into a mesh. It is possible to change the *resolution* of the mesh by changing the gird size used to create the polygonal mesh. The smaller the voxel size used in the grid, the higher the resolutions produced, resulting in smoother surfaces, with smaller mesh triangles and higher polygonal count. Another parameter that can be tweaked in mMaya is the amount of *smoothing* applied to the output mesh. Smoothing a surface results in a more topologically uniform and polished surface. The smoothing is controlled via three different parameters: *Smooths* that controls the lengths of the triangle's edges; *Threshold* that influences the density of overlapping particles, and *Blobby rad* that controls the radius at which the surface is calculated from each particle . In all cases increasing these parameters results in a smoother surface. Molecular Maya was used in the work presented in this thesis to create the protein geometries, while ePMV, presented in the following section, was used to define the binding sites location and their areas onto the protein geometries.

2.1.2 Sequence selection with the Embedded Python Molecular Viewer ePMV

The embedded Python Molecular Viewer (ePMV) is a free and open source software developed by Graham Johnson at the Olson Laboratory of the Scripps Research Institute (Johnson et al., 2011). ePMV is a cross-platform tool that runs molecular modelling software directly in several professional 3D animation applications like Maya, CINEMA 4D and Blender (Autin et al., 2012).

Similarly to mMaya, with ePMV it is possible to automatically load PDB structures into Maya. The loaded proteins can be manipulated in terms of atom, backbone, and surface representations. The classical colouring schemes for proteins such as CPK or residue type are present along with more sophisticated ones, like a temperature factor based. Moreover, several extensions are available for ePMV such as one for protein-ligand scoring with AutoDock or PyRosetta, a Modeller optimisation, and a molecular dynamics visualisation extension. For this work I took advantage of the sequence selection feature of ePMV for highlighting binding site sequences on proteins. This feature, which is not present in the Molecular Maya Toolkit, allows for the selection of part of the protein sequence using regular expressions. Once the sequence is highlighted, a different representation can be applied to it, resulting in a very clear geometrical distinction between the highlighted sequence and the rest of the protein geometry.

2.2 Unity: the game engine used as simulation environment

Unity is a game engine developed by Unity Technologies[10]. A game engine is a software dedicated to the creation and development of video games, for different platforms, consoles, personal computers and mobile phones. A game engine is a software framework that abstracts details that are common to all games and makes them reusable in order to make the development of a video game easier and faster.

Creating a video game is rather complex and involves the assembly of several parts such as objects, animations, and sounds, that interact together to make the gaming experience. These parts can be thought of as several layers which their assembly is done inside the game engine. The most visible layer in a game is what we can actually see, the graphics. For each game level, a new environment or terrain is usually developed using a 3D computer graphics application like Maya. The same type of application is also used to create the main and other possible characters in the game.
The graphics need to be rendered inside the game (mostly in real time) by the render engine, in either 3D or 2D, depending on the game type.

The most important keyword in a game is interactivity, denoting in the interaction between the player and the game environment. This means that movements are at the core of a video game. Movements inside a video game can be achieved in two different ways. One, is through animation where movements are scripted in advance and encoded in a way that can be recalled at the required time. This type of animation is for example commonly used to make the idle/walking/running/jumping cycles of the player's character, which is triggered by the player's input. Another way to generate movement is to rely on physics, where movements are based on the starting conditions of the scene and physics laws that are encoded in the game. The physics layer is integrated in a game using a physics engine which like the render engine, is a dedicated software integrated into the game engine.

There are many games that rely heavily on a physics engine, and even when the game is not physics driven, the physics engine is always at the foundation of the character-environment interaction. A famous example of a physics driven game is the 2D game

[10] http://unity3d.com/

Angry Birds[11]: the player sets the angle and velocity at which a bird is thrown towards some objects. What happens next is only based on simple physics laws like gravity, friction and energy transfer upon collisions.

While the above game is physics driven, basically every game even when not physics driven extensively relies on the physics engine. The opening of a door or picking up of an object are other common examples of how physics is at core of the interaction between a player's character and environment. In both cases the character needs to get close enough to the object to trigger a collision between the character and the object itself. This collision is detected by the physic engine of the game and upon this collision an action, such as the opening of a door, is triggered.

Another omnipresent game layer is sound, while less noticeable are other common components such as artificial intelligence (AI), networking, streaming and memory management. All these parts need to be assembled together carefully and a game engine is the software designed to do it with. There are several game engines available and they can be categorised by the level of abstraction that they use. While the higher level game engines are easier to interact with, low level game engines are usually more flexible. Unity is one of the high level game engine available, like GameMaker[12], and Torque Game Builder[13]. Unity was recently used as visualisation tool for different biological dataset from protein structure (prepared separately) to protein networks using the Unity extension UnityMol (Lv et al., 2013).

I choose Unity to be the simulation engine for the work reported in this thesis over other software for its high compatibility with Maya, its good performances as 3D game engine, the availability of a free and functional version, the vast user community that provides a great support and the large availability of excellent learning tools such as tutorials and books. Version 3.0 to 4.1 were used at various stages of this work.

2.3 Definition of a mesh

In 3D computer graphics, 3D modelling is a way to mathematically represent the surface of three-dimensional objects, using specialised software. A common type of 3D modelling

[11] http://www.angrybirds.com/

[12] https://www.yoyogames.com/studio

[13] http://www.garagegames.com/products/torque–2d

is polygonal modelling in which simple polygons, usually triangles or quadrilaterals, are connected edge to edge to create a mesh of complex shape.

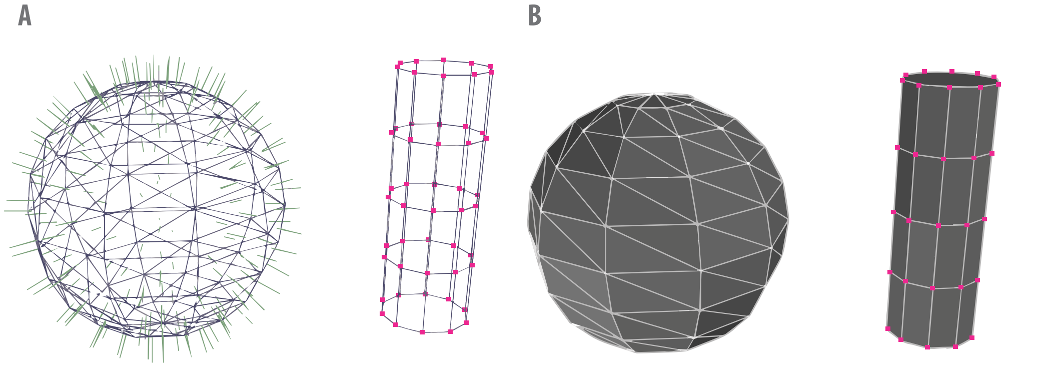


Figure 2.2: Mathematical description of geometrical entities. Panel A: wire-frame representation of a polygonal sphere composed of triangles, and a cylinder composed of quads. Highlighted in green, on each face of the sphere, the normals, a vector perpendicular to the face plane, starting from its centre. Highlighted in pink, on the cylinder are the vertices that composed the mesh. Panel B, solid representation of the same geometries, the contours of each face is highlighted in white, for both polygons.

The fundamental definition of a polygonal mesh is a vertex, a point defined in the three dimensional space (see figure 2.2). The line that connects two vertices is called an edge, and when three vertices are connected with each other by three edges, they define a triangle. Triangles are the most common type of polygon used in 3D modelling, since they are the simplest polygon that can be created in Euclidean space and have the property of inhabiting a single plane. The flat nature of triangles aids the determinations of face normals, a 3D vector perpendicular to the polygon surface. Normals are used to determine the face orientation of a mesh towards an external source which, in computer graphics, is often used in the rendering process to calculate how light interacts with the mesh surface. Quadrilateral polygons, often referred to as quads, are becoming more and more common in polygonal modelling even if they might not lie on a single plane. In the context of a polygonal mesh, a single polygon is called a face and the total number of faces required to create a mesh is called polygonal count or poly count. The polygonal count is used as an indication on the amount of information encoded in a mesh: the higher the polygonal count the higher the amount of details encoded.

2.4 Maya workflow: protein model preparation

The first part of the pipeline is focused on preparing the protein models so they can be simulated inside Unity. A schematic representation of this part is depicted in figure 2.3. The first step of the protein model preparation workflow is to retrieve structural data for all the proteins of interest. Protein structural data can be found in the Protein Data Bank (Berman et al., 2003), a unified worldwide database where solved structures are deposited in the PDB file format. A PDB file contains atomic coordinates of all the atoms of a protein, data that may be derived from X-Ray or NMR experiments. A PDB file is all that is required to have a complete profile of the structure, atomic composition, total and local charges and sequence of a protein.

Not all the protein structures are yet deposited in the Protein Data Bank, but they may be computed *in silico* if necessary, using known structures as templates. This computational approach, called homology modelling is a knowledge-based method for protein structure prediction (Johnson et al., 1994). The sequence of a protein with unknown structure (protein target) is aligned to several proteins with known structure (templates). This method is based on the assumption that closely related proteins share the same protein structure (Xu et al., 2000; Wiltgen and Tilz, 2009). Between the several platforms available for homology modelling calculations, the one of choice for this study was I-TASSER (Roy et al., 2010).

I-TASSER is an automated homology modelling server that takes the protein target sequence as input and provides five structural models as output. It was ranked as number one server for protein prediction in CASP7, CASP8, CASP9 and CASP10[14] experiments.

The homology modelling pipeline of I-TASSER is divided into three consecutive steps: firstly, the target sequence is matched against a non redundant sequence database which generates a sequence profile. A dataset of templates is retrieved using this profile. The second step of the pipeline is the alignment between the target and the templates. Continuous fragments of the alignments are subsequently excised from the template structures to generate an assembly of structural conformations. *Ab initio* calculations are also used at this stage when required. In the last step, the structure is reassembled using the conformation assembly and subsequently refined for the global topology as well

[14] Critical Assessment of protein Structure Prediction (CASP) experiments aims to establish the current state of the art in protein structure prediction by providing a benchmark competition every two years. CASP10 was hosted in 2012. http://predictioncenter.org/

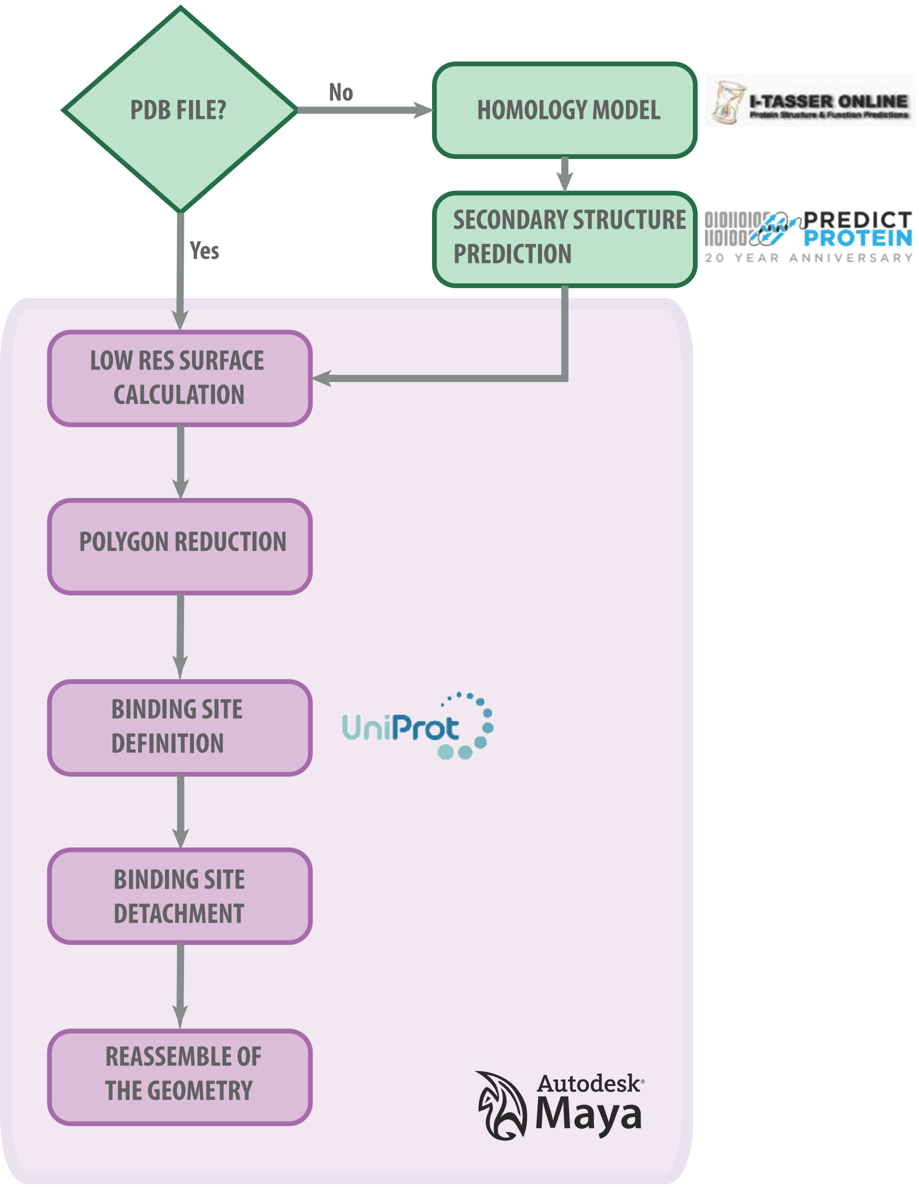


Figure 2.3: Schematic overview of the Maya Protein Preparation workflow: If a PDB file is not available then proceed to start the homology model procedure using the online server I-TASSER, the result from I-TASSER are scored according to how well present the secondary structure elements predicted by the algorithm Predict Protein. Once the PDB is obtained and imported into Maya a low poly surface is calculated. The surface is then further reduced to reach a poly count in the order of 250. Using the information stored in UniProt about the binding site sequence, a binding site is cut out of the geometry and then reassembled in a hierarchical fashion. Colour scheme: every task that was performed in Maya is highlighted in light purple while everything else in green.

as for any steric clashes.

I-TASSER lists five resulting structure models as output. In order to select one of the models produced by I-TASSER, the five calculated models were subsequently analysed based on the external secondary structure sequence prediction server PredictProtein[15] (Rost et al., 2004). The models were scored based on the correct match between the secondary structure prediction and the structure present in the 3D models.

The structural information needed by Unity, to simulate protein-protein interactions, is the protein surface. Surfaces are commonly used to represent protein structures, especially when needed to emphasise a pocket-like structures or visualise a protein-protein complex. There are many molecular visualisation programs that can produce surfaces from PDB files, such as Chimera (Pettersen et al., 2004) and VMD (Humphrey et al., 1996). These programs generate accessible or Connoly surfaces with good accuracy and a high level of detail. A Connoly surface is a surface generated by taking a sphere of the radius of a water molecule (1.4 Å) and rolling it to the van der Waalls representation of the desired molecule (Connolly, 1983; Connolly, 1993), as shown in figure 2.4.

Unity on the other hand, can work with two different types of surfaces: a very detailed one, with high polygon count, and a low definition one, with low poly count. In case of the high poly count geometries, the surface is not used to calculate collisions, but instead one needs to manually define the boundaries of the geometry, using primitives like cubes, spheres and cylinders. With this technique the high polygonal surface is only used for rendering and not for collision calculations, which can be ideal in cases when the accuracy of the simulations are less important than the visual results. In case of the low poly geometry, the surface is used directly as boundary for collision calculations. The requirements for the low poly geometry are that the poly count does not exceed 255 triangles and that the protein present a convex geometry.

In order to fulfil those requirements and therefore allow the collision calculations to be based on protein geometries, is necessary to drastically reduce the level of details from the high poly count Connoly surfaces. To do so, I set up a procedure to consistently generate low polygonal count meshes from PDB files (subsection 2.4.1).

The last part of the protein model preparation workflow aims to create separate

[15] Predict Protein is an automatic service for protein secondary structure and function predictions. The secondary structure is inferred by generating multiple alignments with several databases and using secondary structure prediction algorithms as well as solvent accessible prediction algorithms.

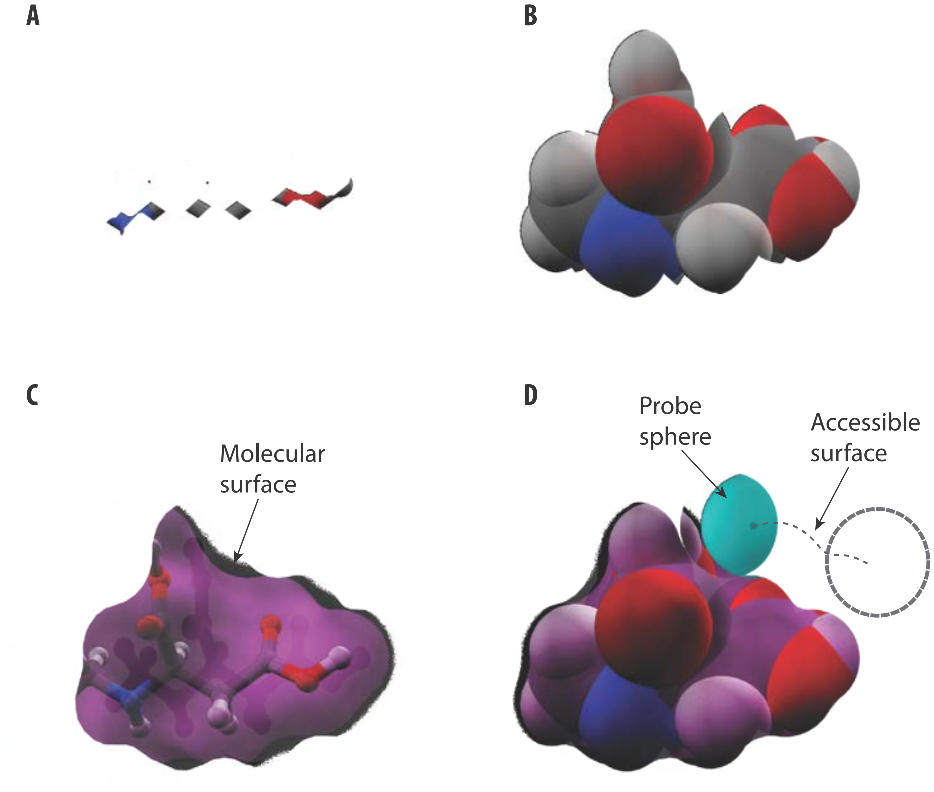


Figure 2.4: Different molecular representation. Panel A: Ball and stick representation of the small chemical compound N-Methyl-D-Aspartatic acid NMDA. Atoms are represented as spheres standardly coloured according to the atom types (Carbon grey, Oxygen red, Hydrogen white, Nitrogen blue) while bonds between atoms are represented as cylinder. Panel B: van der Waals (vdw) representation of the same molecule NMDA, also known as CPK or space filling. Each atom is represented as a sphere with a radius equal to their vdw radii (r˙w) The contour of the overlapping vdw spheres create the vdw surface. Panel C: Molecular surface of the NMDA molecule overlapped to the ball and stick representation. Panel D: Molecular surface and solvent accessible surface of the NMDA molecule. Both surfaces are calculated letting a probe sphere, with a radius of 1.4 Å, rolling on the van der Waals surface of the molecule. The inward trace of the probe will give the molecular surface, while the accessible surface is defined by the trace generated from the centre of the same probe.

geometries for the possible binding sites of interest on the protein. This separation is required by Unity, in order to keep the behaviour of each binding site independent from each other. Data on the sequence forming the binding sites were retrieved from the Uniprot database and mapped onto the protein surface. Subsequently each binding site was cut and separated from the surface (subsection 2.4.2). The separated binding sites and the remaining part of the protein geometry are then reassembled in a hierarchical structure which is fundamental for the simulation in Unity (subsection 2.4.3).

2.4.1 Low resolution surfaces generation

As previously mentioned, several available programs are capable of producing a high resolution surface of a protein, but none of them allow for a sufficient polygon reduction. Usually the only controllable parameter in creating a protein surface is the vertex density, defined as the number of vertices per square angstrom. The vertex density is by default set to 2.0 and usually restricted to a minimum value of 0.3. Even with the minimum allowed setting, it is impossible for these programs to produce a mesh with a polygon count around 250 triangles, as the surface poly count of a medium sized protein can be easily over one hundred thousand polygons, as shown in the example of figure 2.5.

Due to the need for the mesh reduction, I required a software program that allowed me to modify three dimensional geometries with high precision. Although not commonly used in scientific research, computer graphics programs are highly developed in this area and offer very powerful tools for manipulating 3D geometries. The software of choice, in the computer graphics category, was Autodesk Maya (section 2.1). Apart from its inbuilt features for modelling, animations and rendering, Maya had the advantage of having two toolkits for PDB manipulation: molecular Maya (subsection 2.1.1) and ePMV (subsection 2.1.2) which were both used for this pipeline.

Using mMaya is possible to generate a low resolution mesh directly from a PDB. Even when the mesh is produced using the low resolution setting, the surface still retains quite a high poly count. As shown in figure 2.5, when the surface of PI3K is calculated as Connoly surface with standard setting the resulted poly count is 184298; when it is calculated using mMaya with the low resolution settings, the count drops to 3516 triangles. Although tweaking the parameters that influence the mesh calculation in mMaya leads to lower resolution meshes, using the same parameters settings on different proteins never gave a similar result and moreover, never within the limit of the 255 threshold.

Maya itself has an inbuilt algorithm for polygon reduction. This algorithm reduces the number of polygons on a geometry while keeping the borders and overall shape as close as possible to the mesh input by preserving, when possible, the original vertex positions. The amount of reduction applied is proportional to a percentage specified by the user.

The poly count reduction is an active field and new improved algorithms are made

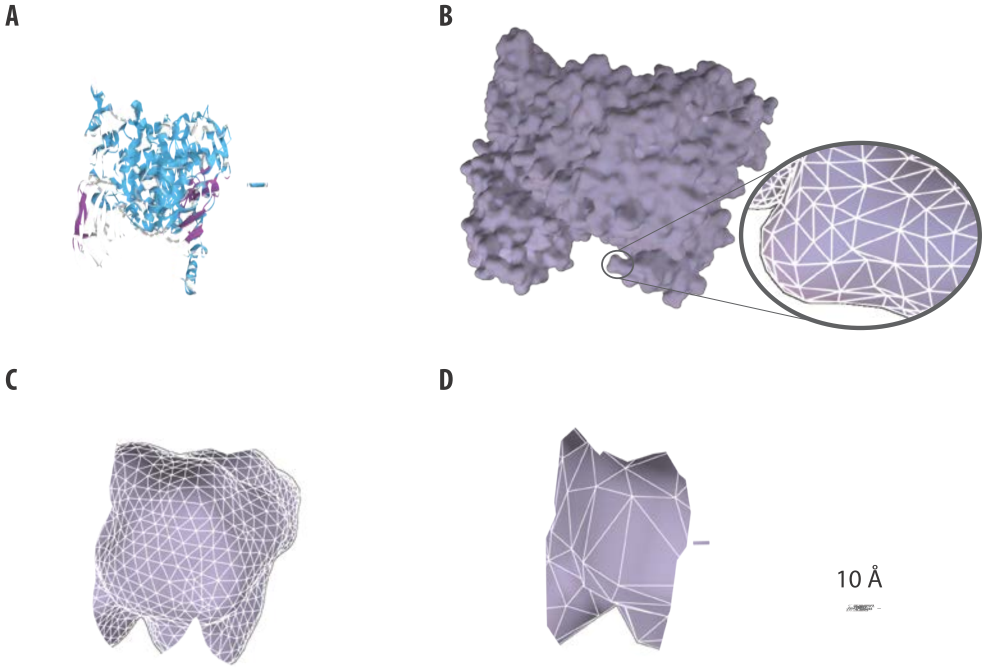


Figure 2.5: Level of detail modelling protein surfaces. Panel A: Ribbon diagram of the phosphoinositide 3-kinase (PI3K), PDB ID 1E8Z. Alpha helices are represented in blue, beta sheets in pink and loops in light grey. Panel B: Molecular surface of PI3K protein. The surface is calculated using the standard method of rolling a probe sphere of 1.4 Å radius on the van der Waals surface, keeping a triangle density of 3.0 triangles per $\mathring{A}^2$. The number of triangle (also known as the poly count) that form the surface is 184298. Panel C: Simplified surface of PI3K produce with molecular Maya using the low res setting. The resulting poly count was 3516. Panel D: Surface fully simplified using mesh reduction algorithm of Maya. Polygon count of 252.

available, yet none of them can perform polygonal reduction while preserving the exact input shape (Shontz and Nistor, 2013; Wang et al., 2013; Jingsong, 2013). Not only is the topology modified during the procedure, but the extent of this modification depends on the shape itself and its tessellation. Fine details, such as the small bumps that are clearly visible with a high resolution mesh (figure 2.5, panel B) are bound to disappear. Moreover, the accuracy of the results decreases with the increase of complexity of the protein surfaces, such a protein presenting deep pockets or cavities.

Another factor to take into account is the difference between the poly count of the input mesh and the poly count goal. The higher the difference the more unlikely it is to reach the goal in a single iteration and when multiple iterations need to be applied, the probability of topological errors increases. Typical errors are: finding multiple vertices overlapping at the same position, having four or more sided faces, and having non-manifold[16] geometries in general.

In order to minimise these errors and to avoid manual manipulation of the mesh topology, each protein was treated separately and the amount of polygon reduction, as well as the number of algorithm iterations, was decided case by case. While this makes automation of the procedure impossible to achieve, I optimised this step of the workflow to be as standard as possible.

To obtain low resolution surfaces that have a poly count within the threshold imposed by Unity, the following procedure is applied. Once the PDB is available for the protein of interest, it is loaded into Maya using the mMaya toolkit. A first surface is generated with mMaya as described in subsection 2.1.1 using the low resolution setting available from the preset. The generated surface will then undergo a first iteration of the polygon reduction algorithm of Maya. This algorithm has several parameters that can be changed according to specific needs. Firstly, it is possible to set the percentage of polygons that the algorithm will try to reduce.When set too high, this number will never be equal to the final reduction, but will allow for the maximum reduction possible in a single iteration without creating topology errors. For the first iteration of the algorithm the reduction should be set to 80%. It is possible to specify a parameter value that controls the degree to which Maya sacrifices mesh shape accuracy to produce better tessellation. Values

[16] Non-manifold topology polygons have a configuration that cannot be unfolded into a continuous flat piece. Examples of non-manifold geometries are: when two or more faces share a vertex but not an edge, when an edge is shared by three or more faces, and when adjacent faces have opposite normals.

Table 2.1: Examples of polygon reduction

PDB ID	Low-Res Poly Count	Iterations Strength	Final Poly Count
1E82	1596	80% + 20%	250
1G16	1056	80%	212
1J0X	7068	80% + 80% + 10%	254
1YPH	4464	80% + 72%	250
3C13	1768	80% + 30%	248
3CPI	2150	80% + 42%	250
2EED	2832	80% + 42%	250
3CPI	2150	80% + 55%	254
3V03	5596	80% + 70% + 25%	252
821P	1084	80%	216

closer to 0 will preserve the shape, while values closer to 1 will produce a more even tessellation. This value should be set to 0.3, as preserving the shape is more important than the overall tessellation quality, but lower values are likely to cause tessellation errors. In order to improve the accuracy of the final shape it is possible to force the algorithm to attempt to preserve the shape of polygon borders, which are edges that are not shared by other polygons, and keep the vertex positions as much as possible.

Once the first iteration is completed with the above settings, the number of additional iterations and their strength is decided depending on the amount of polygon left on the mesh. For very large proteins (more than 500 amino acids) a second iteration is necessary with a 50–80% reduction, without changing the other parameter values. For proteins composed of 300–400 amino acid, like PI3K or ERK2 as shown in figure 2.6, a second iteration at 30–35% reduction is usually sufficient. Depending on whether the current poly count exceeds the 255 threshold and by how much, other iterations may be needed using smaller percentages ranging from 5 to 25%. Examples of the number of iterations and their strength used for several proteins are reported in table 2.1

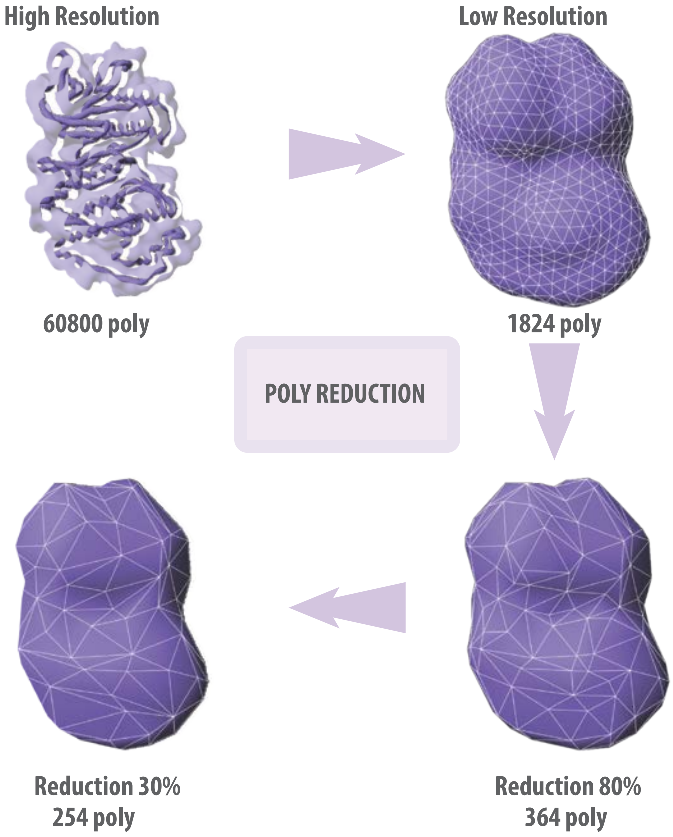


Figure 2.6: Example of the polygon reduction procedure applied to a protein kinase: a high resolution surface was calculated for the protein kinase ERK2 (PDB id 3ERK). The protein consist of 364 amino acids and at high resolution its surface has a poly count of 60800. Using molecular Maya the low resolution surface was generated with a poly count of 18240. Already after the first iteration of the poly reduction algorithm of Maya set at 80% of reduction, the count decreased to 364 triangles. Only a second iteration of the algorithm, at 30% reduction, was necessary to reach 254 triangles.

2.4.2 Binding site definition

In order to simulate protein-protein interactions accurately, binding sites need to be defined on the protein geometries. The information required for this task is the sequence that forms the binding site and with that information it is possible to map the binding site onto the reduced surface. Data about binding partner and binding site location can be found in the literature, but more conveniently in the UniProt database. Its curated version, UniProtKB (Magrane and Consortium, 2011), is a highly reliable and exhaustive source of information about proteins. Both binding partners and sequence annotations can be found directly in the database, together with extensive cross-links to other databases, such as the Protein Data Bank. The cross reference to the Protein Data Bank works both ways, so another way of reaching the sequence annotation is to directly access UniProt from the PDB ID page of the Protein Data Bank.

On the protein geometry, the binding site needs to be represented as part of the surface, but with binding properties that other parts of the protein do not show. Once the low resolution surface is generated with the procedure mentioned above, the connection between the protein's atom locations and the surface is lost. The disruption of this link makes it impossible to retrieve directly the information of the binding site location on the reduced surface. Although not directly possible, the binding site location can be recovered using a new instance of the same protein, and remapping the binding site area calculated from the new instance to the low resolution surface.

Using the molecular visualisation plugin ePMV, a new instance of the same protein can be loaded into Maya and the binding site sequence highlighted on it. When the PDB structure is superposed on the low resolution mesh, like in figure 2.7, the portion of the surface that will create the binding site can be defined by selecting the triangles that lie on top of the highlighted sequence.

As mentioned before, once inside Unity, only binding sites need to show binding abilities and each binding site needs to be distinct from other binding sites, if multiple binding sites are present on the same protein. In order to be able to make this distinction, binding sites need to be defined as a separate entity albeit connected to the rest of the protein. To do this, the binding sites are cut from the rest of the surface and the protein is reassembled as a hierarchy formed by binding sites and the rest of the protein. The selected triangles that define the binding site on the low resolution mesh, can be cut using a Maya script[17] that will create a separate new object from the selection, as shown in figure 2.7 panels C and D. If multiple binding sites are present on a protein, each one needs to be treated separately and each needs to have a separate geometry.

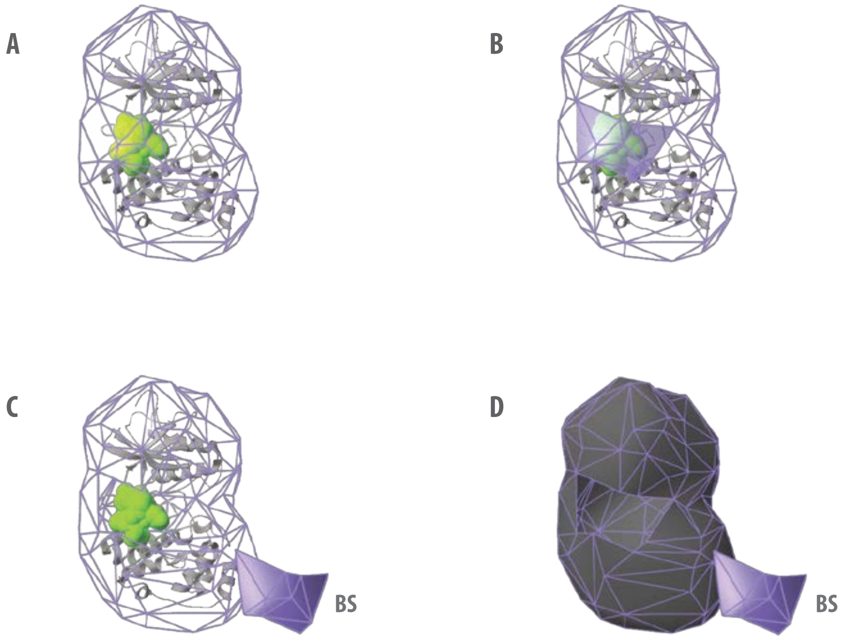


Figure 2.7: Binding site definition. Panel A: using information from UniProt, the sequence is highlighted (green) on the ribbon representation of the protein (grey). Panel B: from the superposed mesh, the triangles that best represent the area highlighted are selected (transparent purple). Panel C: the selected area is cut from the rest of the geometry because they need to acquire binding properties. For the purpose of this representation the binding site was moved lateral to the protein. Panel D: representation of both the binding site and the cut protein surface as solid geometries.

The selection of triangles that will form the binding site is not only dependent on the binding site sequence, but also on the tiling of the surface itself. The smaller the triangles that form the surface, the more accurate the selection will represent the real

[17] The script named *DetachSeparate* was downloaded from Creative Crash at www.creativecrash.com/maya/script/detachseparate-mel

binding site area. Since the number of triangles, the poly count, needs to be kept within the threshold, increasing the resolution is not an option. The best solution for this issue is a compromise between accuracy and the goals of the simulation. When the binding site is known as a well defined sequence, and the simulation aims to give to some insight on the binding kinetics, then a very restricted selection should be made. For cases where the binding site is defined as an entire protein domain, the triangles selection can be made more loosely. Since the protein-protein interaction will only happen in the course of the simulation when two proteins interact with their binding sites, the binding site dimension will influence the likelihood of binding. An in depth discussion about this topic will follow in (subsection 3.2.3).

2.4.3 Reassembly of the protein geometry

I designed the simulator in Unity to be an agent based simulator, in which different sets of rules affect the behaviour of the proteins at different levels. Using an agent based approach, is essential in dealing with the combinatorial explosion caused by the complex formation process. When forming a multi-complex protein assembly, as explained in details in subsection 2.6.1, the possible protein binding combinations grow very quickly with the number of proteins involved, making it impossible to encode for all the combinations prior the simulation. With an agent based approach no encoding of the sequence in which proteins bind to each other is necessary, but only general rules for binding need to be encoded.

As explained in more details in section 2.6 the simulation rules be can categorised as global, local or hyper-local rules, depending on where they are located and what they affect. For example, while a protein needs to move in space as a whole, only the parts of the protein geometry that are defined as binding sites need to posses the ability of binding. In order to be able to make this distinction whilst mantaining a complete geometry that can move all its parts in the same way, a hierarchical representation of protein structures is essential. This hierarchy needs to be assembled as shown in figure 2.8: where each binding site that has been already defined and separated from the main protein geometry is connected to a root that does not contain any geometry. This root will function as the behavioural root of the protein, where local rules will be stored. All the separated binding sites that will hold the hyper-local binding rules, and the main geometry will be defined as a child of this root.

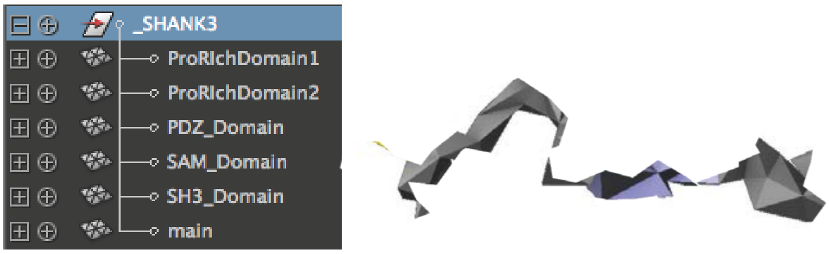


Figure 2.8: Hierarchical representation of a protein: as an example the protein SHANK3 is prepared according to the procedure and the all the five binding sites were separated from the rest of the protein geometry. The protein was reassembled in a hierarchy using an empty root and labelling binding sites correctly. Proline Rich Domain 1 and 2 depicted in blue and dark blue respectively. PDZ domain green, SAM domain yellow and SH3 domain represented in pink. The rest of the protein geometry is labeled as main and coloured in gray.

It should be noted that for very large proteins, especially if composed of more than one subunit, it is better to further divide the geometry, and to reassemble it as a hierarchy. For example, a tetrameric protein can be easily split into the separate subunits allowing for the threshold limit of 255 triangles to be applied on each subunit, resulting in a considerable gain in shape accuracy. The same method is advisable for proteins that are very far from a convex shape. If for example a rod-like protein is bent to assume a C like shape, it is convenient to divide the geometry into more linear pieces as described in subsection 2.6.4. Once imported into Unity, colliders used to calculate collisions by the physics engine are created from the geometry of the protein models themselves. These colliders will be more accurate the more convex the shape is.

2.4.4 Protein orientations

Last to consider, for the protein models preparation is the orientation of the molecules. While cytosolic proteins will diffuse in all three dimensions, membrane bound proteins will diffuse on the membrane's surface therefore in two dimensions. Membrane proteins can be either integral in case for example of transmembrane receptors, or peripheral for which a domain can associate with the membrane or a lipophilic group can be inserted into it. The association with the membrane provides a specific orientation for which membrane bound proteins will diffuse. In order to be able to maintain this orientation while the protein is diffusing, the orientation needs to be defined a priori in the protein geometry.

In 3D modelling it is possible to make a distinction between the global coordinate system and the local one. For global coordinates, often called world coordinate, we refer to the coordinate system of the entire environment. While for local coordinates we refer to the coordinate system of an object inside that environment. In this way each object that populates the world, will have its own coordinate system, as shown in figure 2.9. Moreover, if an object is represented as a hierarchy, with several parts linked to it, like in the case of protein B in figure 2.9, each part will have its own local coordinate system.

This distinction between coordinate systems facilitates calculation of relative movements of objects and also allows for a high degree of flexibility during the design of the object. I used this feature to assign a specific local orientation of the membrane bound proteins, that will define the local plane in which these proteins will then diffuse. I decided to reorient the protein in such a way that the local y axis of the protein domain that is in contact with the membrane is aligned with the global Y axis. Doing so, as explained in (subsection 2.6.7), defines the 2D plane in which the membrane proteins will diffuse as their own local x,z plane. Now that the protein models are fully prepared they are ready to be imported into Unity.

2.5 Simulation container

Depending on the type of simulation that one wants to execute, a simulation container needs to be designed or imported when available. A simulation container can be either a simple box or a sphere to a geometry obtained from reconstruction of microscopic imaging. The container has the function of defining the simulation environment boundaries, its volume, and whether membrane proteins are present in the simulation to provide the membrane representation. If a simple box or a sphere is sufficient for the purpose of the simulation, as in cases where the geometry of the environment is not as important as its volume, then the container can be created directly in Maya. In cases where a more detailed representation is needed, geometries of cell compartments or entire cells

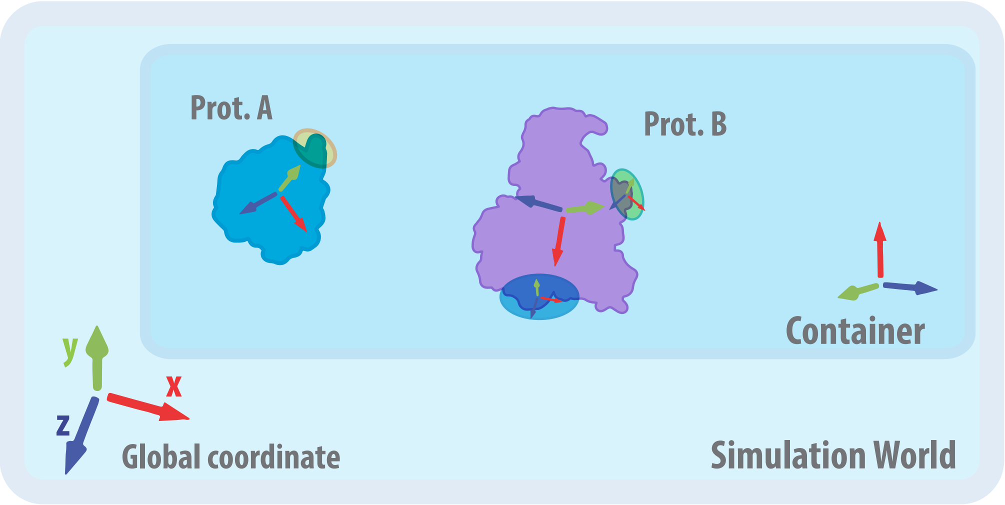


Figure 2.9: Global and Local Coordinate Systems: the global coordinate system is defined as the Cartesian coordinate system of the word. This means that every object that is placed inside the word, has its own local coordinate system. As shown for protein B even separate parts inside an object have their own local coordinate systems. Absolute positions are calculated from the global coordinate system while relative position, like for example the position of protein A, can be calculated relative to the container.

reconstructed from experiments can be found in the literature or from specific databases[18]. Once the geometry is created or obtained, Unity needs to define the volume inside the geometry as the one in which the simulation will occur. This can be achieved in Maya by simply inverting the normals of the geometry. This operation defines the volume inside the geometry as empty, allowing it to be populated with the desired proteins. The boundaries of the simulation will then be automatically defined as the internal surface of the container geometry, which will also translate to the representation of a membrane, depending on the simulation.

For containers, the polygonal count threshold imposed for proteins does not apply since, unlike proteins, the container will not move during the course of the simulation. Not having a defined limit to the resolution of the container mesh allows the use of potentially any desired geometry, making this method a highly flexible.

2.6 Simulating protein assembly with the developed Unity extension T.A.R.S.I.D.

The second part of the pipeline focuses on importing and simulating in Unity the prepared proteins in the simulator container, adding all the necessary behavioural rules and then starting the simulation. Those processes were carried out using the Unity extension T.A.R.S.I.D. which I specifically developed for this work.

There are several advantages in using Unity as a simulator. Firstly, since Unity is a game engine and in every game calculating collisions is very important, Unity uses a physics engine that has been developed for fast and accurate collision detections, PhysX.

Another useful feature of Unity, is its modular design, which made it easier to develop these simulations as agent based. As it will be explained in subsection 2.6.1, this was crucial in the development of an efficient binding strategy. Another important feature of Unity is the hierarchical approach to rule definitions. For example, properties can be defined at a global level and then be extended or overridden at a local level. This allows for the use of global rules, like rules that define how complexes are made, which affect all the proteins in the system, and local rules, like binding rules, which are specific to a single type of protein and affect only a specific binding site.

[18] http://ccdb.ucsd.edu/index.shtm

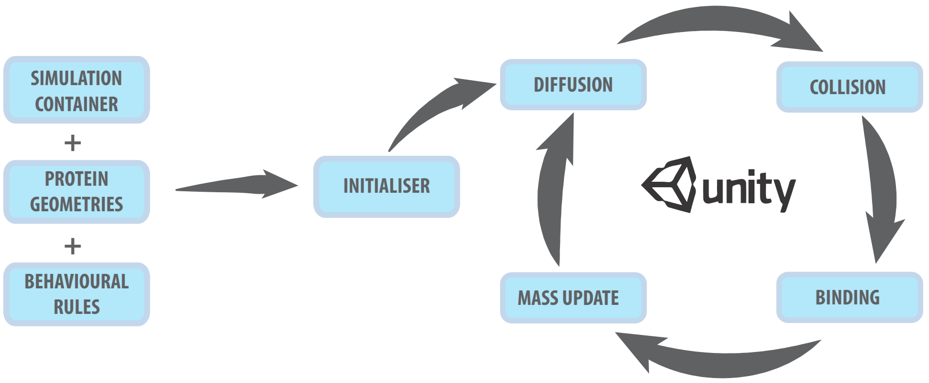


Figure 2.10: T.A.R.S.I.D workflow overview: the simulator container and the protein geometries, prepared as explained in the Maya workflow, are imported into Unity and the desired behavioural rules are added. A first step called Initialiser is loaded in which all the protein types are spawned with random positions and orientations. The Initialiser will stop after reaching the desired stoichiometry for each protein type. The output file with position and rotation is fed to the Simulator which will start the cycle of mass driven diffusion, collision checking, complex formation when two entities collide with the correct binding sites, and mass updates as the sum of the mass of the two entities that form new complexes. The updated mass is fed into the diffusion rule and the cycle starts again.

A general overview of the workflow of this Unity extension, T.A.R.S.I.D. which stand for Translation Association and Rotation of Solid bodies in 2 and 3D, is represented in figure 2.10 . Once the simulator container and all the protein geometries are imported the behavioural rules are added. Global rules are added to the simulation environment, while local rules are added to single proteins, with hyper-local rules added to specific binding sites. The first step of the simulation, the Initialiser, aims to insert the all required proteins with the correct stoichiometry inside the container. When each protein type is present in the system in the desired quantity, the simulation can start. At each time step the proteins diffuse inside the simulation container while collision events are detected. In case a collision occurs between binding sites, the possibility of complex formation is checked. When a complex is formed, the mass of the complex, defined as the sum of the two colliding proteins, will be updated in the module that drives the diffusion. At the next time step the complex will diffuse according to the new mass, and the cycle will start again.

In the following sessions I will explain in detail the rules used in the Initialiser and in the Simulator and their functioning.

2.6.1 Agent-based modelling

Agent-based modelling is largely and widely used in different scientific domains (Niazi and Hussain, 2011). The definition of an agent can vary from domain to domain, but this modelling approach presents some common characteristics. Firstly, the behaviour of an agent-based model is defined by a set of rules that applies to its agents. This implies that each agent decides what to do based on its own rules. Secondly, through only the use of simple local rules on agents, it is possible to model complex behaviour. Lastly using this approach it is possible to make stochastic single particle simulations in which the behaviour of each agent can be tracked individually.

One of the problems that arises from single particle modelling of complex systems is dealing with the combinatorial explosion of all the possible entities in the system. In the case of multi-protein complex formation, all the possible intermediate complexes should be considered as different entities and therefore encoded in the simulation separately, causing the number of different species in a model to grow rapidly: a phenomenon known as combinatorial explosion.
An example of combinatorial explosion can be made considering 3 different molecules that bind to each other: A, B, and C. Firstly, the behaviour of the 3 single molecules needs to be defined and since all the events are simulated in time is necessary to taken into accounts all the possible intermediate states. The intermediates in this case are 6 different types of dimers: AB, BA, AC, CA, BC and CB. Lastly, all the possible trimers need to taken into account: ABC, ACB, BAC, BCA, CAB, and CBA. Resulting in a total number of 15 different entities.

This estimate is the upper-limit estimate thought as the symmetry of a complex is not taken into account; for example the dimer AB is equivalent to the dimer BA for the purposes of this type of simulation, but in this estimate are still considered as different. As the multi-complex grows accounting for symmetry becomes more complex. Already in the case of the trimer ABC, where A is bound to B, and B is bound to C, it is equal to the complex CBA but not equal to ACB, where A is bound to C and not B.

Increasing the number of protomers in the simulation will also increase the number of different behaviours one should encode, and this number upper-limit (N) can be determined by the following expression:

$$N = \sum_{k=1}^{m} (m!/(m-k)!) \tag{2.1}$$

Where m is the number of starting monomers in the simulation, and k is the polymerisation state, meaning that for a dimer $k = 2$, for a trimer $k = 3$, and so on. With 5 starting monomers, instead of 3, there are a maximum number of 325 behaviours to take into account, whereas with 6 starting monomers this number will increase to 1956. It is clear that a classical approach is not feasible when dealing with systems, like multi-proteins complex formation, that leads to a combinatorial explosion. For this reason I decided to design the simulator taking advantage of the agent-based modelling principles, focusing on the monomeric perspective.

Thanks to the modularity of Unity, I could encode different rules at different levels, as shown in figure 2.11. There are three different types of rules in used in T.A.R.S.I.D. : Global, local and hyper-local. *Global rules* are rules that apply to all the proteins, the agents in the system; examples are: the rule that defines how to form a complex, and the rules that keep track of the global simulation state and write outputs. Global rules are located in the simulation environment, where each agent can easily access them. *Local rules* are rules that apply directly to the entire protein, like the rule for diffusion, the rule that defines the protein state and the rule that manages the mass of the protein. Local rules are applied to each protein on their behavioural root, which holds the whole protein geometry. *Hyper-local rules* are located to the binding sites of proteins. For each protein type, binding rules are defined for each binding site present on the protein. This allows each binding site to act independently from one another, even on the same protein. This strategy is very important when a protein can bind two or more different proteins. Each protein type will start the simulation with the same local and hyper-local rules setting, but during the course of the simulation, each protein will modify its settings and consequently its behaviour according to its specific history.

Underneath these layers of rules, lies an extra layer: the Unity physics engine, PhysX. The main role of PhysX in T.A.R.S.I.D. is to detect all the collisions in the system and calculate their exit trajectories.

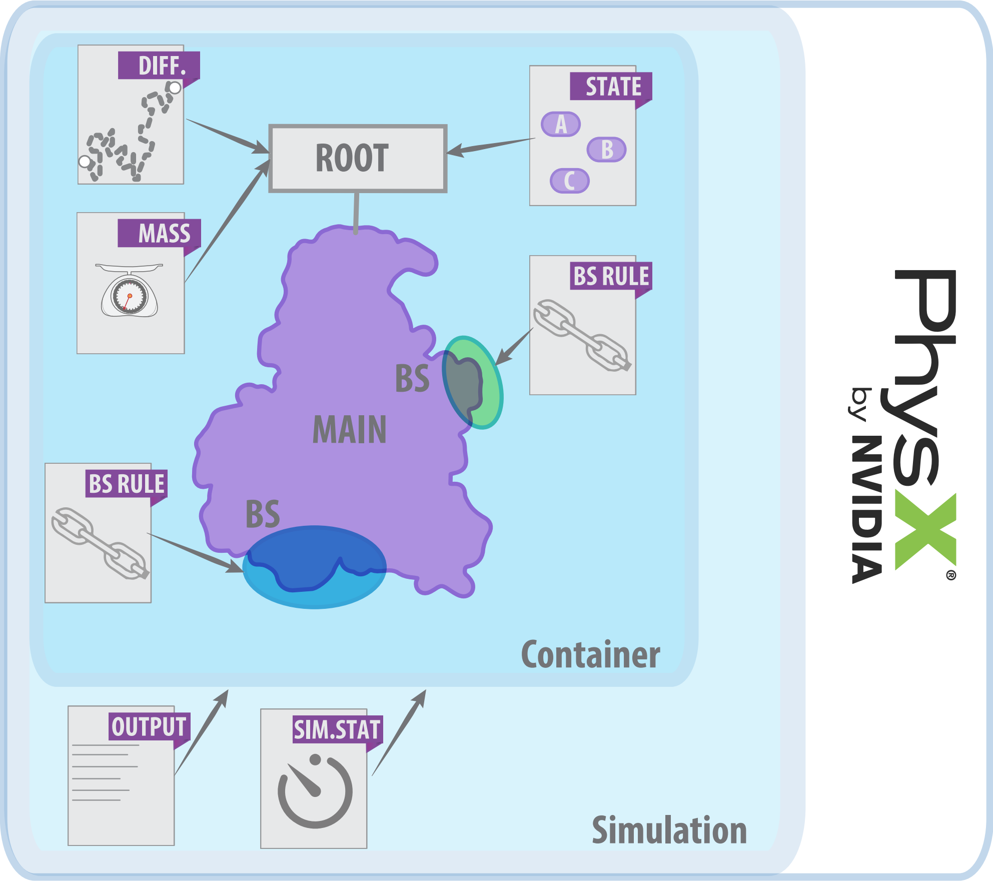


Figure 2.11: Local and global rules: schematic representation of the different rules in the simulation. The diffusion module, mass manager, and protein state are local rules, since they need to affect the whole protein, and are located on the protein behavioural root. The hyper-local binding rules are located directly on binding sites, allowing each binding site to act independently from one another, even if on the same protein. On a higher level, global rules are in place to keep track of the simulation status and to write outputs. Behind everything, there is Unity physics engine PhysX, mainly used for collision detection.

2.6.2 Initialiser: spawning and starting positions

Due to the spatial nature of these simulations, the starting position of proteins will influence the time course of the simulations. Considering, for example, an extreme case with only two proteins diffusing in a box container: if at time zero both of the proteins are in one corner of the box, they will have a higher probability of colliding in a relative short period of time, while if the starting positions are in different corners of the box, the probability of collision within the same time frame will drop. Since 3D diffusion is a distant-independent process, given an infinite time, all the proteins will have exactly the same probability of colliding with each other; the only factor that the starting position influences, is the collision probability at the beginning of the simulation. A way to minimise this influence is to randomise the starting positions. As proteins in these simulations are represented as rigid bodies with a finite volume, when assigning random positions the volume and shape of each protein need to be considered in order to avoid interpenetrations. At the start of the simulation, the proteins must be present in the system with a random position and random rotation without overlapping with each other or penetrating the container. Calculating such a starting point, leads to non trivial and computationally expensive iterations of collisions checking and corrective replacements of the molecules.

In order to produce random starting positions and random orientations, with a lower computational cost, I created a first step for the simulation, called the Initialiser, in which proteins are instantiated with the desired stoichiometry. The Initialiser is a short simulation that uses the same simulation container, but a much simpler set of rules than the one used during simulations. Each protein type will have an instantiator module located inside the container, from which the proteins will be spawned one by one, until reaching the desired stoichiometry. As Unity is a game engine, an efficient way to encode the Initialiser into the simulation was to encode it as a game level.

Due to the different behaviour of membrane proteins, two different local rules were created for cytosolic and membrane spawning. Cytosolic proteins after being spawned from the instantiator module, will start to diffuse in 3D using a simplified diffusion rule that only uses a vector of constant length but random direction. Cytosolic proteins will then diffuse in the container and when they will collide, with either another protein or the container surface, PhysX will calculate the exit trajectories. This simplified diffusion allows the proteins to randomly distribute inside the container assuming a random rotation. The diffusion process will continue until all the proteins are spawned inside the container. Binding rules and protein state rules are not present in the Initialiser since the proteins should not be allowed to form complexes at this stage.

Instantiation of membrane bound proteins is more complex, as the membrane, the surface of the container, can be of any shape and dimension, from a simple sphere to a reconstruction of a dendritic spine from EM images. However, high flexibility in the geometry of the container results in an automated procedure to detect the membrane, one that is not influenced by the concave or convex nature of the surface itself. Moreover, the membrane bound proteins are in contact with a membrane with a specific orientation, which also needs to be calculated in respect to the surface curvature.

The accurate implementation of this process, as described in subsection 2.6.7, is time consuming and not necessary for this first step. For this reason, in the Initialiser I used a simplified version of the diffusion rule for membrane bound proteins. As explained in more details in (subsection 2.6.7) my solution for positioning and moving membrane proteins is based on the ray-casting feature in Unity. Ray-casting is the process where an invisible ray is cast from an object toward a certain direction for a certain distance. Once an object is hit by the ray, information about the position where the ray hit and the normal of the part of surface hit, can be retrieved. From the spawning position a ray is cast in a random direction for an infinite length. As soon as the ray hits the container surface it retrieves the information of the point on the surface that has been hit, and that position is used to place the protein on the membrane. After positioning the protein on the membrane a rotation is applied in order to align the protein y axis to the normal of the surface. After being spawned, membrane proteins will diffuse on the membrane. This is the same principle explained in subsection 2.6.7 but using a 2D vector of constant length and random direction.

Until all the protein types with the desired stoichiometry are present in the container, the instantiator module will keep spawning proteins, and the proteins already present in the system will keep diffusing.

When the stoichiometry is reached, the Initialiser writes the output files. Output text files are generated for each protein type, containing the position and rotation of each protein spawned in the scene. These files are then read by the Simulation, and used as starting positions.

2.6.3 Simulator algorithm workflow

Once all the data of starting positions are stored into the output data files of the Initialiser, the Simulation level can be loaded. The Simulation is focused on two major processes: diffusion and protein-protein interaction. As figure 2.12 shows, the Simulation algorithm takes as first input the mass of the protein. This is a very important step because upon complex formation, the diffusion properties of the complex must be recalculated. Approximation could be done for a binary complex, such as assuming that the diffusion coefficient of the complex can be approximated to the smaller diffusion coefficient between the two proteins. Another possibility would be to specify a different diffusion coefficient for each complex but as explained in subsection 2.6.1 due to combinatorial explosion the number of intermediate states to define is too high. For this reason, I developed the diffusion as mass driven, in which diffusion coefficients are estimated by the sum of the proteins masses, independently from the combination of proteins forming the complex.

As discussed before, a main distinction is done for cytosolic proteins and for membrane proteins in terms of diffusion. At each time step, cytosolic proteins diffuse in 3D with a 3D translation and 3D rotation, while for membrane proteins translation happens on the membrane surface in 2D, and rotation is only allowed on a single axis, i.e. one dimension.

Collisions occurring in the simulations can be classified in two types: one in which two binding sites are involved, and the other where the collision involves everything else that is not labeled as a binding site. For the latter, the internal physics engine of Unity PhysX, will calculate the exit trajectories. When two binding sites are involved in a collision, the local binding rules, located on binding sites, check if both of the binding sites are available for binding. If so, the rules check if the collision is occurring with the correct binding partner.

The kinetics of in-solution reactions can be either limited by the diffusion of the reactant (Smoluchowski, 1916) or by the association constant of the reaction. In cases where non-diffusion limited associations need to specified, an association probability can be input. The probability is checked against a random number in order to lower the overall association speed by the desired amount stochastically. Upon binding the mass of the complex is updated, via the mass manager, as the sum of the two protein masses and the cycle continues with a new diffusion step.

In general proteins will diffuse independently from each other, following the translation and rotation given by their local diffusion module. Where a protein enters into contact with the binding site area of another protein, it will trigger a local reaction between the two proteins involved in the collision. This event triggers a reaction that checks whether the collision happened between two free binding sites, and whether the two binding sites are the correct pair according to a local binding rule. If all the prerequisite are fulfilled then a complex is formed according to a global rule, dependent on both proteins states.

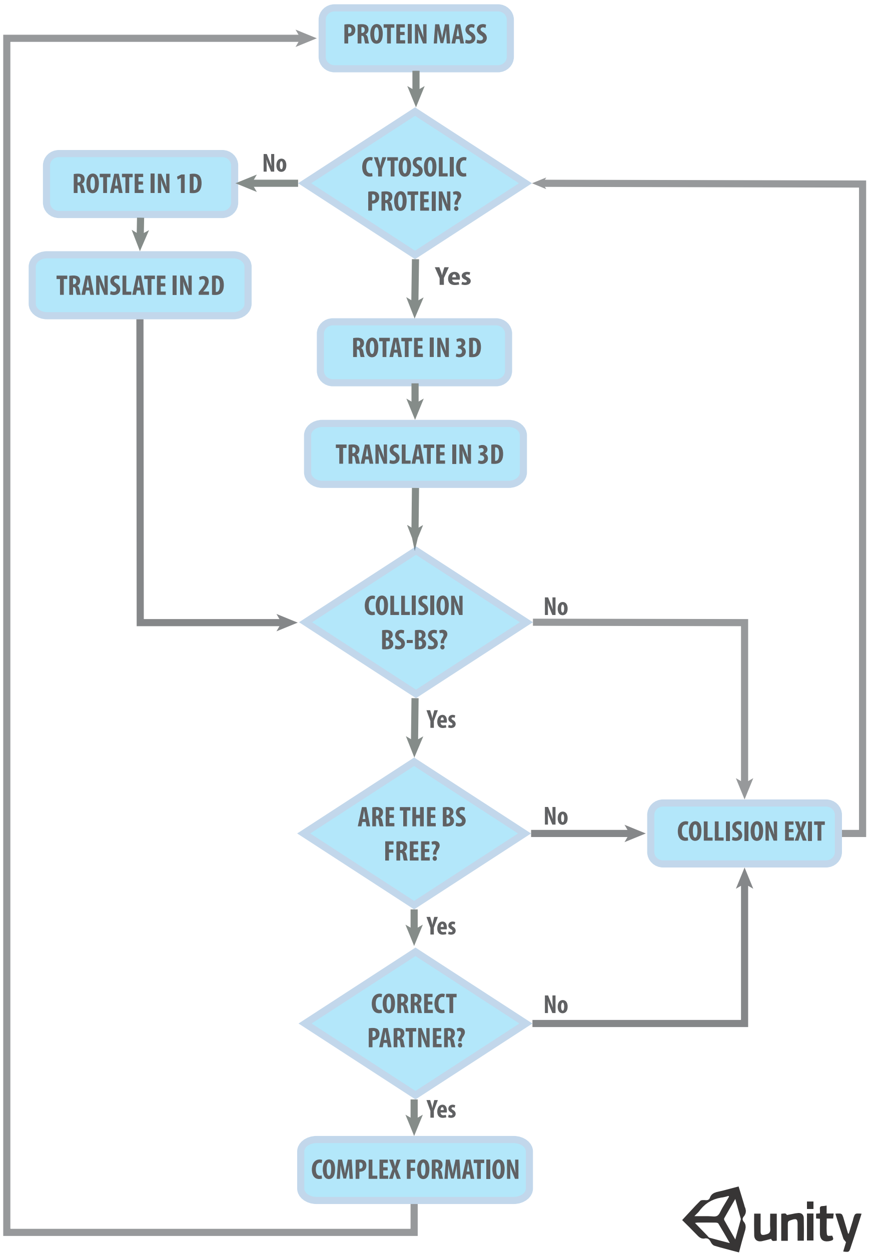


Figure 2.12: Simulation algorithm workflow: the algorithm takes the mass as input for calculating the diffusion. If a collision happens the algorithm checks whether a binding site is involved and if so, whether a complex can be formed. Upon complex formation, the mass of the complex is updated and a new diffusion coefficient calculated.

2.6.4 Collision detection and collision exit trajectories

As mentioned before the internal physics engine of Unity, PhysX, takes care of collisions detection. The geometries need to be interpreted by PhysX in order to compute collisions, ray-casting and movements, and the interpretation requires colliders, triggers and rigid-body definitions.

Colliders define the boundaries of an object to the physics engine, meaning that the shape and dimension of a collider is what the physics engine reads as the shape and dimension of the object. There are several types of collider in Unity that differ for geometries, properties and limitations. The most basic colliders are the *primitive colliders*, colliders that possess a primitive geometry such as a sphere or a cube. Due to their simple geometry, computing collisions using primitive colliders is fast, but using them on complex geometries makes collision detection not accurate.
In Unity there are two colliders that use directly the geometry of the object the collider is applied to: *mesh colliders* and *convex mesh colliders*. While the physics engine is able to detect collisions between convex mesh colliders, it is not able to do so for mesh colliders. This implies that mesh colliders cannot be used for moving objects in the simulation, since the collisions between them will not be detected. Convex mesh colliders have a 255 triangle limit in the polygonal count of the input mesh, whilst mesh colliders do not. Convex mesh colliders, as the name suggests, also need to consider the input mesh as convex. Convex objects are highly suitable for use in collision detections, as calculations for deciding if two objects are colliding are much simpler compared to concave objects, making the simulation less computationally expensive. For proteins that possess a complex geometry that cannot be accurately described with a sphere or a cube, and that need to move in the simulation environment and interact with other proteins, the most suitable collider type is a convex mesh collider.

The generation of a convex mesh collider from the imported mesh geometry is done automatically in Unity. As shown in figure 2.13, for real convex geometry like the letter I, the resulting mesh collider has the exact same shape as the input mesh. For non-convex objects, like the letter C, the resulting convex mesh collider is not accurate enough. The characteristic opening of the letter C has been closed, in order to create a convex geometry (grey area in figure 2.13, panel B). If this collider is used in a simulation, no other object will be able to make contact with the surface that defines that opening.

Rod-like proteins could share the same problem when a convex mesh collider is generated. To avoid the loss of important surface contacts, I propose the simple solution of dividing the protein into separate parts. Using the same principle when defining a binding site, the surface of the protein can be divided in several parts and then reassembled into a hierarchy. For each part, Unity will compute a convex mesh collider, and since the concave curvature of each part is much lower than the curvature for the entire geometry, the final result is much more accurate. An example of this principle is shown in figure 2.13, where the letter C has been divided into 4 different parts, labeled with different colours. The resulted convex mesh collider now shows the opening of the letter C, meaning that sufficiently small objects will be able to access that area.

Defining a collider for the simulation container is quite straightforward. Since the container is not required to move but only to sense collisions with other proteins, the collider of choice was a mesh collider. For mesh colliders, concave or convex shape is not important, but in this case, the limitation is that a mesh collider cannot collide with another mesh collider, for which reason mesh colliders cannot be used for proteins.
For its characteristics the mesh collider is the ideal collider for the container: even a very complex shape derived from 3D reconstruction of microscopy images, can be used without loss of detail. Once the geometry, created in Maya, is imported into Unity a *mesh collider* can be automatically generated.

Another category of colliders are *triggers*. A trigger is a special type of collider that has a unique set of messages that are sent upon collision with a trigger. These event driven messages are sent when a collider gets into contact with the trigger, when it exits and for the duration of the contact. Triggers are used in this framework to define binding sites. When a collider enters into contact with a binding site, the trigger starts a process to check whether a binding reaction can occur or not, according to the binding rules of the binding site itself.

Colliders define the boundaries of an object for collision detection, but in order for an object to be influenced by the physics engine, it also needs to be defined as a rigid-body. With a rigid-body definition, an object trajectory can be influenced upon collision, which is foundamental in these simulations. Although the trajectory is normally encoded by the translation part of the diffusion rule (subsection 2.6.5), upon collision the physics engine overrides the translation module and calculates what is going to happen to the two molecules that are colliding. If no binding results from a collision, a collision exit

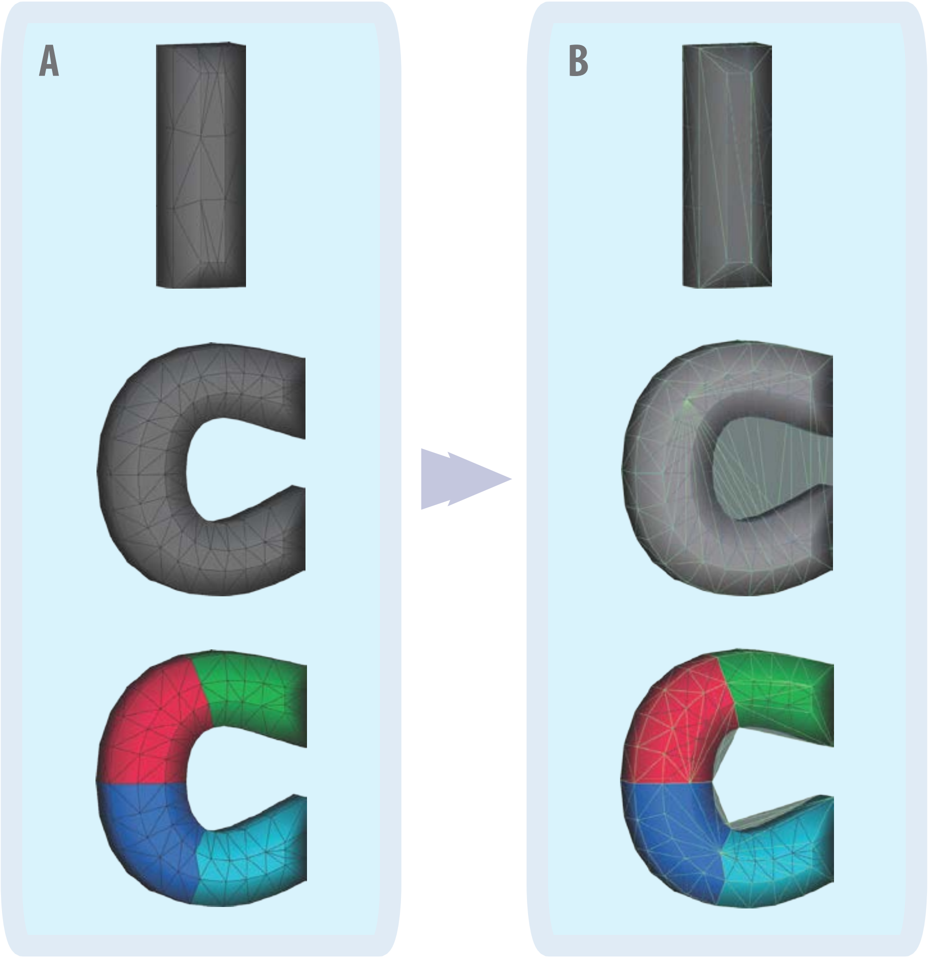


Figure 2.13: Mesh Collider in Unity. Panel A show an example of a convex object, the letter I, and an example of a concave object, the letter C. The letter C is presented twice: the grey C is a single mesh, while the multicolour C has been divided into 4 separate surfaces, labeled with different colours, and reorganised into a hierarchy. Panel B: the meshes from panel A are imported into Unity where a convex mesh collider is automatically generated for each mesh. The tessellation of the convex mesh collider is depicted in green.

trajectory needs to be calculated. The physics engine takes into account the direction the velocity and mass of the proteins that are colliding, and calculates the new trajectory vectors as result of an elastic collision.

2.6.5 Diffusion rules: translation

For both cytosolic and membrane bound proteins, the first step towards translation is the calculation of the translational diffusion coefficient. Regardless of whether the object is composed of one or more proteins, the updated value of the mass is taken from the mass manager and used in the calculation of the translational diffusion coefficient with equation 2.2 derived from the work of Young and Carroad (Young and Carroad, 1980).

$$D_t = 8.34 * 10^{-8}(T/\eta M^{1/3}) \tag{2.2}$$

In which T is the absolute temperature of the system, η is the viscosity of the medium expressed in cPa, and M is the mass of the protein in Da. This equation was derived from the Stokes-Einstein equation of diffusion (Einstein, 1905) that in its general form is:

$$D_t = kT/6\pi\eta r \tag{2.3}$$

Where the translational diffusion coefficient is calculated from: k the Boltzmann's constant, T the absolute temperature of the system, η the viscosity of the medium and r the radius of the molecule.

Assuming that proteins share a medium partial specific volume of 0.73 cm^3/g (Young and Carroad, 1980) it is possible to derive the relationship between diffusion coefficient and mass as in 2.2. This equation is used to calculate the diffusion coefficient of cytosolic proteins. A study on diffusion of membrane bound proteins (McCloskey and Poo, 1986) shows that on average membrane bound proteins exhibit a diffusion coefficient that is 1000 fold lower than their cytosolic counterparts. Based on this study, I calculate the diffusion coefficient for membrane bound proteins using the equation 2.2 and afterword diving it by 1000.

As explained in subsection 1.4.5, the probability distribution of a particle position on a single axis, is a Gaussian distribution with variance $\sigma_x^2 = 2D_t t$, equation 1.9. It is possible to calculate, for each time step, an Euclidian vector that has for components random values taken from a normal distribution with variance $2D_t$. For cytosolic proteins 3 different values from that normal distribution are calculated for each time step, for translating them in 3D space, while for membrane bound proteins, only two values are needed, as the translation will happen in 2D space. Since all the membrane bound proteins are prepared with the membrane binding domain aligned on the y axis, the 2D plane of choice for translation is the protein x,z plane. This plane is a local plane, meaning that each protein has a different plane orientation in respect to the global axes of the simulation.

2.6.6 Diffusion rules: rotation

Proteins do not have a sense of direction therefore allowing them to rotate while diffusing is highly important. While for simulators that model proteins as dimensionless points or as regular spheres it is not necessary, as soon as asymmetry is introduced, rotation has an important role. Spheres do not require explicit rotation due to their perfect symmetry but as soon as a part of the sphere surface is labeled with different characteristics, such as defining a binding site, the sphere looses its symmetry, and different rotations will result in distinguishable orientations.

Following the same principle applied to the translational diffusion coefficient, from the Stokes-Einstein equation for rotational diffusion

$$D_r = kT/8\pi\eta r^3 \tag{2.4}$$

Where k is the Boltzmann's constant, T the absolute temperature of the system, η the viscosity of the medium and r the radius of the molecule, it is possible to derive the relationship between the rotational diffusion coefficient and the mass of a protein, as in equation 2.5.

$$D_r = 27.89(T/\eta M) \tag{2.5}$$

In which T is the absolute temperature of the system, η is the viscosity of the medium expressed in cPa, and M is the mass of the protein in Da. As well as the squared variance for translation the one for rotational diffusion on a single axis is the following:

$$\sigma^2 = 2D_r t \tag{2.6}$$

Like for the translation, this was implemented as a rotational vector that has for components random values taken from a normal distribution with variance $2D_r$. While for cytosolic proteins the rotation is calculated on each axis, for membrane bound proteins, the rotation is only allowed on the protein y axis, the axis perpendicular to the membrane, so a uni-dimensional vector is applied.

In order to better understand the importance of rotation in cases of asymmetrical objects, I made a simulation in T.A.R.S.I.D. for the simple case of two identical spheres A and B, each with a 2.5 Åradius. Both spheres have one quarter of their surface defined as a binding site. If 200 of such spheres were simulated using a mass of 11000 Da to calculate diffusion coefficients, in only 216.45 msec 95% of the proteins will be in a bound state, as dimers (as shown in figure 2.14, blue line).

If the rotation module is removed from the simulation, but keeping the ability of the physics engine to calculate rotations for collision exit trajectories, the time needed to reach 95% of the proteins in bound state will increase to 339.73 msec (as shown in figure 2.14red line, labeled as Collisional Rotation). In cases where rotation is not allowed at all, the orientation acquired by the spheres during spawning will remain constant for the entire the course of the simulation. This lowers the probability that the two spheres will collide with the two binding sites and increases the time at which the simulation reaches the 95% of proteins in a dimeric form, to 640.23 msec. Switching from collisional rotation to no rotation allowed, almost double the time needed for the 95% of spheres to be in an efficient orientation for binding, while going from explicit rotation to no rotation allowed, tripled this time. The time needed for a certain number of proteins to be in a bound state is directly linked to the binding constant of that binding reaction. Rotation is clearly playing an important role in accurately simulating binding kinetics as these simple simulations show.

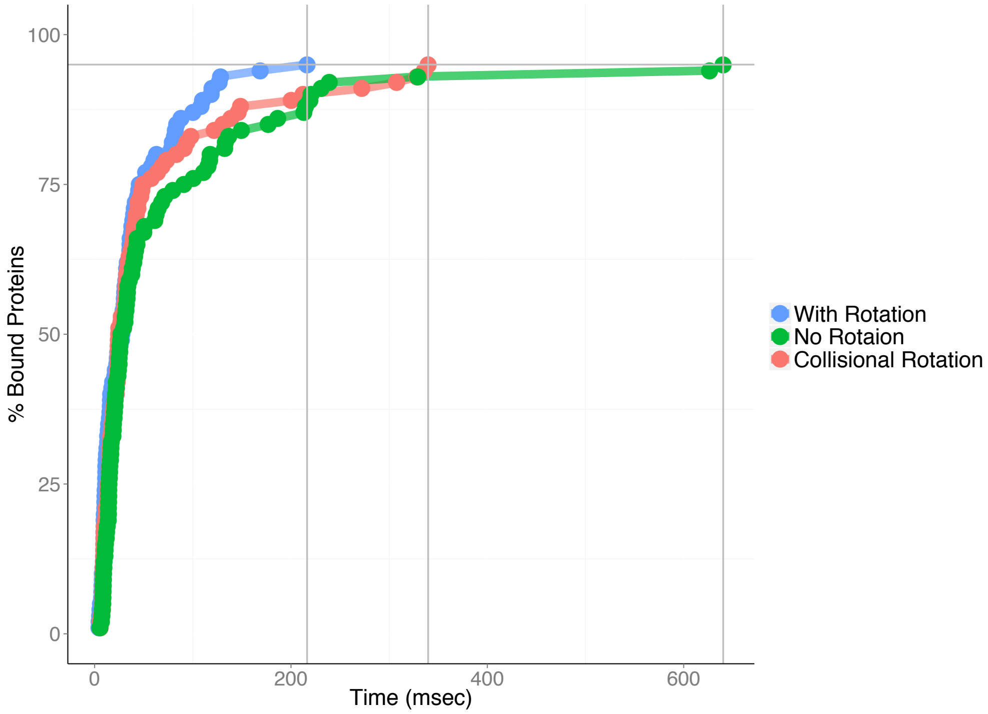


Figure 2.14: Influence of Rotation on Binding: simulations of 2 pools, A and B, of 100 spheres with radius 2.5 Å and 1/4 of the surface defined as a binding site. Binding events can only occur between the binding site of A and the binding site of B, with a 1:1 stoichiometry. The diffusion coefficient was calculated using a mass of 11000 Da. Three different simulation setups were used: all the spheres were allowed to rotate while diffusing (blue line, labeled With Rotation); the spheres were allowed to rotate after collisions events (red line, No Rotation); and lastly no rotation was allowed (green line, No Rotation). Horizontal grey line signifies that 95% of the proteins are in a bound state, while the vertical lines mark the time at which the simulations reached that state.

2.6.7 Difference between diffusion on membrane and in cytosol

At each time step, the position of all the proteins in the system is recorded and rotational vectors and translational vectors are calculated. These vectors are used to update the rotation and position to each molecule by adding a step length, represented by the translational vector added to the previous position, and adding a new step rotation to the previous one.

While for cytosolic proteins this procedure is enough to assure a Brownian diffusion, it is not the case for membrane bound proteins. As pointed out in the description of the Initialiser (subsection 2.6.2), the shape of the container can be completely arbitrary, meaning that its surface does not have to lie on a single plane nor have a fixed curvature. With this type of surface it is not possible to calculate a priori a translational 2D vector that will keep the protein on the membrane. Other simulations, such as Smoldyn, avoid this problem by diffusing membrane bound proteins in 3D and then projecting them back onto the membrane. My solution to this problem was to introduce a checkpoint to determine whether the membrane protein is still in contact with the membrane or not after the diffusion step. In case it is not, the protein is repositioned again on the membrane using a force vector with module proportional to the distance from the membrane, and of direction equal to the normal of the membrane surface.

Figure 2.15 shows a graphical representation of the algorithm. Firstly the algorithm checks whether the protein is correctly in contact with the membrane. To do so, it checks when the protein enters in collision with the membrane flagging it as bound to the membrane, and when the protein exits the collision, flagging it as not bound to the membrane. When a protein is flagged as bound to the membrane it is normally translated in 2D, on the protein x,z plane and rotated on the protein y axis, with the same principle explained in subsection 2.6.5 and subsection 2.6.6 respectively.

In cases where a protein is diffusing in an area where the membrane is changing its curvature, as shown in figure 2.15, even very small changes in direction cause the protein to detach from the membrane. When this happens, the algorithm flags the protein as not bound to the membrane and it starts the procedure to reposition the protein on the membrane. As with the instantiation of proteins on the membrane in the Initialiser, this procedure uses the ray-casting feature of Unity.

A ray is cast from the protein flagged as not bound to the membrane, in its local y

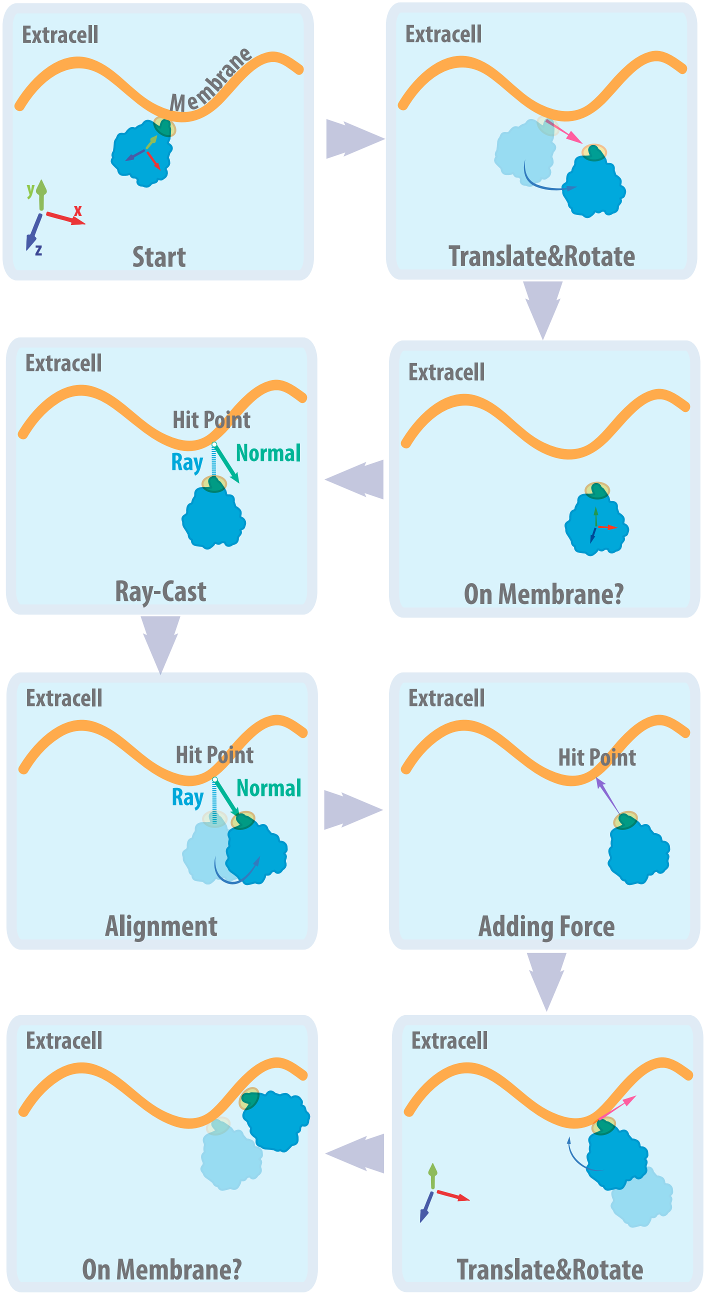


Figure 2.15: Diffusion on membrane workflow: After a diffusion step, the algorithm checks if the protein in the new position is still attached to the membrane. If not using ray-casting a point on the membrane is found, and with it the distance and the rotation that the protein needs to undergo are calculated. A force proportional to the distance from the membrane is applied after the protein is aligned to be perpendicular to the membrane surface. At this point the protein will take the next diffusion step and the next cycle will start again with a new check for membrane contact.

direction. When the ray hits the membrane, the normal of the hit face and the distance between the protein and the hit point are recorded. The protein y axis is then aligned to the face normal. This rotation sets the protein in the correct orientation, perpendicular to the membrane, but the protein is still not in direct contact with the membrane itself. The protein is then pushed towards the membrane with a force proportional to the distance of the hit point, in the protein's y direction. Using a force instead of a brute displacement of the protein on the hit point coordinates, is desirable. Since proteins are defined as rigid bodies that cannot interpenetrate with each other, using a brute displacement can lead to errors in collision detections and simulation instabilities. These errors are mostly floating-point errors on the position of vertices of a recomputed mesh after movement, leading to small interpenetration of the protein in the membrane. When these small interpenetrations are identified by the collision detection algorithm of Unity, the proteins involved are quickly separated from the membrane, as a high velocity collision with the membrane occurred. Using force instead of brute displacement, completely avoids this issues and lead to more uniform results.

As soon as the protein collides with the membrane, it is flagged again as bound to the membrane and is normally translated in 2D and rotated on the protein's y axis. A new cycle starts with a new check to determine whether the protein is in contact with the membrane or not.

2.6.8 Binding rules: How to create a complex

For complex formation 3 different rules are involved. The hyper-local binding rule on the binding site of a protein, the binding manager on the protein root and the global rules of binding that are located in the simulation. The binding rule located on the binding site of a protein defines the binding partner for that specific binding site, and its stoichiometry. When two binding sites collide and both sites are available for binding, a call is made to the global binding rules (figure 2.16). These global rules, which are the same for all the proteins in the simulation are used to define how the complex will be formed in the simulation environment and to update the mass of the new complex.

In spatial simulators such as CDS (Byrne et al., 2010), that performs rigid bodies simulations, a new entity is created when a complex is formed. The single entities, in this case the two proteins, are physically deleted from the simulation environment, and a new entity for the complex is created. Depending on the level of detail of the simulation, the new complex could simply be a sphere with a radius equal to the sum of the two protein radii, or maintain a more complex shape. In both cases, the simulator needs to introduce this new object into the environment, deciding its position and for more complex shapes its orientation. The complex needs to be positioned in the area previously occupied by the two proteins and rotated in a way that will not create collisions or interpenetrations with other entities in the surrounding area. In this process of instantiation of the new complex entity it is usually iterated for a finite number of times and, if no solution is found, the two single proteins will be re-instanciated and the binding will no longer occur.

In order to avoid these calculations, that become increasingly difficult the more the complex grows, I decided to not create a new entity upon complex formations, but to parent the two protein geometries involved in the binding process. Parenting two entities means that one of the two proteins will become a child in the hierarchy of the other protein. Using parenting gives the advantage that the proteins are not deleted from the simulation nor moved. Upon complex formation, the two proteins are locked together with the exact position and orientation, but they never cede to exist as single entities. So a complex is a protein that has inside its hierarchy another protein, and like every other protein, its root has the crucial function of keeping all the rules that affect the behaviour of the entire protein, or in this case of the complex.

To define which of the two proteins will become the parent, and so the new behavioural

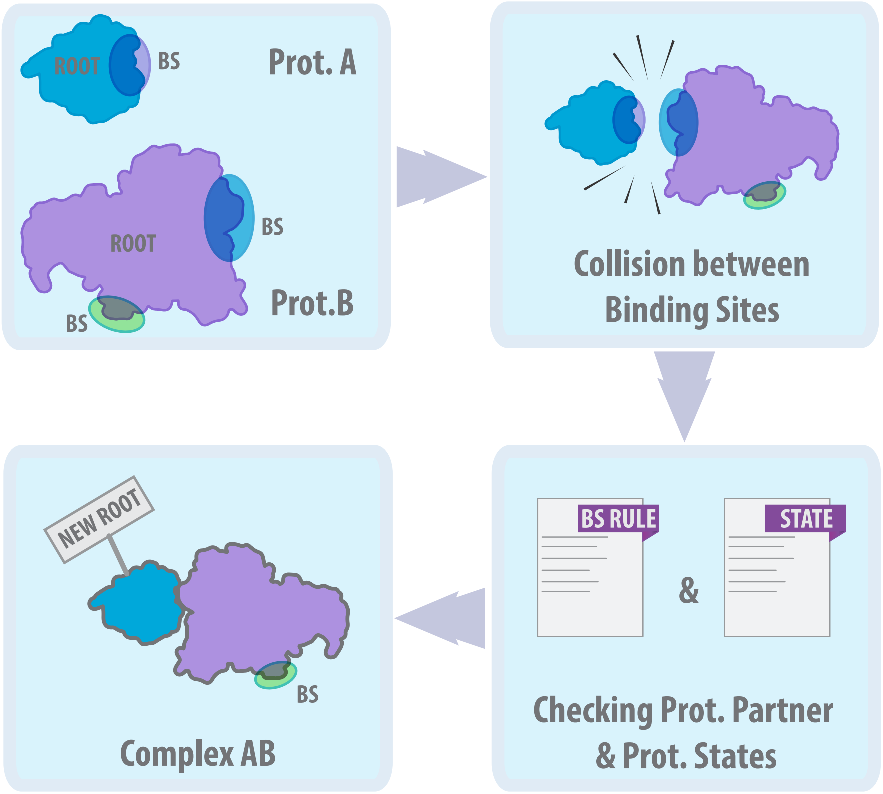


Figure 2.16: Complex formation workflow: Protein A has a single binding site for interacting with protein B (purple BS). Protein B on the other hand, has two binding sites, one to interact with protein A (blue BS), and a second to interact with another protein. When protein A and B collide with the purple and blue binding sites respectively, the algorithm checks if both BS are free for binding, and if so checks the state of both proteins and the binding rule for this binding. A complex AB is then formed and a new root is assigned according to the states of the proteins.

Table 2.2: Rule based modelling: States combination

States	**BTM**	**Bound**	**Alone**
BTM	Rnd	BTM	BTM
Bound	BTM	Rnd	Bound
Alone	BTM	Bound	Rnd

Global rules of binding: states combination. When two proteins form a complex, in order to decide which of the two proteins will be the root of the new entity, states are assigned. The Bound To Membrane (BTM) state is the one with higher priority, followed by the Bound state and lastly, the state Alone. Combinations of the same state lead to a random choice of the the new root (Rnd).

root for the complex, and which will then become the child, I set up a set of rules, the global rules of binding, that are based on protein states. Deciding the new root, means to decide which protein will drive the diffusion of the complex. In case of cytosolic proteins is not really important, but when a cytosolic protein binds a protein bound to the membrane, is critical that the complex will not detach from it (unless specifically required).

I established 3 different protein states: Alone, Bound, and Bound To Membrane (BTM). The states have different priorities: the highest priority is given to the BTM state, and the lowest to the Alone state. The Alone state represents the monomeric state of a cytosolic protein. Bound represents the state of a cytosolic protein that has at least one other protein in its hierarchy. BTM defines a protein bound to the membrane, regardless of whether other proteins are in its hierarchy. As recapitulated in table 2.2 when two proteins of the same state have to form a complex, the new root is chosen randomly (Rnd). For any other combination the root is chosen according to the protein that possesses the higher priority state. For example if a protein in the Alone state has to form a complex with a protein in the Bound state, the new root will be the protein in the Bound state, while the protein in the Alone state will become the child.

When the root is chosen, then the mass of the complex is updated on the new behavioural root as the sum of the root protein and the child protein and the diffusion module on the child eliminated.

2.6.9 Velocity clamping rule

According to the MaxwellBoltzmann distribution, the speed of gaseous particles is a function of the temperature of the system and the mass of the particles (Maxwell, 1860). Although the distribution applies to ideal gases close to thermodynamic equilibrium, the mean velocity that derives from it can be used as a reference for a maximum allowed velocity. The probability density function for the speed, is defined as

$$f(v) = \sqrt{\left(\frac{m}{2\pi kT}\right)^3} 4\pi v^2 exp\left(-\frac{mv^2}{2kT}\right) \tag{2.7}$$

Where m is the mass of the particle, T is the absolute temperature of the system, and k is the Boltzmann constant. The mode of this distribution, or the most probable speed for a single particle is defined as

$$v_p = \sqrt{2kT/m} \tag{2.8}$$

Of course, this speed will be much higher than the most probable speed for any given protein, but can be assumed as the upper limit that cannot be exceeded. For this reason I defined a local velocity clamping rule, which is located on each protein on the behavioural root. The rule is checked when exiting a collision, when the PhysX calculates the exit trajectory and velocity. When two proteins, or a protein and the container, exit a collision, the magnitude of the velocity is checked against the Maxwell-Boltzmann mode speed and, if higher, clamped to its value.

2.6.10 Ending of the simulation

There are two main ways to end a simulation. One common solution adopted by many simulators, is to decide how long the simulation will last, and then stop the simulation when the desired time is reached. More interesting possibilities depend on the outcome of the simulation itself. Since the simulation environment was created taking into account the biological problem in the first place, the overall state of the simulation is constantly monitored. Doing protein-protein interaction simulations put the focus on complex formation, so one possibility that I encoded is to end the simulation when a desired number of complexes are formed. Taking advantage of the protein states, at any given time in the simulation it is possible to check if the number of monomers is less than a certain percentage. This is very advantageous in cases where the simulation time required to reach a certain end result is not known. Instead of making several attempts, the simulation can be simply stopped when, for example, 95% of all the proteins in the system are in a complex, or in the Bound To Membrane state (BTM). In T.A.R.S.I.D. it is possible to terminate a simulation either specifying a time limit or a state percentage.

2.6.11 Possible Outputs

Several parameters can be tracked during the course of a simulation. For instance, each molecule can be tracked singularly, and both position and rotation can be tracked at fixed intervals. This interval can be each time step of the simulation or any other time interval considered more convenient. It is possible to write a unique log file for each protein with position, rotation and the mean squared displacement, calculated according to equation 1.4 (subsection 1.4.5). The mean squared displacement (MSD) is a useful indicator for controlling the overall type of diffusion of a protein. Normal diffusions show a linear relationship between time and MSD during the course of a simulation, while anomalous diffusions are characterised by deviations from the linearity.

Upon binding events, the positions and rotations of binding partners can also be recorded in a log file. This data can be useful to create a tridimensional map of the binding events.

Other interesting parameters that can be retrieved from simulations are the overall number of collisions and the collisions that resulted in binding events. This type of data can be used in kinetic studies. Two main ways of counting the total number of collisions are provided. The first one is specific to each protein, thanks to a local collision counter module present on each protein. The collision module records the time at which the protein collides with other molecules in the system, and the respective collision counter.

The second method, is based on binding events. Every time a binding is performed in the simulation, all the collisions are counted from the collision counter modules on each protein. A single log file for each simulation is produced with the time at which the collisions were counted (time of the binding event), the total collision count, together with its mean, median, and standard deviation. For both methods, collisions between proteins and the simulation container can be disregarded if desired, so that only collisions between proteins are counted.

Another possible output of these simulations are the complexed protein geometries. Potentially it could be interesting to retrieve the geometry of the multi-protein complex at the end of the simulation, as a single mesh. A similar result can be achieved by using the recorded positions and rotations of each of the protein and then re-create in a 3D software program, like Maya, the last time step of the simulation or even the entire time sequence. Geometrical data, such as data of the meshes of large complexes generated by a simulation, can be subsequently used in other simulations.

Chapter 3

Simulation results and validation

> People assume that time is a strict progression of cause to effect, but actually — from a non-linear, non-subjective viewpoint — it's more like a big ball of wibbly-wobbly... timey-wimey... stuff.
>
> *The Doctor*
> Doctor Who

In this chapter, I present the validation of the method presented in the previous chapter, and two studies on binding kinetics.

The validation focuses on the diffusion of cytosolic and membrane proteins for 3D and 2D diffusion respectively (section 3.1). The main validation analysis consists of accessing the random walk nature of the simulated diffusion, since the method was implemented as such.

Two studies on binding kinetics follow. The first aims to assess the influence that different binding site areas have on the kinetics of a binding reaction (subsection 3.2.3). Two proteins were chosen to represent the two main important shape categories: globular and rod-like. The effect of different binding site areas were analysed for both proteins in different experimental setups. The second study shows that shape influences binding kinetics as well. This study compares the result from the previous one with data from simulations of primitive geometries, spheres and cylinders, that share the same surface area or the same volume of the globular and rod-like proteins.

In addition I present validation of the accuracy of the complexes obtained with my methodology. For this validation, I take advantage of an already established benchmark for protein-protein docking software, which aim to address the same question: comparing simulated complexes with experimentally derived structures (section 3.3).

3.1 Diffusion validation

3.1.1 Introduction

The diffusion validation consists of two parts: the first validates the diffusion of cytosolic proteins in 3D, the second validates the diffusion of membrane proteins in 2D. When particles move according to random walk, they display a linear relationship between the mean squared displacement (MSD) and time, as explained in subsection 1.4.5. Verifying that the simulated MSD is in linear relationship with the simulation time is a frequently used test to establish if a simulator is computing an unbiased random walk. In order to verify the accuracy of the method, diffusion coefficients were recalculated from the simulated MSD and compared to the theoretical ones used as input.

3.1.2 Methods

All the protein geometries were prepared starting from PDB files according to the procedure explained in section 2.4, apart from the cytosolic tails of the NMDAR and AMPAR receptors used for the 2D diffusion validation.The cytosolic tails of these receptors are intrinsically unstructured, like many other protein termini (Uversky, 2013), therefore they do not possess a unique structure which can be observed experimentally and then made available in the Protein Data Bank (Minezaki et al., 2007; Choi et al., 2013). These cytoplasmic tails were modelled according to data from Chen et al. (2008b), due to the lack of available structural information (figure 3.3). In their work, electron microscopy techniques were used to characterise the dimensions of NMDAR and AMPAR tails in hippocampal dendritic spines (table 3.1).

Diffusion and binding processes were simulated as described in section 2.6. For the diffusion validation sections, the mean squared displacements (MSDs) were calculated according to equation 1.5. From the MSD the diffusion coefficients of cytosolic proteins were calculated with the following equation:

$$MSD_{3D} = 6Dt \tag{3.1}$$

in which D is the diffusion coefficient expressed in $\mu m^2/sec$, and t the time in seconds. While the diffusion coefficients of membrane proteins were calculated following equation

Table 3.1: Glutamate receptor cytosolic tails dimension

Protein	Length (nm)	Width (nm)	Height (nm)	sample size (num)
AMPAR	18 ± 3	10 ± 1	5 ± 1	18
NMDAR	20 ± 2	14 ± 2	16 ± 4	14

3.2.

$$MSD_{2D} = 4Dt \tag{3.2}$$

In which D is the diffusion coefficient expressed in $\mu m^2/sec$, and t the time in seconds.

Data analysis were performed using the statistical package R 3.0.0 (R Development Core Team, 2013) and plots were produced using the same software program.

3.1.3 3D Diffusion validation

Three different proteins were used for validation of the 3D diffusion: bovine Chymotrypsin (PDB ID 1YPH), bovine Serum Albumin (PDB ID 3V03), and the rabbit muscle Glyceraldehide–3-phosphate dehydrogenase (PDB ID 1J0X) (figure 3.1). Experimental diffusion coefficients for these proteins were taken from the literature (Young and Carroad, 1980).

In the simulations the proteins were translated and rotated according to their mass, as explained in subsection 2.6.5 and subsection 2.6.6 using equations 2.2 and 2.5. The simulations were run without a container, giving the proteins a potential infinite volume to diffuse in, in order to reproduce the dilute experimental reference condition and to avoid any crowding effects. For each protein, a simulation containing 100 instances was run for 1800 sec, with a time step of 0.01 sec. For each of the tested proteins, a linear relationship between time and MSD can be observed, as shown in figure 3.2. The linear correlation illustrate that the simulator is producing unbiased random walks for proteins diffusing in 3D.

A comparison of the diffusion coefficients calculated from the simulations (simulated diffusion coefficient), and the one used as input (theoretical diffusion coefficient) are

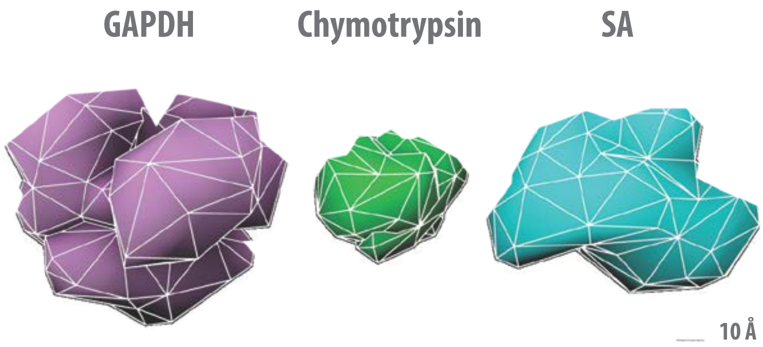


Figure 3.1: 3D diffusion validation proteins: three proteins were used for the validation of the diffusion in cytosol, GAPDH glyceraldehide-3-phosphate dehydrogenase (PDB ID 1J0X), Chymotrypsin (PDB ID 1YPH), and serum albumin (PDB ID 3V03). All three proteins were prepared according to the protein preparation procedure described in the previous chapter.

Table 3.2: Comparison of diffusion coefficients for test proteins diffusing in 3D

Protein	Mass (Da)	Simulated Diffusion Coefficient (μm^2/sec)	Theoretical Diffusion Coefficient (μm^2/sec)
Chymotrypsin	21600	87.9	87.6
Serum albumin	67000	60.4	60.2
Glyceraldehide 3 phosphate dehydrogenase	136800	47.5	47.4

presented in table 3.2. Comparing the diffusion coefficients calculated from the simulations and the one used as input for calculating the displacement in first instance, is part of a methodology accuracy test (Andrews et al., 2010). A smaller error between the simulated and the theoretical diffusion coefficients indicates more accurate simulations.

The diffusion coefficients recalculated from the simulation results reproduce the theoretical ones used as input well, with a percentage error of 0.34% for chymotrypsin, 0.33% for serum albumin, and 0.21% for GAPDH. The main source of these errors are probably rounding errors due to floating point arithmetics in step displacement calculations.

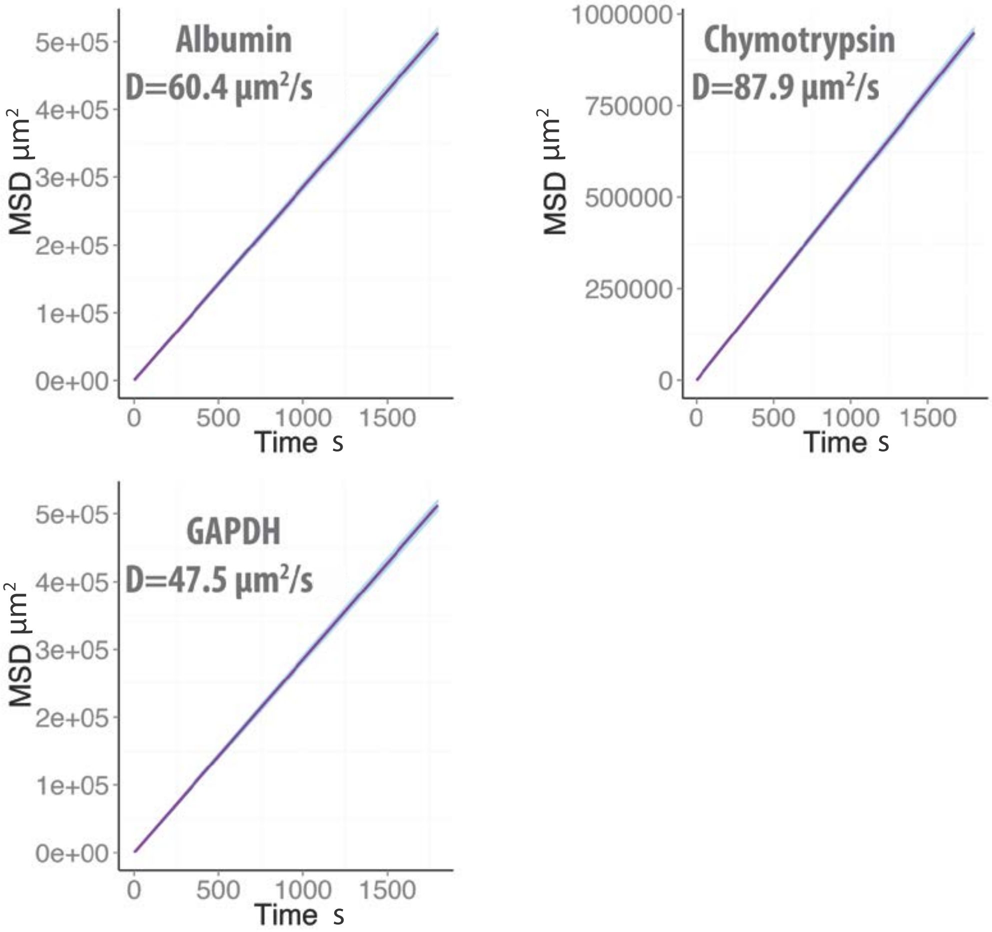


Figure 3.2: 3D diffusion validation: 100 molecules of chymotrypsin, 100 molecules of serum albumin (SA) and 100 molecules of Glyceraldehide 3-phosphate dehydrogenase (GAPDH) were simulated for 1800 seconds with a time step of 0.01s. The MSD for each protein, was calculated at 1 sec interval for the duration of the simulation, and plotted against time. For each time point, the standard deviation of the mean was used to estimate the error on the MSD, shown as light blue error bars with extension mean ± standard deviation. The linear regression line is shown in purple, $R^2 = 1$ for chymotrypsin and GAPDH, R^2=0.999 for albumin.

3.1.4 2D Diffusion validation

For the 2D validation, I used a coarse grained representation of two glutamate receptors, the N-methyl-D-aspartate receptor (NMDAR) and alpha-amino–3-hydroxy–5-methyl–4-isoxazolepropionic acid receptor (AMPAR) (figure 3.3).

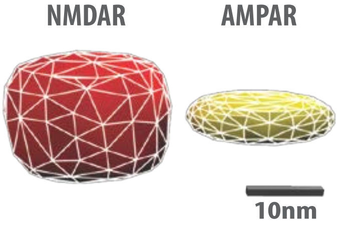


Figure 3.3: 2D diffusion validation proteins: two proteins were used in this validation AMPAR (yellow) and NMDAR (red). Both receptors were modelled based on the experimental data presented in Chen et al. (2008b) and reported in table 3.1.

Simulations were performed with 100 NMDAR and 100 AMPAR molecules for 1800 seconds, on a spherical container with a radius of $100\mu m$. For each protein species, the squared displacement was computed every second, and the mean calculated over the 100 molecules for NMDAR and AMPAR. The diffusion coefficients used as input in the simulation were calculated using a molecular weight of 543480 Da for NMDAR, and 400654 Da for AMPAR. Diffusion coefficients for membrane bound proteins need to be scaled by a factor of 0.001, as explained in subsection 2.6.5. This resulted in a diffusion coefficient of $0.033\mu m^2/sec$ for AMPAR and $0.030\mu m^2/sec$ for NMDAR. Experimentally derived diffusion coefficients for these receptors are difficult to estimate, due to the vast amount of interactions that both receptors have with scaffolding and signalling proteins in their native environment, a dendritic spine. Nonetheless, these receptors are less frequently involved in binding and show faster diffusions in the extra-synaptic space (Choquet, 2003). Different studies show different diffusion coefficients but for both receptors the fast diffusion happens in a range from 10^{-1} to $10^{-2}\mu m^2/sec$ (Bard et al., 2010; Groc, 2006; Alcor et al., 2009; Ehlers et al., 2007; Groc et al., 2004), which compares well to the estimated coefficients used as input for these simulations.

Table 3.3: Comparison of diffusion coefficients for test proteins diffusing in 2D

Protein	Mass (Da)	Simulated Diffusion Coefficient (μm^2/sec)	Theoretical Diffusion Coefficient (μm^2/sec)
NMDAR	543480	0.030	0.030
AMPAR	400654	0.033	0.033

The Mean Squared Displacement (MSD) is in linear relationship with time, evidence of an unbiased random walk for both receptors as shown in figure 3.4. The diffusion coefficients calculated from the simulation results are equivalent to the ones used input in the simulations, verifying that the simulator is performing correctly and consistently in cases of 2D diffusion (see table 3.3).

For membrane receptors like NMDAR and AMPAR, experimentally derived diffusion coefficients shows that they can diffuse faster or slower depending on the physical location of the receptors, as discussed above for synaptic and extra-synaptic environments. Membrane receptors can also show different types of diffusion. While a diffusion that shows a linear relationship between MSD and time is categorised as normal, two other types of diffusion are known, generally called anomalous diffusion: superdiffusion and subdiffusion. A molecule superdiffuses when its MSD has a non-linear relationship with time, and the MSD increases over time; it subdiffuses, when the MSD decreases over time in a non-linear relationship (Dix and Verkman, 2008). The latter is more common for membrane receptors and seems to be caused by interactions with lipids and other proteins (Saxton, 1994) (Saxton, 1996).

One of the main features of the method developed in this thesis is its flexibility in terms of geometries used in the simulation. Proteins as well as the simulation container itself have arbitrary geometries. To assess that the use of a non-uniform container will not affect the behaviour of the proteins that diffuse on its surface, I performed a test using an irregular container for the simulation of NMDA and AMPA receptor diffusion. The sphere used as container in the previous test was deformed in Maya using a sinusoidal wave deformer resulting in the geometry presented in figure 3.5. The simulations were analysed as for the previous test and the results are reported in figure 3.5. Although using an irregular container could potentially cause accumulations of diffusing molecules in its concave parts, resulting in erroneous anomalous diffusions, the simulated diffusion

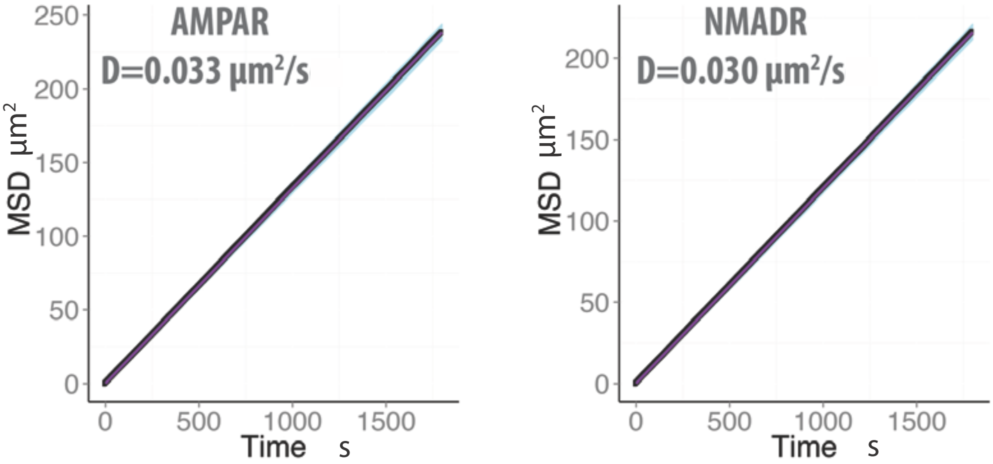


Figure 3.4: 2D diffusion of membrane receptors: 100 AMPAR and 100 NMDAR were simulated for 1800 seconds, with a time step of 0.01 sec. The MSD was calculated at 1 sec interval for the duration of the simulation for each protein, and plotted against time. For each time point, the standard deviation of the mean was used to estimate the error on the MSD, shown as light blue error bars with extension mean ± standard deviation. The linear regression line is shown in purple, $R^2 = 1$ for both AMPAR and NMDAR.

resulted in an unbiased random walk. This demonstrates that the simulator does not create abnormalities in the 2D diffusion of proteins when irregular surfaces are used. Moreover the recalculated diffusion coefficients from the linear regressions were the same as before and they accurately reproduced the diffusion coefficients used as input.

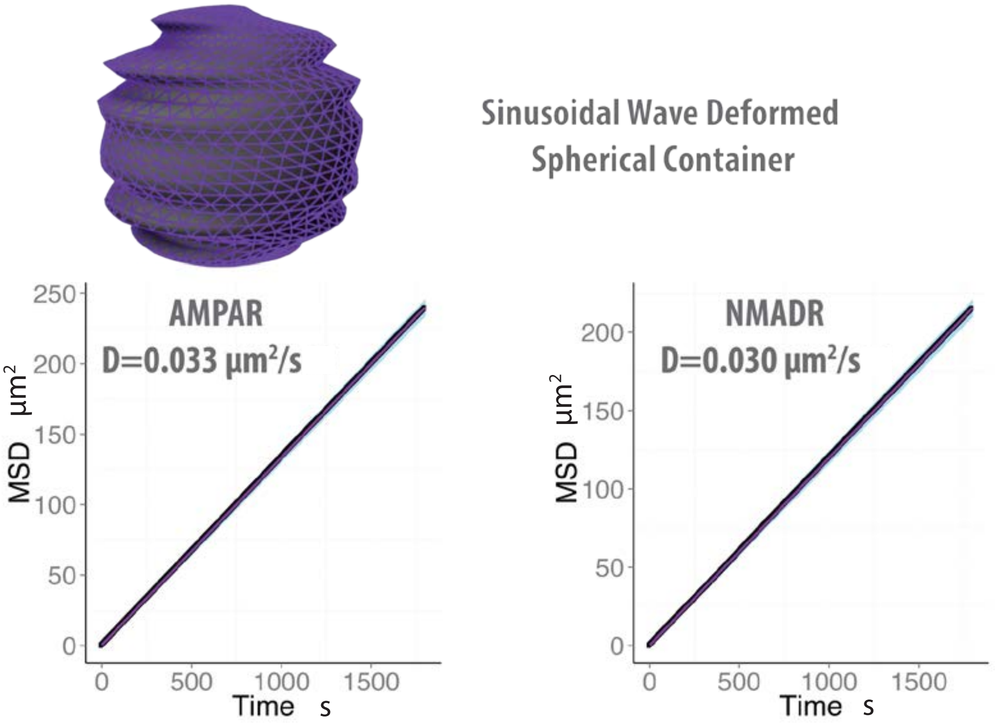


Figure 3.5: 2D diffusion of membrane receptors on rough surface. The spherical container was deformed using the nonlinear sine deformation algorithm of Maya. Simulation setups and representation as for figure 3.4

3.2 Study on protein-protein association kinetics

3.2.1 Introduction

Protein associations are the core of signal transduction and therefore at the base of every physiological processes. To study the kinetics of a reaction, or the velocity of a reaction, firstly one must analyse the time course of the reaction in respect to the reactant or the products (Marangoni, 2003) and then determine its rate constant. Rate constants are a concentration independent measure of the velocity of a reaction, which makes them a very good parameter for comparison of different reactions and different condition setups. In order to calculate rate constants the *rate equation* of the reaction of interest, must be evaluated. The *rate equation* is a quantitative expression of the change in concentration of reactant or product molecules over time. For a generic reaction like the one in equation 3.3 the rate equation will be formulated as in 3.4 .

$$aA + bB \rightarrow cC \tag{3.3}$$

$$rate = -\frac{1}{a}\frac{\mathrm{d}[A]}{\mathrm{d}t} = -\frac{1}{b}\frac{\mathrm{d}[B]}{\mathrm{d}t} = \frac{1}{c}\frac{\mathrm{d}[C]}{\mathrm{d}t} \tag{3.4}$$

The rate equation can be equally expressed as in the disappearance of the reactant or the appearance of the products over time. Considering, for example, one of the reactants, the rate equation can be expressed as follow:

$$rate = -\frac{1}{a}\frac{\mathrm{d}[A]}{\mathrm{d}t} = k[A]^a \tag{3.5}$$

where k is defined as the rate constant of the reaction in equation 3.3. By integration of the rate equation, is possible to obtain an expression that describes the variation of the reactant concentration over time, and the calculation of the rate constant k. For bimolecular reactions like the one used in these simulations, the integrated equation is expressed in 3.6.

$$\frac{1}{[A_t]} = \frac{1}{[A_0]} + kt \tag{3.6}$$

Where $[A_t]$ is the concentration of the reactant A at time t, and $[A_0]$ is the concentration of A at time $t = 0$. Plotting the inverse of the time dependent concentrations of the reactant A over time will generate a linear plot, in which the slope of the curve is the rate constant k.

The binding of two proteins is influenced by many factors. According to collision theory, the first step of any interaction is the collision between the two proteins, yet not all the occurring collisions result in a binding event. In order for two molecules to bind, they need to be in the correct orientation and possess a sufficient amount of energy to overcome the activation energy (Upadhyay, 2010). In the method presented in this thesis, energy levels are not explicitly modelled, while orientation has a major impact on the result of these simulations. For example, if two proteins A and B were to react in a simulation to give the complex AB (figure 3.6) upon collision not all the orientations of the two proteins will lead to complex formation, but only the one involving the two binding sites will be successful collisions.
With this premise, it is obvious that the definition of the binding site area will influence the number of collisions that will be considered successful collisions. The bigger the area, the more likely it is for two molecules to collide with the correct, binding site to binding site, orientation.

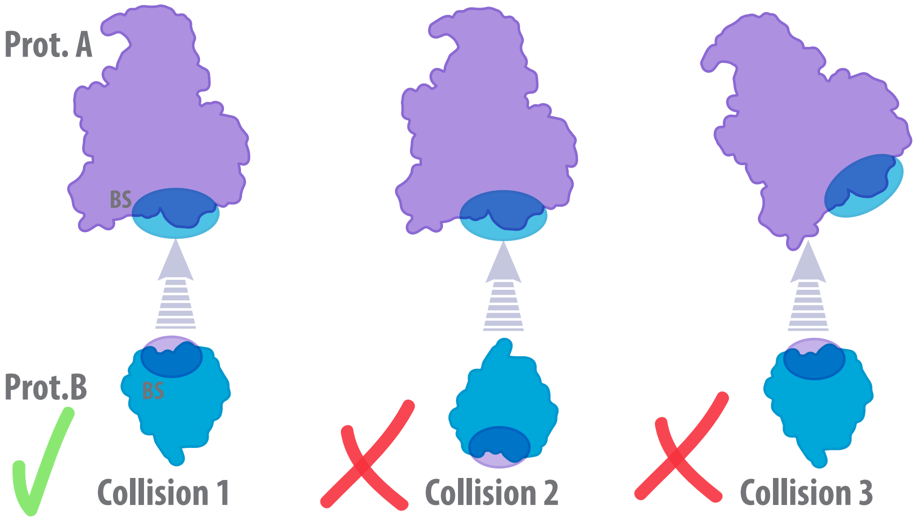


Figure 3.6: Collision Efficiency. Protein A and protein B form a complex AB interacting by their binding site (labeled BS) but not all the collisions between the two result in the complex formation. Only when the two binding sites collide with each other, as represented in collision 1, the collision is considered efficient and lead to complex formation.

3.2.2 Methods

Diffusion and binding processes were simulated as described in section 2.6. Data analyses were performed using the statistical package R 3.0.0 (R Development Core Team, 2013) and plots were produced using the same software program.

3.2.3 Influence of binding site area on binding kinetics

To study in details the effect that different binding site areas have on the amount of successful collisions and ultimately on the binding kinetics, I set up a series of simulations for two different dimerisation reactions. One dimerisation reaction occurs between the two globular monomers of the beta-ketoacyl-acyl carrier protein synthase III (FabH) (PDB ID 1EBL), and the other one occurs between the two monomers of the rod-like protein insect antifreeze protein (AFP) (PDB ID 1EZG). These two proteins were selected for this study since they both form dimers, and their experimentally solved geometries highly resemble one of a sphere (FabH) and the other one of a cylinder (AFP). These characteristics make the two proteins a good choice not only for kinetic studies but also for a comparative study with primitive geometries (as described in subsection 3.2.4).

The two monomers were prepared as described in section 2.4 and for both monomers, 4 different instances were prepared with 4 different binding site areas. One binding site area was defined as the entire protein surface, allowing the entire protein area to be available for binding. Another instance was prepared with about half of the protein surface defined as binding site, and others were defined with a big and a small binding site area. The percentages of protein areas defined as bindable for each monomer are shown in table 3.4. The difference in percentages between the two proteins is due to the different tessellation of the two meshes and their low poly count, for which adding a triangle more on the selection creates a considerable difference in the percentage of the selected area.

The simulations were carried out in a cubic container of dimensions 160x160x160 nm^3 until 95% of the monomers had reacted to form the dimeric complex. Three different sets of conditions were also considered. First a classic homodimerisation reaction was encoded in the binding rules of the monomers according to the following equation:

$$2A \rightarrow C \tag{3.7}$$

Table 3.4: Different binding site areas defined for FabH and AFP

Binding Site Area	Percentage of FabH Area	Percentage of AFP Area
All	100	100
Half	53.5	43.1
Big	44.2	33.2
Small	15.7	12.7

Table 3.5: Condition setups for the influence of binding site area on binding kinetics simulations

Setup Label	Reaction Equation	Monomer Concentration (# of molecule)
2A	$2A \rightarrow C$	200A
2A HalfCon	$2A \rightarrow C$	100A
A+B	$A + B \rightarrow C$	100A + 100B

In which 2 instances of the protein A form the complex C, the homodimer. The first condition setup is defined by this type reaction and a concentration of monomer used as 200 molecules. The second setup also uses the same reaction type, but half the concentration of the monomer. In these two conditions, each protein is able to bind every other protein in the system, since they are all exactly equivalent. The third condition setup uses a different reaction rule which is encoded by the following equation:

$$A + B \rightarrow C \tag{3.8}$$

Equation 3.8 is a classical bimolecular reaction, in which two different molecules A and B react to form a complex C. In this setup, the monomers were divided into two pools, one labeled as A and the other as B. The product C of such a reaction is identical to the one encoded using equation 3.7 but in this case a monomer labeled as A is only able to interact with one labeled as B and vice versa. Interactions A-A or B-B are not permitted in this setup. The concentrations used for this condition setup were 100 molecules of monomer A and 100 molecules of monomer B. The three condition setups with the respective monomer concentrations used are summarised in table 3.5.

Twelve simulations were carried out for each homodimer with the four different binding site areas, (table 3.4), in the three different conditions (table 3.5). From time course analysis of the simulations the influence of binding site areas on collision efficiency is reported in figure 3.7 for the globular protein FabH, and 3.8 for the rod-like protein AFP. The plots show the inefficient collisions which are expressed as the number of collisions that did not result in a binding event. As expected both proteins show the same trend: increasing the binding site area decreases the number of inefficient collisions to almost zero when the entire protein surface is defined as binding site (labeled All, red line for both plots). Suggesting that when the entire protein is defined as bindable, almost all the collisions in the system result in a binding event, which does not accurately reflect with experimental evidence (Upadhyay, 2010).

Although it is clear that changes in the amount of surface defined as binding site influence the overall collision efficacy, it is interesting to study in more details its influence on the binding reaction kinetic.

For both proteins and each simulation setup, a linear plot was created following equation 3.6 and the respective rate constant was calculated from the coefficient of each linear regression. In order to present the results in a more compact way, a histogram with all the calculated rate constants was created for both proteins (figure 3.9 a and 3.10 b). From the plots in figure 3.9 and 3.10, it is clear that for both proteins, the bigger the binding site area the faster is the binding reaction. In all cases the association rates are higher than the experimentally calculated ones, which range from 10^3 to 10^9 $M^{-1}sec^{-1}$ (Schreiber et al., 2009). This discrepancy could be due to an over estimation of the protein diffusion rate in the simulation volume, that could lead to a faster kinetic. Moreover a clear difference in the association rates between FabH and AFP can be appreciated comparing figure 3.9 and 3.10. In all the settings FabH shows much higher association rates compare to the rod-like protein AFP. This discrepancy could be cause by not only to different diffusion rates, but also to the significant difference in geometry conformation of the two proteins that, as shown in the following section, can have an impact on the binding kinetics as well.

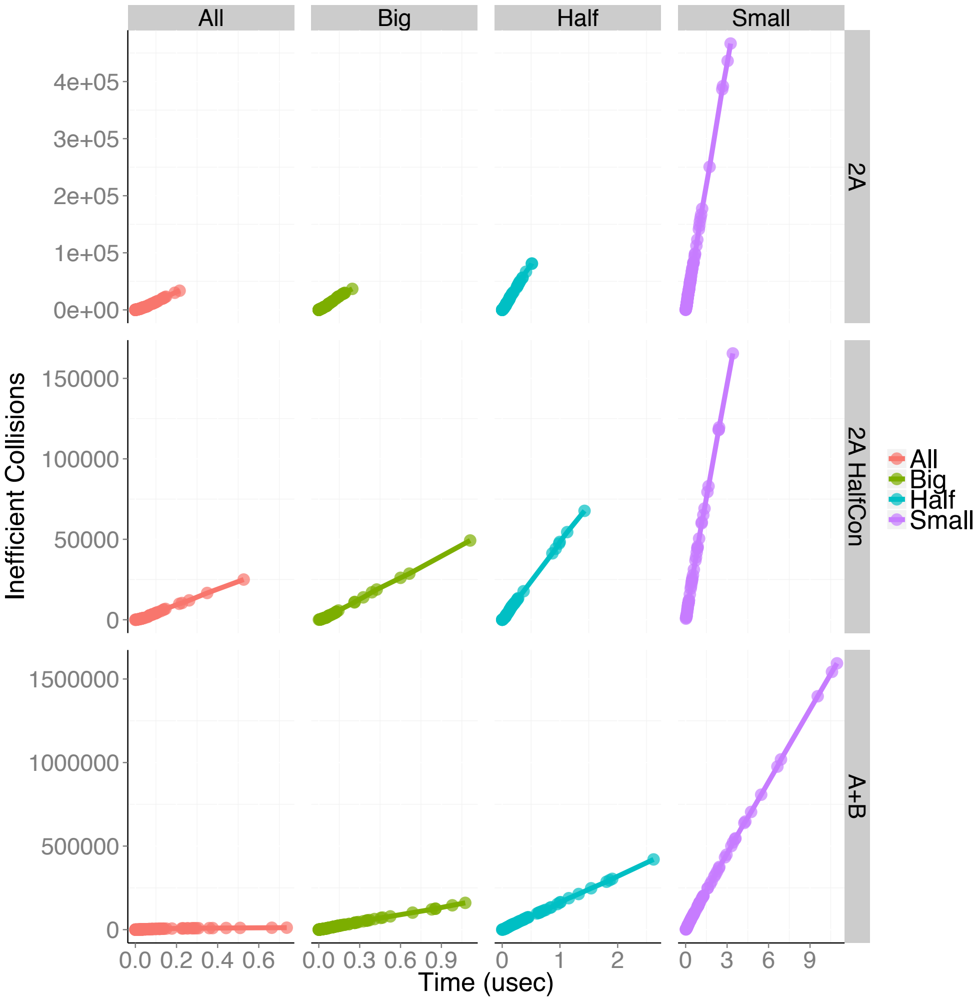


Figure 3.7: Dependency of the collision efficacy for the FabH protein with the defined binding site areas: Time course of the collision efficacy calculated as the number of occurred collisions that did not result in binding events, for a globular protein FabH. Twelve experiments were carried out with four different binding site areas (all, half, big and small) in three different conditions (2A, 2A HalfCon and A+B). Lines with a slope close to zero indicate that all the occurred collisions resulted in a binding event, while for inclined lines the number of inefficient collision increases over time. Differences in the duration of the simulations are due to difference in kinetics, since every simulation was stopped when 95% of the monomers reacted.

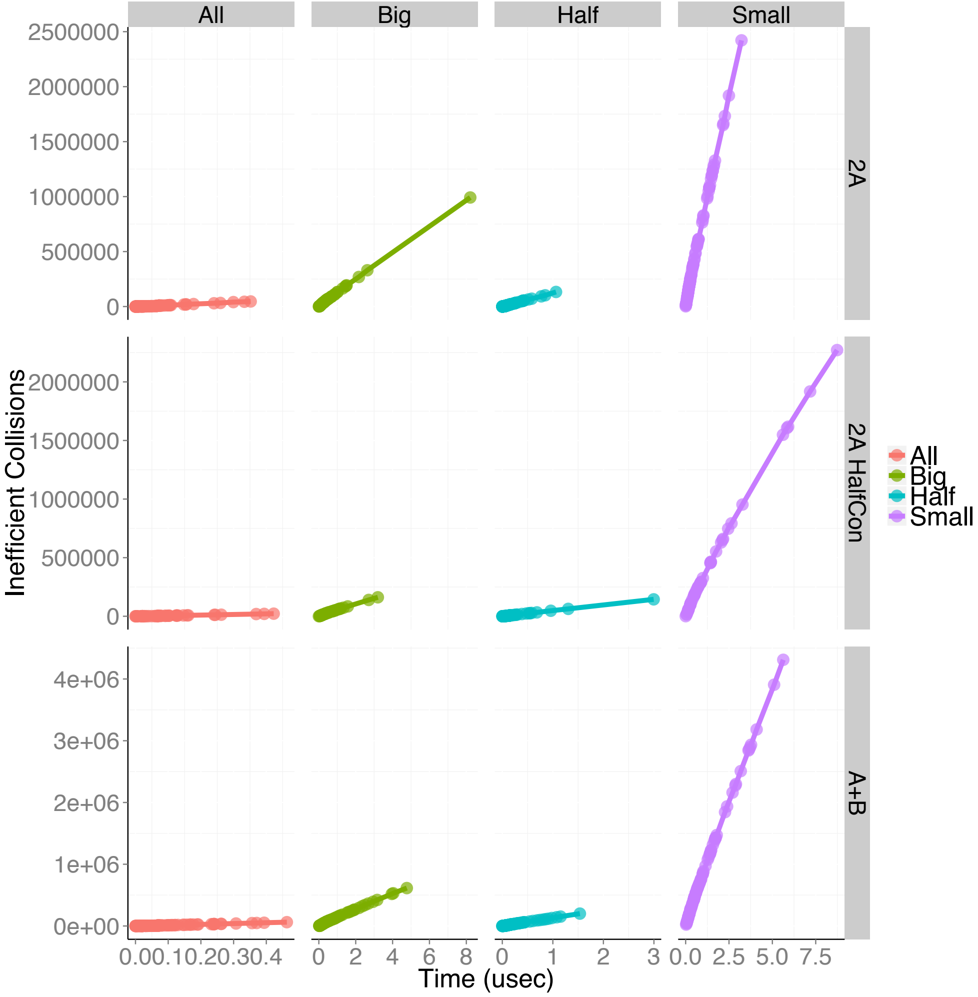
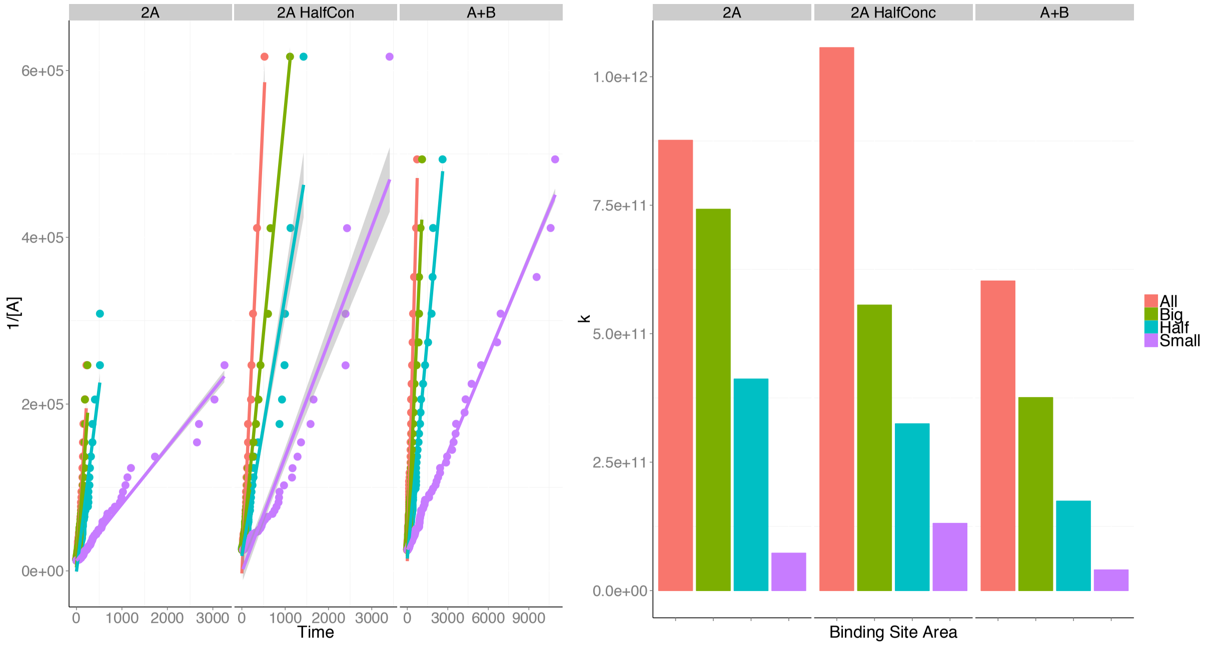


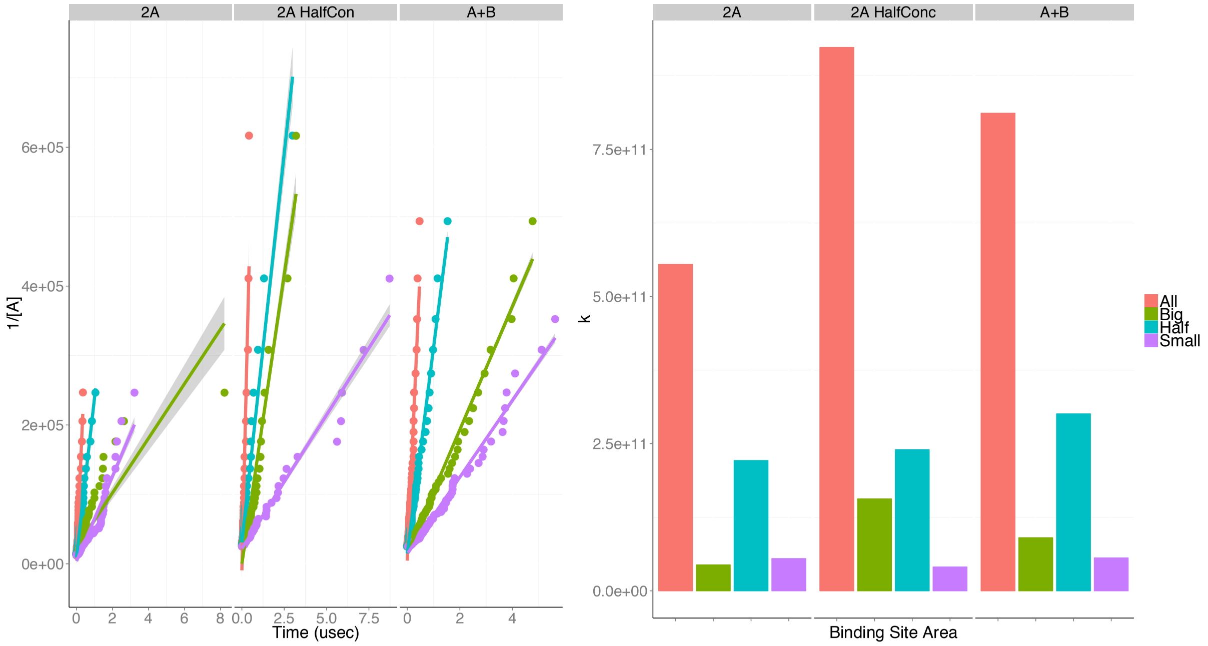
Figure 3.8: Dependency of the collision efficacy for the AFP protein with the defined binding site areas: Time course of the collision efficacy calculate as the number of occurred collisions that did not result in binding events, for a rod-like protein AFP. Twelve experiments were carried out with the same setup as in figure 3.7.



(a) Linear plot

(b) Histogram of the rate constant

Figure 3.9: Influence of binding site areas on binding kinetics for the globular protein FabH: (a) Linear plot of changes of reactant concentration over time, for a second order reaction used for the calculation of the kinetic constant k. (b) Histogram representation of the kinetic constants calculated as the slope of the linear regression curves from plot (a)



(a) Linear plot

(b) Histogram of the rate constant

Figure 3.10: Influence of binding site areas on binding kinetics for the rod-like protein AFP: (a) Linear plot of changes of reactant concentration over time, for a second order reaction used for the calculation of the kinetic constant k. Standard deviation represented in grey. (b) Histogram representation of the kinetic constants calculated as the slope of the linear regression curves from plot (a).

3.2.4 Influence of shape on binding kinetics

Apart from the influence of different binding site area, the effect that different shapes have on the binding kinetics is another important factor to analyse. More specifically I aimed to investigate the difference between using the models prepared according to the proposed method in this thesis (section 2.4), and a simple primitive geometry such as spheres and cylinders. Since FabH is a globular protein, its assigned primitive geometry was a sphere, while for the rod-like protein AFP a cylinder was chosen. For each type of primitive geometry two different models were created: one calculated to be equivalent in volume and one calculated to be equivalent to its surface area. From the PDB geometries of FabH and AFP(produced following the methodology reported in section 2.4) volumes and surface areas were calculated. Then two spheres were created for FabH: one with the same volume of the FabH PDB geometry (equivalent volume), and one with the same surface area of the FabH PDB geometry (equivalent area). The same process was applied to AFP in which two cylinders were created to be equivalent in volume and in surface area to the AFP PDB geometry.

In order to fully compare the simulations presented in the previous section, the same binding site areas setups (as in Table 3.5) were defined for each equivalent sphere or cylinder, resulting in a total of 36 different simulation setups.

Overall for the globular protein FabH, reducing the protein shape to a sphere seems to have a greater effect the smaller the defined binding site area, as shown in figure 3.11. Also intrinsically slower reactions like A+B seem to increase the effect that different shapes have on the binding kinetics. For the fast reactions like the one with 2A kinetic and with more than 45% of surface area defined as binding site, the equivalent area sphere perform similarly to the PDB based model. While with slower reactions, like the one involving small binding sites, the PDB model shows the slowest kinetics.

A similar effect can be seen for the protein AFP, figure 3.12. For all the conditions except for the one that involved the entire protein surface defined as binding site, the PDB based model shows a slower kinetic compared to the equivalent area and equivalent volume models. For AFP, when the binding site area was defined as the entire surface, the model that performed similarly to the PDB based one was the equivalent volume. This finding is in contrast to the protein FabH, for which the equivalent surface area models showed a similar behaviour to the PDB models. A possible reason for this difference is the difference in dimension of the two models: the cylindrical equivalent surface area models were bigger than the equivalent volume models, resulting in a higher collision probability. Since the binding sites were defined as the entire protein surface, all the collisions between the objects resulted in a binding event, causing to the bigger geometry to display the fastest kinetics.

From collision theory, it is possible to estimate the amount of collisions in a system over a certain period of time. For the estimate, the molecules are considered as rigid spheres and moving with a velocity proportional to their mass. In case of a system composed only by a single type of molecule, the number of collisions per second or the collision frequency is calculated with the following equation (Upadhyay, 2010):

$$CollisionFrequency = 2\sigma^2 n^2 \sqrt{\pi k_b T/m} \tag{3.9}$$

In which σ is the diameter of the molecule, n is the number of molecules in the system per unit of volume, k_b is the Boltzmann constant, T is the absolute temperature of the system, and m is the mass of the molecule.

Comparison of the collision frequency calculated from theory and the simulated collision frequency of the system, can give insight on the simulation accuracy. For each simulation setup previously discussed, the collision frequency was calculated as the total number of collisions in the system at the end of the simulation divided by the simulation time. Data are divided in two plots, one for the globular protein cases, and one for the rod-like (figure 3.13 and 3.14, respectively). Four theoretical collision frequencies were calculated for each plot using different molecule diameters and molecules numbers: the frequency calculated with equivalent volume sphere diameter at normal, and half concentration; and the frequency calculated with the equivalent area spheres at normal and half concentration. The same principle was applied for the rod-like cases.

This analysis shows that the simulated collision frequency are indeed in the same range as the ones calculated from collision theory. Moreover the analysis indicates that the binding site areas does not play a role in the overall collision frequency, confirming that the different binding kinetics reported in the previous paragraph are due to changes in the amount of efficient collisions. Differences between the simulation conditions are seen when different shapes are used, in particular for the case of the globular protein FabH and its two equivalent spheres: for the three experimental conditions the PDB

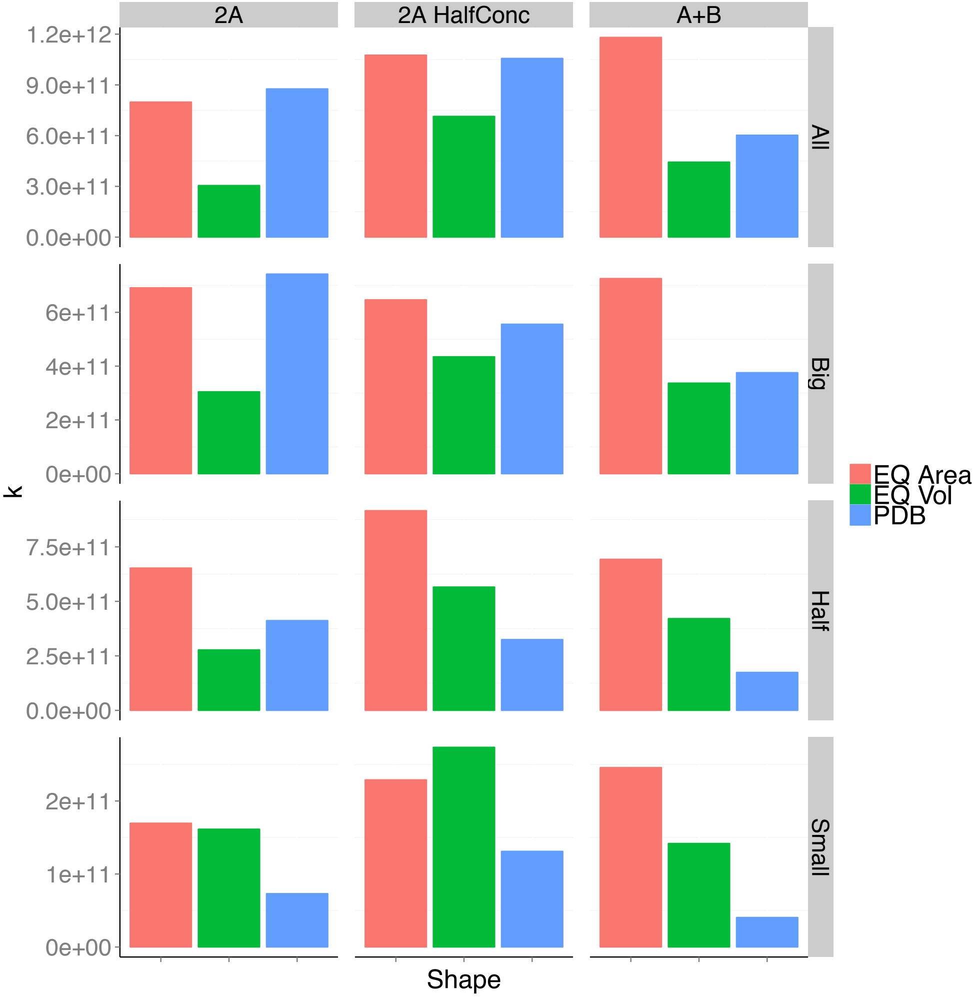


Figure 3.11: Effect of shape on binding kinetics for the globular protein FabH: data are split between the different simulation conditions (2A: bimolecular reaction of type $2A \rightarrow C$ with initial concentration of 200 molecules, 2AHalfConc same reaction but with a starting concentration of 100 molecules, A+B bimolecular reaction of type $A + B \rightarrow C$ with initial concentration of 100 molecules of A and 100 molecules of B) and different binding site areas (different percentages of the overall areas defined as binding site are reported in table 3.4). For each condition the binding kinetics for the three different geometries are reported: labeled as PDB is the geometry calculated from the PDB, EQ Area is a sphere with an area equivalent to the PDB model, while EQ Vol is a sphere with a volume equivalent to the PDB model.

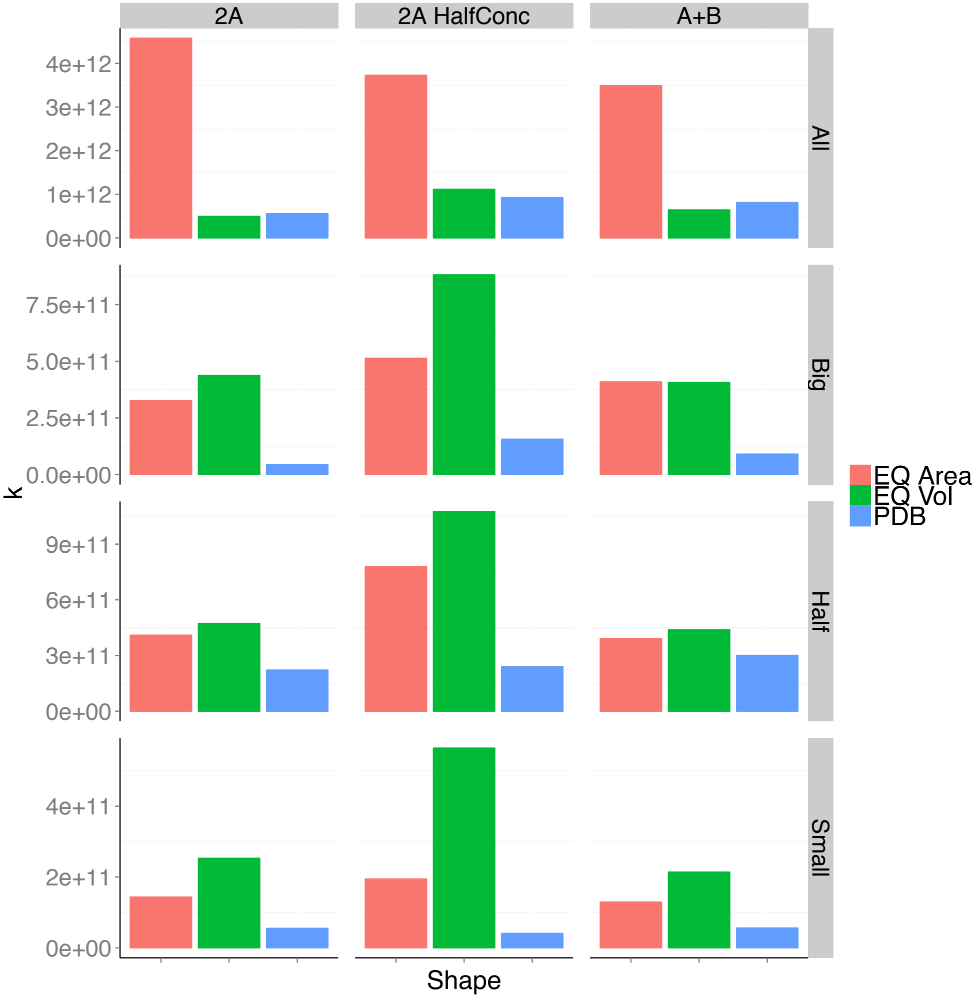


Figure 3.12: Effect of shape on binding kinetics for the rod-like protein AFP: data are split between the different simulation conditions and different binding site areas as for figure 3.11. For each condition the binding kinetics for the three different geometries are reported: labeled as PDB is the geometry calculated from the PDB, EQ Area is a cylinder with an area equivalent to the PDB model, while EQ Vol is a cylinder with a volume equivalent to the PDB model.

derived model shows a higher collision frequency compared to the equivalent volume and equivalent area spheres. This is in itself an indication that shape plays a role in the binding dynamics already influencing the overall collision frequency of the molecules. The increase in collision frequency is probably due to the irregular shape of the PDB derived geometries, that thanks to their protrusions are more likely to get in contact with other molecules.

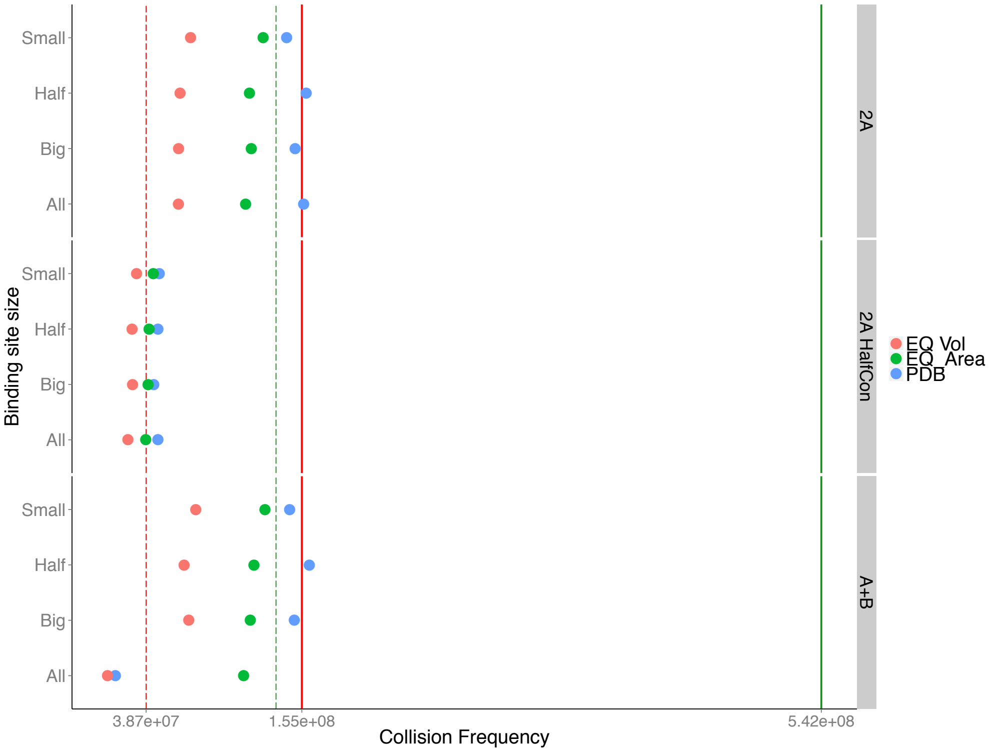


Figure 3.13: Comparison between simulated collision frequency and theoretical for globular shapes: collision frequency for the different shapes (PDB derived in blue, equivalent volume sphere in red, and equivalent area sphere in green) were calculated for the different binding sites (all, half, big, small as in table 3.4) as the number of total collisions in the system divided by the simulation time. Theoretical frequencies were calculated according to equation 3.9 and adjusted for the simulation volume. Red solid line represent the theoretical frequency calculated with equivalent volume sphere diameter at 200 molecules concentration, while red dashed line is for the half concentration (100 molecules); the green lines represent the frequencies calculated with the equivalent area spheres at normal (solid line) and half concentration (dashed line).

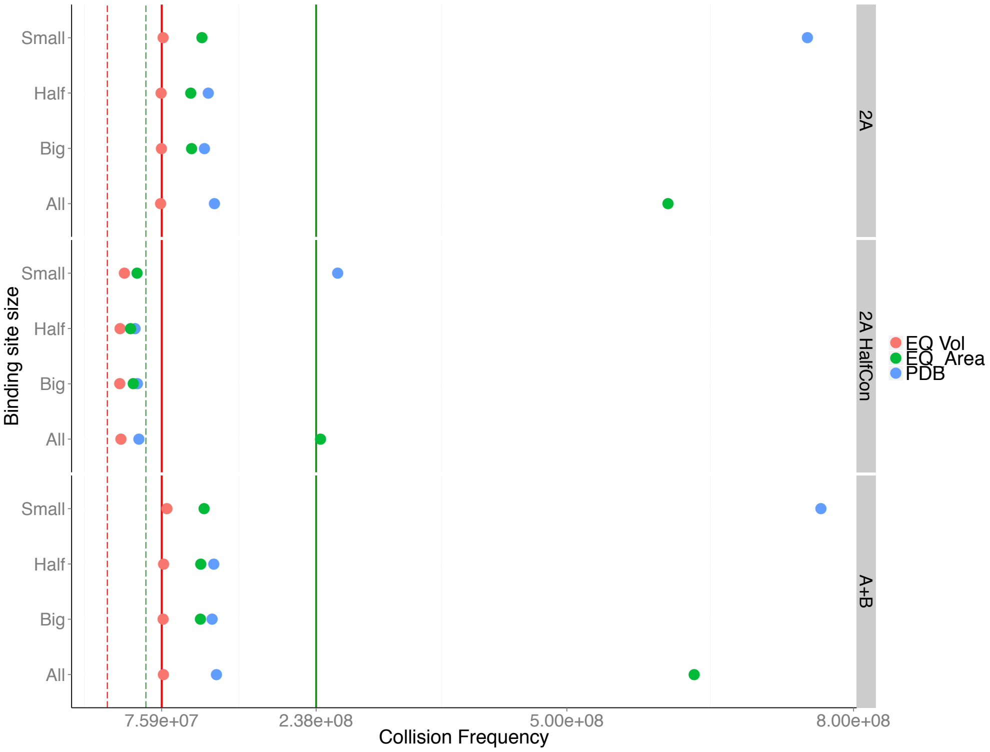


Figure 3.14: Comparison between simulated collision frequency and theoretical for rod-like shapes:collision frequency for the different shapes were calculated for the different binding sites, as the number of total collisions in the system divided by the simulation time. Theoretical frequencies were calculated according to equation 3.9 and adjusted for the simulation volume. Red solid line represent the theoretical frequency calculated considering the equivalent volume at a concentration of 200 molecules, while red dashed line is for the half concentration (100 molecules); the green lines represent the frequencies calculated with the equivalent area at normal (solid line) and half concentration (dashed line).

3.3 Complex formation validation

3.3.1 Introduction

To assess the performance of the method with regards to complex formation, I took advantage of an already well established method used for protein-protein docking benchmarking. Protein-protein docking is the computational modelling of the quaternary complex formed by two or more proteins from the crystal structures of the separate units. The benchmark used for protein-protein docking consists of a set of test cases for which the experimentally solved structure of a complex and its separate subunits are available. The separately solved subunits are used in the simulations and the resulting complex is compared with the experimental one.

Protein-protein docking started in 1991 with the work of Shoichet and collegues (Shoichet and Kuntz, 1991) and currently several software programs are available to perform this type of calculations, importantly: Z-Dock (Pierce et al., 2011), RosettaDock (Gray et al., 2003) and GRAMM-X (Tovchigrechko and Vakser, 2006). Despite some differences, all of them use a scoring function to drive the complex formation based on residue contacts, shape complementarity, force field based free energy estimations, and clustering coefficients. All the parameters of the scoring function are calculated from the atomic protein coordinates and data stored in the PDB files of the separate units. The only information required to compute a three dimensional structure of a binary complex is the structure of the two separate subunits.

A first standardised benchmark was made publicly available in 2002 in order to compare how well different protein docking methods performed. This benchmark, known as Benchmark 0.0, consisted of 54 binary complex test cases. Each test case comprises of a PDB structure of the complex used as the experimental reference, and the two separately solved crystal structures of the two proteins forming the complex used as input for the protein-protein docking calculation. Performance assessment between different methods is carried out by comparing the experimentally solved geometry of the complexes and the one obtained with the docking software. The comparison is made by superimposing the modelled complex with the experimental one and subsequently calculating the average distance between the two. This distance parameter termed RMSD, for Root Mean

Squared Displacement is calculated as in equation 3.10.

$$RMSD(v,w) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(v_{ix} - w_{ix})^2 + (v_{iy} - w_{iy})^2 + (v_{iz} - w_{iz})^2} \tag{3.10}$$

The RMSD is a pairwise squared distance calculated between each atom in the modelled complex v_i and its corespondening atom in the experimental complex w_i. The squared distances are averaged and the RMSD is defined as its square root. Usually, the RMSD is calculated only between the C_α atoms forming the protein backbone, since these are less sensitive to thermal motion and therefore more informative for detecting conformational changes.

The benchmark used for the validation of the complex formation in my methodology is the 2012 update of Benchmark 4.0 (Hwang et al., 2010) which includes 176 test cases divided into three categories: rigid-bodies, medium difficulty and difficult. These categories are classified based on the iRMSD score of the proteins. The iRMSD performs a superimposition of the two considered structures prior the normal RMSD calculation. The iRMSD will always return the minimum RMSD for that protein pair, independently of how distant two proteins can physically be, and because of this it is used as a score of the conformational changes between the monomer in the unbound and bound state.

Proteins classified in the rigid-body category have an iRMSD score lower than 1.5 $\AA$ and almost do not undergo conformational changes, and for this reason is considered the easiest category. Proteins in the medium difficulty category present a conformational changes between the bound and unbound state in the range of 1.5 and 2.2 $\AA$. For the category labeled as difficult, the iRMSD scores are higher than 2.2 $\AA$. The higher the iRMSD score, the bigger the conformational change, and the more challenging the prediction of how the two proteins will bind is.

Not all the PDB found in Benchmark 4.0 were suitable to be used for the validation of the complex formation procedure of my method. In my method proteins are represented as simplified surfaces, meaning that the level of detail encoded in each geometry is limited. When peptides and small proteins are converted into these simplified geometries too much information is lost to make a clear definition of a binding site, due to their small dimensions. For this reason, I decided to not consider test cases from Benchmark 4.0 that involved proteins with less than 150 amino acids or peptides. As described in subsection 2.4.2, the binding site definition proposed for this method, is based on sequence information found in UniProt. Therefore all the test candidates required binding site annotations for both monomers in UniProt.

The following three cases were selected: the holoenzyme of human protein casein kinase II (1JWH), the Ras-related protein SEC4 in complex with Rab-GDI (1HE8), and RAS G12V in complex with PI3-Kinaseγ (3CPH). The first complex 1JWH is part of the rigid-body category, while 1HE8 and 3CPH belong to the medium complexity category.

3.3.2 Methods

Superposition and RMSD calculations are a standard operation for proteins. Even though algorithms for these calculations are available for the major proteins visualisation software, like VMD (Humphrey et al., 1996) and Chimera (Pettersen et al., 2004), these software could not be used for this validation since they do not support RMSD calculation and superposition of surfaces. For this reason I implemented this validation in Maya. For each of the three selected complexes 100 complex formations were simulated and analysed. The analysis consists of two separate steps, similarly to the calculation of iRMSD: superposition and RMSD calculation. A superposition was performed in order to overlap one of the subunits in the simulated complex with its counterpart in the reference complex structure. The rigid-body nature of the proteins in my method, and the binding strategy that does not convert proteins into a different object upon binding, allow protein geometries to be exactly the same in the unbound and the bound state. This had convenient repercussions not only for the superposition, for which the rotational and translation matrix were calculated based on three vertices only, but also for the calculation of the RMSD. The RMSD was calculated between the simulated and the reference structure for each vertex of the surface according to equation 3.10 and averaged between the 100 replicates.

Protein images were also performed in Maya, while RMSD plots were performed in R.

3.3.2.1 Protein models for complex 1HE8

The complex 1HE8 is formed by the human GTPases Ras and the human phosphoinositide 3-kinases gamma (PI3Kγ) (Pacold et al., 2000). The PDB for the experimentally solved single subunits are 1E8Z and 821P for PI3Kγ and Ras respectively.

1E8Z PDB file was loaded in Maya using the mMaya plugin and the surface was generated following the procedure described in subsection 2.4.1 resulting in a single mesh of 228 polygons (figure 3.15 panel A, in green). The binding site for Ras is annotated as PI3K-RBD (Ras Binding Domain)in the PI3Kγ UniProt entry (UniProt ID P48736). The 93 amino acids long sequence of the RBD was mapped onto the mesh resulting in a binding site consisting of 17 polygons.

The geometry for Ras (PDB ID 821P) was obtained with the same procedure resulting in a mesh consisting of 218 polygons (figure 3.15 panel A, in blue). The binding site for PI3Kγ , labeled in UniProt as Effector Region, consists of 9 amino acids and resulted in a binding site comprising 8 polygons(UniProt ID P01112).

3.3.2.2 Protein models for complex 3CPH

The 3CPH complex, like 1HE8, is also part of the medium difficulty category of Benchmark 4.0. The complex is composed of the Ras-related protein SEC4, and the Rab GDP-dissociation inhibitor, both from Saccharomyces cerevisiae.

The crystal structure of the GDP dissociation inhibitor Rab in its unbound state is 3CPI. The protein was prepared as in subsection 2.4.1 resulting in a mesh surface with a polygonal count of 207. The Rab protein, which is 451 amino acids long, has its binding site annotated in UniProt as a sequence of 26 amino acids (UniProt ID P39958) that resulted in a selection of 11 polygons as shown in figure 3.17 panel A, coloured in red with binding site in blue.

The structure used for the protein SEC4 is 1G17. The protein is 213 amino acids long and its binding site, as reported in UniProt, is 9 amino acids long (UniProt ID P07560). The prepared mesh resulted in a geometry with a polygonal count of 212 with a binding site of 9 polygons, figure 3.17 panel A, coloured in blue with binding site in red.

3.3.2.3 Protein models for complex 1JWH

Complex 1JWH is the human protein casein kinase II. The separate subunits for this test case are 3EED and 3C13, for the regulatory subunit (already in a dimeric form) and the catalytic monomer respectively. The PDB files were imported into mMaya for the low poly generation of the surfaces. The obtained geometries were simplified according to subsection 2.4.1. The final surfaces had a poly count of 456 and 230 for 3EED and 3C13 respectively. Each of the two monomers present in 3EED (250 amino acids in length) interact with the catalytic subunit via an area annotated in UniProt (ID P67870) as a sequence of 6 amino acids (figure 3.19 panel A, binding sites coloured in pink). While 3C13 (337 amino acid in length) interacts with each regulatory monomer via a sequence of 6 amino acids (UniProt ID P68400).

After generating the simplified geometries for 3EED and 3C13, using the ePMV extension for Maya, the binding site information was mapped onto each surface. Two binding sites were extracted for 3EED: 13 polygons for chain A and 8 polygons for chain B. This difference in number is due to the different mesh tessellation of the two subunits. The 3C13 binding site is composed of 6 polygons (figure 3.19 panel A, binding sites coloured in blue and yellow).

3.3.3 Analysis of complex 1HE8

Complex 1HE8 is formed by the human GTPases Ras and the human phosphoinositide 3-kinases gamma (PI3Kγ) (Pacold et al., 2000). The PDB for the experimentally solved single subunits are 1E8Z and 821P for PI3Kγ and Ras respectively.

The RMDS was calculated after the superposition and plotted for the 100 complexes as shown in figure 3.16. The mean RMSD value was calculated for a better comparison between test cases. The mean RMSD for the 1HE8 complex is 33.6 Å, with a standard deviation of 8.3 Å.

After superposing all the simulated complexes on their PI3Kγ subunit, the orientation and position of Ras with regards to the kinase is the only parameter that distinguishes complexes. To graphically represent the test case performance, in figure 3.15 panel C all the superposed complexes are shown with an increased transparency of the Ras subunits. The use of transparency allows a better recognition of the areas that are densely populated, while the mesh around it shows the extreme boundaries of the Ras subunit location.

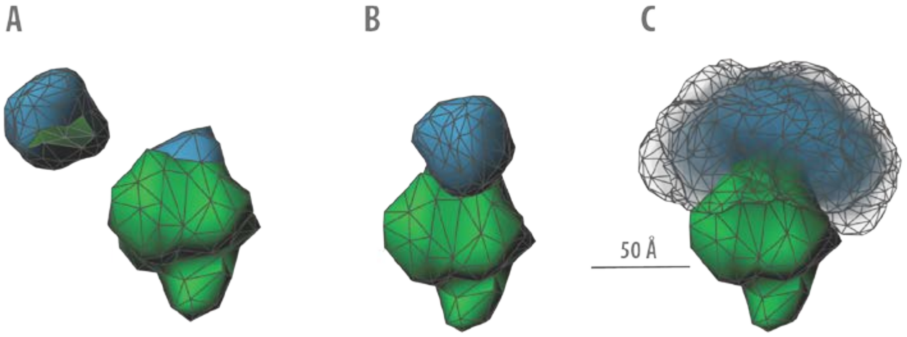


Figure 3.15: Benchmark case 1HE8: geometries. Panel A: separate geometries of Ras represented as a mesh surface in blue, with its binding site for PI3Kγ coloured in green; and PI3Kγ in green with its binding site for Ras colour in blue. Panel B: structure of the complex from 1HE8. Panel C: all the 100 simulated complexes superposed on the PI3Kγ subunits. For a better understanding of the different orientation of Ras on the PI3Kγ binding site, the transparency of Ras was increased so that only highly occupied area will be coloured in blue. The mesh around the high density blue area represents the external boundary of space occupied by Ras.

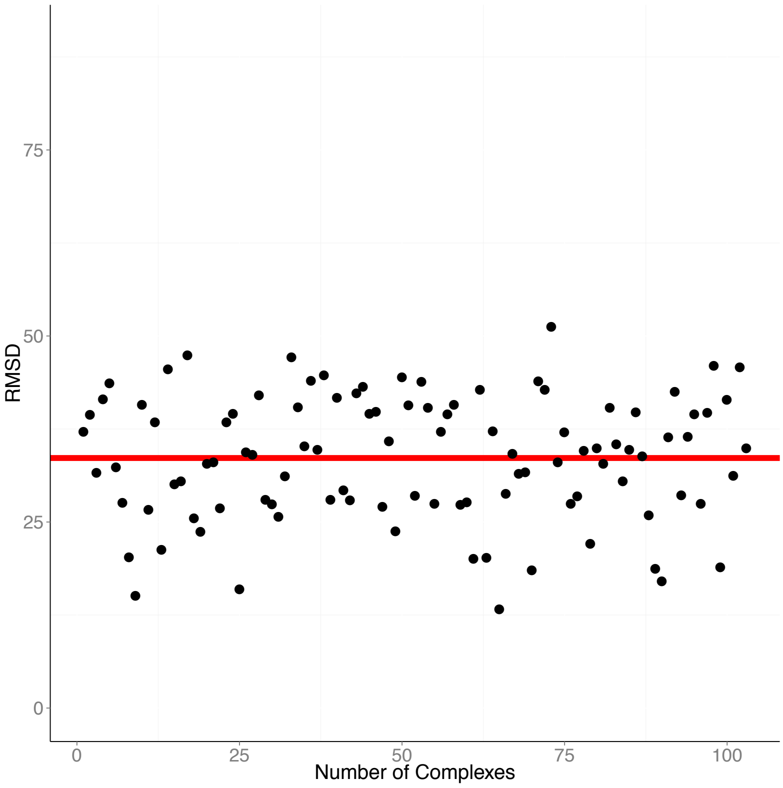


Figure 3.16: Benchmark case 1HE8: RMSD. Each dot represent the RMSD calculated between a simulated complex and the reference structure derived from the PDB 1HE8. The mean RMSD calculated over the 100 complexes is shown as a red line, value 33.6 Åwith a standard deviation of 8.3 Å.

3.3.4 Analysis of complex 3CPH

The 3CPH complex, like 1HE8, is also part of the medium difficulty category of Benchmark 4.0. The complex is composed of the Ras-related protein SEC4, and the Rab GDP-dissociation inhibitor, both from Saccharomyces cerevisiae.

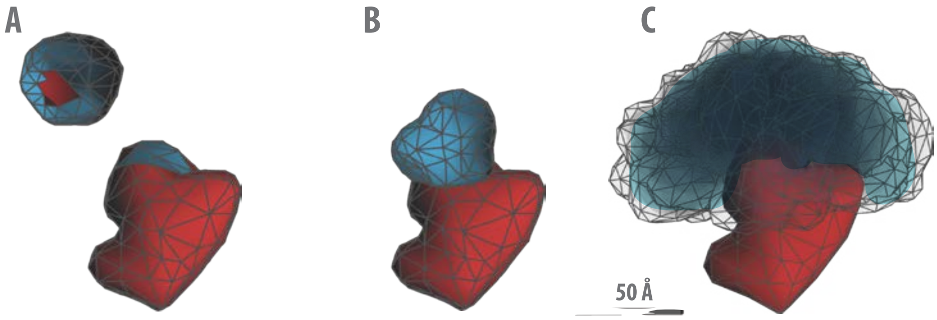


Figure 3.17: Benchmark case 3CPH: geometries. Panel A: separate geometries of Rab in red, and SEC4 in blue. The respective binding sites are coloured as the binding partner. Panel B: complex reference structure obtained from the PDB 3CPH. Panel C: all superimposed simulated structures. The superposition was carried out overlapping the Rab subunits. The transparency of the SEC4 subunits was increased so that only highly occupied area will be coloured in blue. The mesh around the high density blue area represent the external boundary of space occupied by SEC4.

The RMSD calculated for each simulated complex is represented in figure 3.18. The RMSD mean value is 32.4 Å, with a standard deviation of 6.3 Å. A graphical representation of this test case is shown in figure 3.17 panel C, in which the Rab subunit of each complex was superposed, and the variability in the orientation is presented by the position and orientation of the protein SEC4. SEC4 transparency was increased and the outer boundary of SEC4 positions is depicted by the mesh that surrounds it.

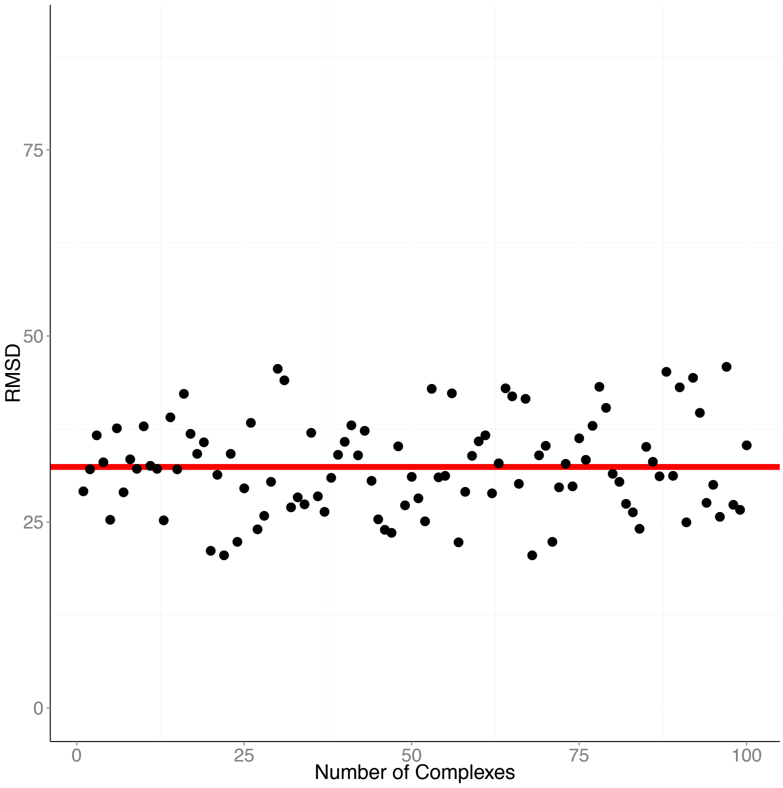


Figure 3.18: Benchmark case 3CPH: RMSD. Each dot represent the RMSD calculated between a simulated complex and the reference structure derived from the PDB 3CPH. The mean RMSD calculated over the 100 complexes is shown as a red line, value 32.4 Åwith a standard deviation of 6.3 Å.

3.3.5 Analysis of complex 1JWH

Complex 1JWH is the human protein casein kinase II. This protein is a holoenzyme formed by two identical regulatory subunits stably linked together (figure 3.19 panel A, coloured in blue and yellow) and two identical catalytic subunits (figure 3.19 panel A, in pink). While all the test cases in Benchmark 4.0 are designed to be only binary, meaning that only two subunits participate in the binding, 1JWH is composed of 4 subunits. According to Benchmark 4.0 only one catalytic subunit should be considered, but I decided to take both subunits into account in order to assess how the pipeline performed in a more complex scenario.

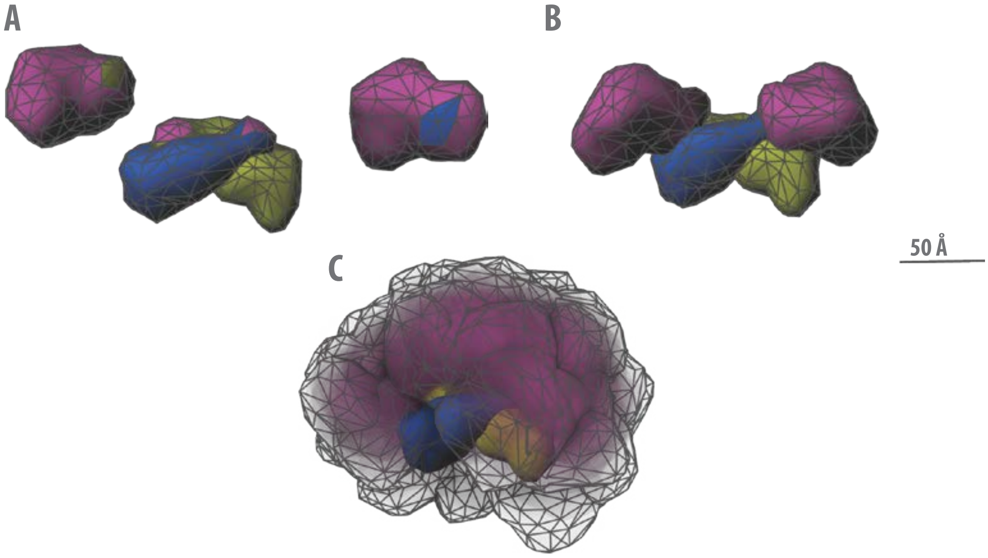


Figure 3.19: Benchmark case 1JWH: geometries. Panel A: separate geometries of the complex 1JWH. The regulatory subunits are represented in the dimeric form as in yellow and blue for chain A and B respectively; their binding site for the catalytic subunits are coloured in pink. The two catalytic subunits are represented in pink and their identical binding sites are coloured in yellow and blue; this was done to emphasise the binding to each of the regulatory subunits, and no distinction was made in the binding rules between the sites. Panel B: complex reference structure obtained from the PDB 1JWH. Panel C: all superimposed simulated structures. The superposition was carried out overlapping the regulatory subunits from 3EED. The transparency of the catalytic subunits, was increased so that only highly occupied area will be coloured in blue. The mesh around the high density blue area represent the external boundary of space occupied by the catalytic subunit.

The RMSD calculated for each simulated complex is presented in figure 3.20, with a mean value of 58.2 *Å* and a standard deviation of 12.8 *Å*

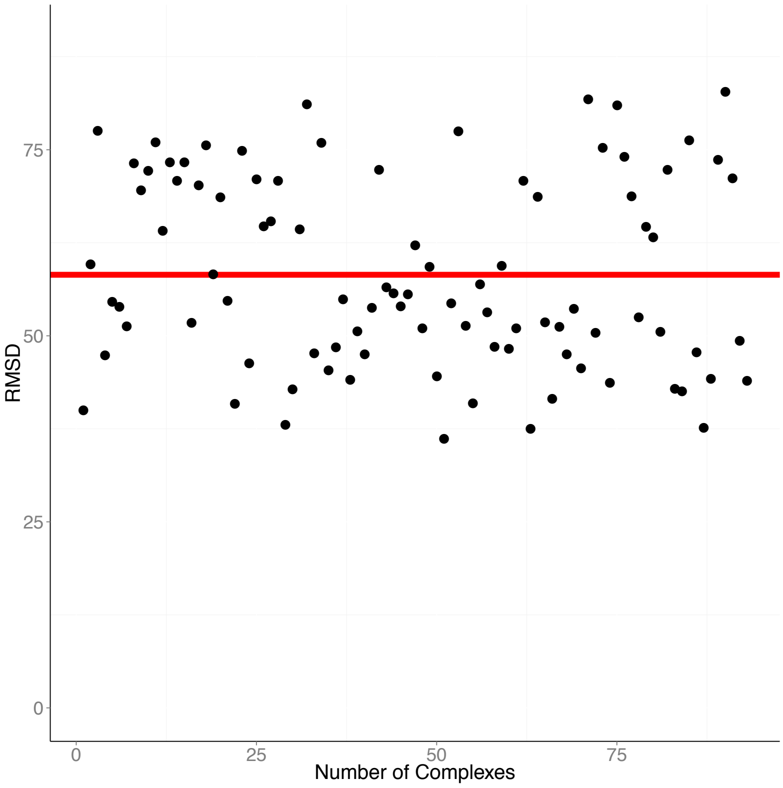


Figure 3.20: Benchmark case 1JWH: RMSD. Each dot represent the RMSD calculated between a simulated complex and the reference structure derived from the PDB 1JWH. The mean RMSD calculated over the 100 complexes is shown as a red line, value 58.2 Åwith a standard deviation of 12.8 Å.

3.4 Discussion

In this chapter I presented two validation sets of experiments for the methodology developed: validation on the diffusion processes and a validation for the complex formation process. Diffusion was validated analysing the medium squared displacement (MSD) calculated from simulations of cytosolic and membrane proteins. The analysis show that in both cases, that the simulated diffusions were accurate and correctly categorised as a random walk. The method still performed accurately, even when non-regular membrane (non regular containers) were used, making the proposed methodology suitable for protein diffusion studies in any arbitrary environments.

Although the validation for complex formation showed that the presented methodology is not as accurate as protein-protein docking software it has to be kept in mind that simplified geometries were used and the binding process is driven by physical contact alone. Given this, the method performed well in producing low resolution complexes. The RMSDs calculated between every vertex of the simulated complex and the reference structure, are in the order of $30\mathring{A}$ for the complex 1HE8 and 3CPH and $60\mathring{A}$ for 1JWH. A resolution of 30 or $60\mathring{A}$ is not as good as any of the classical experimental methods for protein structure identification, like NMR or X-ray crystallography which is less than $5\mathring{A}$[1], but can be compared to the resolution obtained by cryo-electron tomography of large complexes like the nuclear pore, which was $60\mathring{A}$ (Hoelz et al., 2011; Beck et al., 2007).

The use of simplified protein geometries, as proposed in this thesis, alternatively to primitive geometries is supported by the study on the influence of shape on binding kinetics. The study shows that shape has a role in determining how fast a binding reaction is, in a non-intuitive way, influencing not only the number of efficient collisions but also the overall collision events. The most obvious differences in efficient collisions were seen in cases for which the binding site area was defined as less than 15% of the protein surface area (small binding site areas). Since the binding site definition is a manual operation of the presented procedure, one should use the structural information of the binding site where possible, and be aware of its implication in the methodology.

Moreover, the method showed to be in agreement with collision theory, since the

[1] To date (28/10/2015) the mean resolution of protein deposited in the Protein Data Bank according to its statistics is $2.21\mathring{A}$ with a standard deviation of $1.23\mathring{A}$. This data reflects the resolution of the available X-ray and NMR generated protein structures deposited.

simulated collision frequencies are in the same order of magnitude with the one estimated from collision theory.

With these premises the presented methodology has been shown to be a valid method to study protein diffusion, binding interactions and low resolution reconstructions of big assembly, where other available methodologies are too computationally expensive. In the following chapter I will present the application of the method to the study of the large post-synaptic density assembly.

Chapter 4

Reconstruction of a minimal PSD assembly

4.1 Introduction

> I'm watching a dream I'll never wake up from.
>
> *Spike Spiegel*
> Cowboy Bebop

In chapter 1, I presented the importance of dendritic spines and the post-synaptic density (PSD) for processing neuronal information. Dendritic spines are small specialised compartments on the dendritic arborization of neurons. The PSD is a proteinaceous organelle present on the membrane of excitatory spines, in direct opposition to the neurotransmitters release site. The PSD is attached to the postsynaptic membrane and held in place by actin filaments coming from the cytosol of the dendritic spine (Sheng and Kim, 2000). It is composed of glutamate ion channels, scaffolding and signalling proteins. One of the main suggested functions of the PSD is to hold the membrane receptors in place and connect them to their effectors, thereby creating a hub for the post-synaptic signalling. Moreover, the size and composition of the PSD can vary with time and stimulation pattern. Synapses that give a stronger response upon the same stimulus contain more glutamate receptors and show larger PSDs.

It is known that the PSD has a layered structure (section 1.3), but its exact composition is still unknown: experiments report very different compositions. However, a common core of more than 400 different protein types was identified (Collins et al., 2006). This set contains some of the more important and well studied proteins: the glutamate receptors AMPAR and NMDAR, scaffolding proteins of the MAGUK, the Shank and the SAPAP family, the scaffolding protein Homer, the highly abundant signalling protein CaMKII, and the auxiliary TARP family protein, Stargazin (Collins et al., 2006). These proteins are thought to be highly important for PSD function and structure but so far there are no successful functional reconstructions. Moreover, it would be interesting to test if it is possible to form a PSD assembly by only accounting for the protein geometries and their binding networks. The formation of such a minimal assembly would suggest that there are no other mechanisms behind the PSD formation apart from the binding network encoded in the protein structure and function. In the PSD is a self-organised structure, it would be interesting to assess whether every protein in the assembly has the

same importance, or if there are the proteins that are more important for the network organisation than others.
For example, knockdown of the membrane associated guanylate kinase MAGUK protein PSD–95 alone results in a discontinuous PSDs (Chen et al., 2011).

In order to answer those questions I decided to apply the methodology proposed in chapter 2 to simulate the PSD assembly formation using a set of 13 proteins that are the most abundant and biologically relevant. This set is composed of: the calcium/calmodulin dependent protein kinase CaMKII, the membrane associated guanylate kinase MAGUK protein PSD–95, the four isoforms of the GKAP/SAPAP family, the three isoforms of the ProSAP/Shank family, the NMDA and AMPA receptors and the TARP protein Stargazin.

I am going to discuss the role of those 13 proteins in the PSD assembly in more detail in the present chapter. Afterwards, I am going to describe the simulation setups as well as their challenges. Lastly, I am going to present and then discuss the results of the PSD assembly formation simulations.

4.1.1 The NMDA and AMPA receptors

The NMDA and AMPA receptors are part of the large family of excitatory ionotropic glutamate receptors (iGluRs). The NMDAR possesses a uniquely high permeability to calcium ions which grants it a central role in synaptic plasticity.
NMDARs are hetero-tetramers with a very heterogeneous composition. NMDAR subunits are categorised in three subtypes: NR1, NR2, and NR3. There are eight different NR1 subunits generated by alternative splicing from a single gene, four different NR2 subunits (A, B, C and D) and two NR3 subunits (A and B), encoded by different genes (Dingledine et al., 1999). Tetramers are commonly formed by a combination of NR1 and NR2 subunits of the same or different subtypes. In cells expressing the NR3 subunits the ternary complex is often formed by a combination of NR1, NR2, and NR3 subunits (Sasaki et al., 2002). The subunit composition of NMDAR seems to depend on both: brain areas and developmental stages. One of most common form of NMDAR, found in the hippocampus, consists of NR1 (A, C and E splicing variants) and NR2 (A and B subtypes) (Ehlers et al., 1996). The NR1 subunits possess binding sites for the calcium binding protein calmodulin and a PDZ binding domain that can interact with scaffolding proteins like

PSD–95. The NR2B subunits possesses binding sites for CaMKII, which, together with calmodulin, influence the conductance and opening time of the receptor (Paoletti and Neyton, 2007; Gielen et al., 2009).
The NMDAR activation requires the depolarisation of the membrane to release of the magnesium block and the binding of glutamate. Its precise activation mechanism and regulation are still debated, but it is clear that different subunits and subtypes affect both the kinetics and the regulation of the receptor (Foster et al., 2010; Cull-Candy and Leszkiewicz, 2004).

AMPA receptors are also hetero-tetrameric proteins. Their subunits are divided into four types, GluR1 to GluR4 (Hollmann and Heinemann, 1994). The most common AMPARs in the adult hippocampus are the ones formed by GluR1 and GluR2 or by GluR2 and GluR3 (Wenthold et al., 1996). The cytoplasmic tails of the AMPA receptors can interact with the PDZ domains of scaffolding proteins like PSD–95 via proteins from the transmembrane AMPAR regulatory proteins (TARP) family (Bats et al., 2007; Sumioka et al., 2010; Bats et al., 2012) and Shank proteins (Uchino et al., 2006).
AMPA receptors have faster kinetics and are permeable to sodium and potassium ions in contrast to the NMDAR (Malinow and Malenka, 2002; Bassani et al., 2013). AMPARs are regulated, among other means, through phosphorylations mediated by CaMKII and PKA. AMPAR changes its conductance and or trafficking depending on the specific phosphorylated residue (Kennedy et al., 2005; Opazo et al., 2010).

4.1.2 The MAGUK protein PSD–95

PSD–95 was one of the first proteins identified in the PSD due to its high abundance (Cho et al., 1992). It is a member of a large class of scaffolding proteins, that are present in the PSD, with SAPAP, Shank, and Homer. PSD–95 proteins are composed of three PDZ domains, a consensus binding motif of 90 amino acids that is found in several membrane-associated proteins (Kim and Sheng, 2004; Lee and Zheng, 2010). Via these domains, PSD–95 interacts with the cytoplasmic tails of NMARs, TARPs, and many other proteins such as shaker-type voltage-gated potassium channels (Kim et al., 1995b), several inwardly rectifying potassium channels (Nehring et al., 2000), the receptor tyrosine kinase ErbB4 (Garcia et al., 2000) and semaphorin adhesion proteins (Burkhardt et al., 2005).

One of the most important roles of PSD–95 is to place calcium activated signalling enzymes in the proximity of NMAD receptors, similar to the other scaffolding proteins. This creates a subcompartment in which proteins are spatially and temporally restricted (Kennedy, 2000). PSD–95 can also bind to the neuronal nitric oxide synthase (nNOS) and to synGAP, a Ras-GTPase activating protein commonly found in the PSD (Brenman et al., 1996; Kim et al., 1998). At its C-terminal, PSD–95 has a Guanylate Kinase-like (GK) Domain with which it interacts with the GK-associated proteins (GKAPs or SAPAPs) (Boeckers et al., 1999). Due to this interaction PSD–95 links another important scaffolding protein family to the membrane, the Shank family which is known to interact with the SAPAP family (Naisbitt et al., 1999).

PSD–95 is post-translationally modified by PSD–95 palmitoyl transferases, which palmitoylates PSD–95 at cysteine residues 3 and 5 (Fukata et al., 2004). The palmitoylation allows PSD–95 to be localised at the membrane, which is critical for its interaction with NMDA and AMPA receptors (El-Husseini et al., 2002).

4.1.3 The SAPAP family

The SAPAP (SAP90/PSD–95-Associated Proteins) protein family, also known as GKAP (Guanylate Kinase-Associated Protein) family is another scaffolding protein family. As the name suggest, they were discovered due to their interaction with the guanylate kinase-like (GK) domain of the PSD–95 protein (Kim et al., 1997). The GK domain in PSD–95 has supposedly lost its catalytic function and gained a protein-protein binding function, which SAPAPs recognise via their GK binding domain. There are four different isoforms in the SAPAP family, named SAPAP1 to SAPAP4. All isoforms are expressed in hippocampal neurons (Takeuchi et al., 1997) and consist of a GK binding domain and a PDZ binding domain. Their length varies between 1,054 for SAPAP2 and 977 amino acid for SAPAP1. SAPAPs interact with the PDZ domains of Shank proteins, via their PDZ binding domains (Sheng and Kim, 2000; Naisbitt et al., 1999). Therefore, SAPAPs connect the membrane receptors-PSD–95 complexes with the Shank family members.

4.1.4 The Shank family

The Shank family is another abundant family of scaffolding proteins in the PSD. It consists of three isoforms Shank1, Shank2 and Shank3. The full length of Shank is characterised by multiple ankyrin (ANK) repeats, an SH3 domain, a PDZ domain, a long proline-rich region, and a sterile alpha motif (SAM) domain (Naisbitt et al., 1999). Their length varies between 2161 amino acid for SHANK1 and 1470 of SHANK2, which does not contain the 6 ANK repeats. Shanks are able to form dimers tail-to-tail via their SAM domains, leading to the formation of a two-dimensional array (Baron et al., 2006). Shanks are also able to bind another scaffolding protein called Homer via their proline-rich regions (Ehlers, 1999).

Immunogold electron microscopy analysis revealed that the majority of Shank proteins are located deep in the PSD at around 30 nm from the membrane in contrast to PSD–95 that is in direct contact with it (Valtschanoff and Weinberg, 2001). Therefore, Shanks are not considered to function as direct scaffold for the membrane receptors but as organisers of the network of scaffolding proteins composed of SAPAPs and Homer (Kreienkamp, 2008).

4.1.5 The homotetramer Homer

Homer is a synaptic scaffolding protein that possesses an EVH1 domain at its N-term and a coiled-coil domain at its C-term (Xiao et al., 2000). Its family is composed of three members alternatively spliced into a long and a short form. The long forms contain both the EVH1 and the coiled-coil domains and are constitutively expressed, while the short forms contain only the EVH1 domain (Brakeman et al., 1997). The EVH1 domain of Homer binds to various proteins including metabotropic glutamate receptors (mGluRs), IP3 receptors (IP3Rs) (Tu et al., 1998), and Shanks (Tu et al., 1999). The coiled-coil domain allows for tetramerisation, the form of Homer that is able to bind Shanks (Hayashi et al., 2009). This tetrameric form of Homer with its four EVH1 domains is capable of connecting proteins on the plasma membrane, such as mGluRs, with proteins in the PSD, such as Shanks, or intracellular organelles, such as IP3 receptors expressed on the endoplasmic reticulum.

4.1.6 The calcium/calmodulin-dependent protein kinase II

The calcium/calmodulin-dependent protein kinase II (CaMKII) is possibly the best studied kinase in the dendritic spine. CaMKII is a serine/threonine protein kinase that plays a crucial role in synaptic plasticity. It drives the phosphorylation and the insertion of new AMPA receptors into the post-synaptic membrane resulting in an increase of synaptic strength and promotion of LTP (Pi et al., 2010). CaMKII has a highly complex structure: it is formed by 12 subunits held together via their C-terminal. The subunits are randomly composed by alpha and beta isoforms. It is activated by interaction with calmodulin and can also auto-phosphorylate which stabilises its active form (Colbran and Brown, 2004).

About 1% of the total PSD protein mass is composed of CaMKII making it one of the most abundant proteins in the PSD assembly (Erondu and Kennedy, 1985). The holoenzyme can phosphorylate several components of the PSD such as the tails of AMPAR, the TARP protein Stargazin, and the NR1 and NR2 tails of NMDAR (Griffith et al., 2003). CaMKII can stably bind as well as phosphorylate NR2B. Phosphorylation of NR2B at the residue Ser1303 appears to negatively regulate the association between CaMKII with NR2B, promoting its dissociation. Moreover, this interaction seems to promote the desensitisation of the receptor (Sessoms-Sikes et al., 2005). Interestingly, CaMKII can stably bind F-actin, and it was shown that the presence of beta-CaMKII can reorganise actin filaments into bundles (Sanabria et al., 2009).

4.1.7 The transmembrane auxiliary subunits of AMPARs

The transmembrane auxiliary subunits of AMPARs (TARPs) are a family of proteins that influence AMPAR functions. The family is composed of Stargazin (also known as $\gamma 2$) and three other proteins known as $\gamma 3$, $\gamma 4$, and $\gamma 8$ (Nicoll et al., 2006).

Stargazin binds to the AMPA receptors and influences their trafficking, targeting, and biophysical properties (Sumioka et al., 2010). TARPs also anchor AMPAR to the PSD–95 protein network due to the ability of TARPs to bind PSD–95 to its PDZ domains (Fukata et al., 2005). This interaction is also required for the activity-dependent translocation of AMPARs from the extra-synaptic space to the PSD (Bats et al., 2007).

The number of post-synaptic AMPARs decreases dramatically when the PSD–95 localisation is disrupted. On the other hand, PSD–95 over-expression increases the number of AMPARs resulting in an enhancement of AMPA receptor-mediated synaptic transmission that occludes LTP and increases the amplitude of LTD (Stein et al., 2003).

4.2 Methods

4.2.1 Protein models preparation

Protein geometries were modelled based on structural data following the procedures in section 2.4. The only available full-length structure in the Protein Data Bank (PBD) at the time this work was performed was CaMKII. The structures of remaining proteins were obtained using a homology modelling approach.

The homology modelling procedure presented in section 2.4 was performed for the proteins Shank1–3, SAPAP1–4, PSD–95 and the TARP protein Stargazin. NMDA and AMPA receptor tails were modelled as described in section 3.1.2. The tetrameric full-length structure of the long form of Homer was kindly provided by Mariko Hayashi (Hayashi et al., 2009). CaMKII was modelled using the PDB 3SOA. The resulting 3D geometries are presented in figure 4.1.

Binding sites were defined using data taken from the literature and the UniProt database. The entire surface of the glutamate receptors was defined as binding site due to the lack of precise structural informations. The entire surface of the protein Stargazin was defined as a binding site, due to the lack of information regarding the binding site involved in the binding of AMPAR and PSD–95.

4.2.2 Encoded binding network

The protein-protein binding network was encoded in the simulations using the binding rules as explained in ??. Data about binding partners and binding sites was taken from the literature. The rules are mono-directional which implies that for each binding process one of the two proteins carries the binding rule while the other functions as a target. Table 4.1 specifies the interacting binding sites while figure 4.2 represents the protein network.

Binding event outputs were saved for each simulation. The binding sites involved in the binding event as well the protein ids were recorded for every binding event in the binding output. The information was used to build a network of interactions in which each cluster represents an assembly with each node representing a specific protein. The networks were analysed in Cytoscape (Shannon et al., 2003), a platform for visualisation and analysis of complex networks with the plugin NetworkAnalyzer v.2.7 (Doncheva

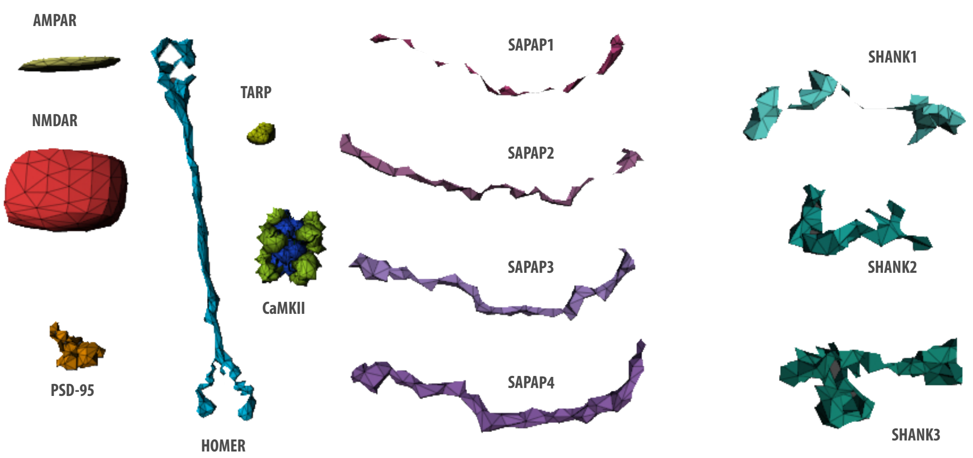


Figure 4.1: Protein models used in the PSD assembly simulations: the geometries have the following polygonal counts: AMPAR 251; NMDAR 246; PSD95 368; SAPAP1 244; SAPAP2 228; SAPAP3 238; SAPAP4 236; Shank1 272; Shank2 330; Shank3 274; TARP 250; Homer 994; CaMKII 2688.

et al., 2012).

The analysis of the protein composition was performed using the statistical package R 3.0.0 (R Development Core Team, 2013) and the related plots were produced using the same software program.

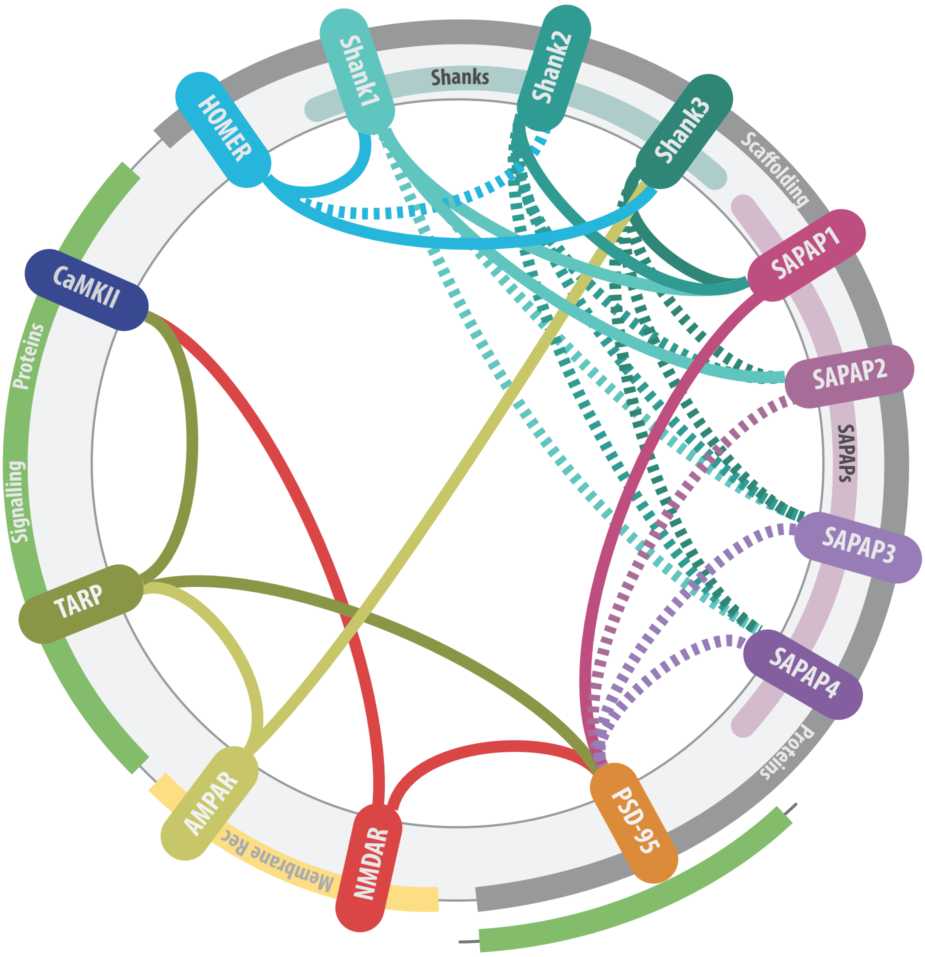


Figure 4.2: Encoded binding network for the PSD assembly simulations: bindings were encoded unidirectionally. The colour scheme indicates the proteins that hold the binding rule. For example, the blue line that links Homer and Shank1 implies that Homer holds the binding rule, while Shank1 is treated as a target. Solid lines indicate experimental evidence of the binding between the proteins, while dashed lines indicate interactions that were inferred by isoform similarity.

Table 4.1: Binding network for the PSD assembly simulations

Protein A	Protein B	Reference
(Binding site with rule)	(Binding site target)	
EVH1 Homer	Proline-Rich Shanks	(Hayashi et al., 2009)
PDZ Shanks	C-term SAPAPs	(Naisbitt et al., 1999)
N-term SAPAPs	GK PSD-95	(Boeckers et al., 1999)
NMDAR	PDZ PSD-95	(Lim et al., 2002)
NMDAR	CaMKII	(Paoletti and Neyton, 2007)
TARP	PDZ PSD-95	(Fukata et al., 2005)
TARP	CaMKII	(Griffith et al., 2003)
AMPAR	PDZ Shanks	(Uchino et al., 2006)
SAM Shanks	SAM Shanks	(Baron et al., 2006)

4.3 Results

4.3.1 Simulation setups

Two main simulation setups were created in order to study the PSD assembly: one mimicking the physiological conditions of a dendritic spine and the second representing a PSD–95 knockdown experiment. A sphere with a volume of 0.01 μm^3 was used as a container and the simulations were run for 1600 msec for both setups. The molecule numbers used for each protein under physiological conditions are reported in table 4.2. The numbers of PSD–95 proteins in the PSD–95 knockdown experiment was decreased to 25. Three replicates were run for both setups.

4.3.2 Technical constraints

Two additional proteins were originally considered for this study: the Synaptic Ras GTPase-activating protein 1 SynGAP, and the Multi-PDZ Domain Protein 1 MUPP1. Both proteins are highly expressed in the PSD and can interact with most of the selected proteins. On the other hand, both proteins are highly unstructured and the homology modelling procedure failed to produce trustworthy results. For the lack of structural data the two proteins were left out of the simulations.

The simulations were run with a reduced system due to technical limitations(see 4.4 for details). Such simulations can still be considered as valid since not only the copy number of each component was reduced, but also the container volume in a ratio calculated to maintain the original components concentration. Protein numbers as well as the protein mass used as input for the diffusion coefficients calculations are reported in table 4.2.

Table 4.2: Binding network for the PSD assembly simulations

Protein	Mass (Da)	Concentration (protein number)
NDMAR	543480	8
AMPAR	400654	25
PSD-95	80495	75
TARP	35966	20
SAPAP1	108873	10
SAPAP2	117620	10
SAPAP3	106040	10
SAPAP3	108012	9
Shank1	224959	12
Shank2	158822	12
Shank3	186295	12
Homer	161104	15
CaMKII	649056	30

4.3.3 Analysis of the simulated PSD assembly

The binding outputs from the simulations were used to analyse the assembly network. The analysis showed that assemblies of different sizes were formed for all replicates. This evidence alone suggests that the formation of the PSD is driven by the interactions encoded in the binding network itself and in the geometry and diffusion properties of the considered proteins.

Overall, the networks are composed of several clusters of different sizes. The binding output does not include single proteins, since it registers binding events during the course of the simulations. Therefore, the smallest clusters are composed of two proteins. The presence of smaller clusters suggests that bigger assemblies grow gradually over time, incorporating more proteins (figure 4.3). These binary interactions involve PSD–95 and its partners for around 70% of the total binary clusters. Moreover, bigger clusters always contain PSD–95 and Homer proteins, as well as proteins from the Shank and SAPAP family. Notably, the glutamate receptors are almost absent from the bigger networks. This suggests that the receptors are not the initialisation points for the assembly formation, but they are probably trapped in the assembly later in time or when it reaches a larger size.

CaMKII is also almost always absent from the big clusters. This could be an artifact due to the low molecule number present in the system compared to the physiological one.

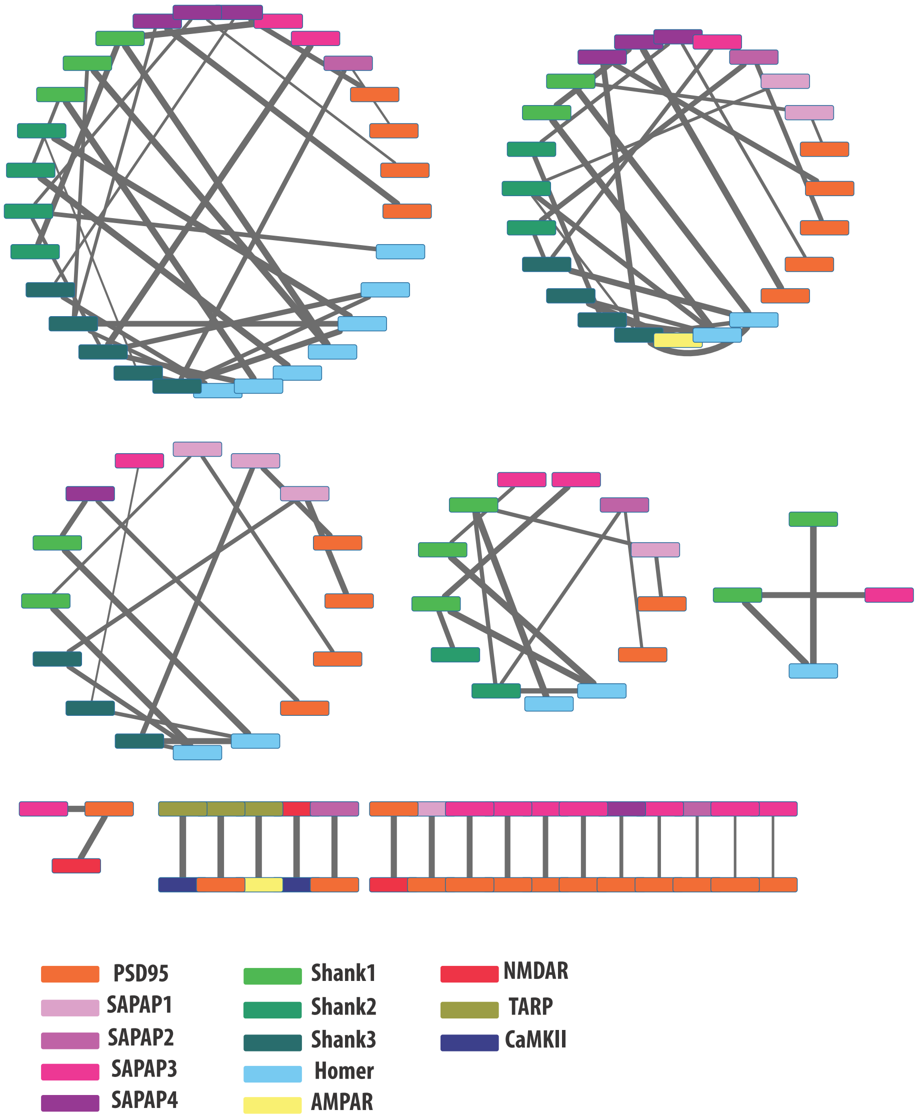


Figure 4.3: PSD assembly network composition: layout organised by species name. The thickness of the line indicates the time course of the binding events: events that happened in early stages of the simulation are represented with thick lines, while towards the end of the simulation with thin lines. The colour scheme is represented at the bottom of the figure.

The networks were analysed in Cytoscape in order to understand if all the proteins in the networks contributed equally to the assembly formations. In particular the *betweeness centrality* parameter was used to discriminate network hubs. This parameter ranks the nodes according to their position in the network indicating the amount of control that this node exerts over the interactions of other nodes in the network (Brandes, 2001). The *betweeness centrality* was used in figure 4.4 as a scaling factor for the node diameters, in which the same network of figure 4.3 is presented in a hierarchical layout. The hierarchical representation of the network highlights a layered structure in which it is possible to distinguish between Homer, Shank and SAPAP proteins, as well as PSD–95. The structure of the simulated networks shows that Shank and SAPAP proteins are often in separate yet closely related layers, while Homer is always present at the top layer. This structure, which is a direct consequence of the encoded binding network reflects the layered nature of the PSD (see figure 1.7 from section 1.3). Starting from the membrane, PSD–95 and the membrane receptors form the first layer of proteins in the PSD. The second and third layer are in close proximity with each other, and are likely formed by SAPAP proteins, Shanks and/or possibly Homer (Sheng and Kim, 2011).

The analysis of the centrality of the proteins present in the clusters revealed that not all the proteins contribute equally to the regulation of the assembly structure in the resulting model. Homer and Shank3 are hubs of the PSD assembly, with the higher values of the *betweeness centrality* parameter. While Homer was previously suggested to be an important component of the PSD (Xiao et al., 2000; Brakeman et al., 1997; Hayashi et al., 2006), our results propose for the first time that Shank3 has a pivotal role in the assembly.

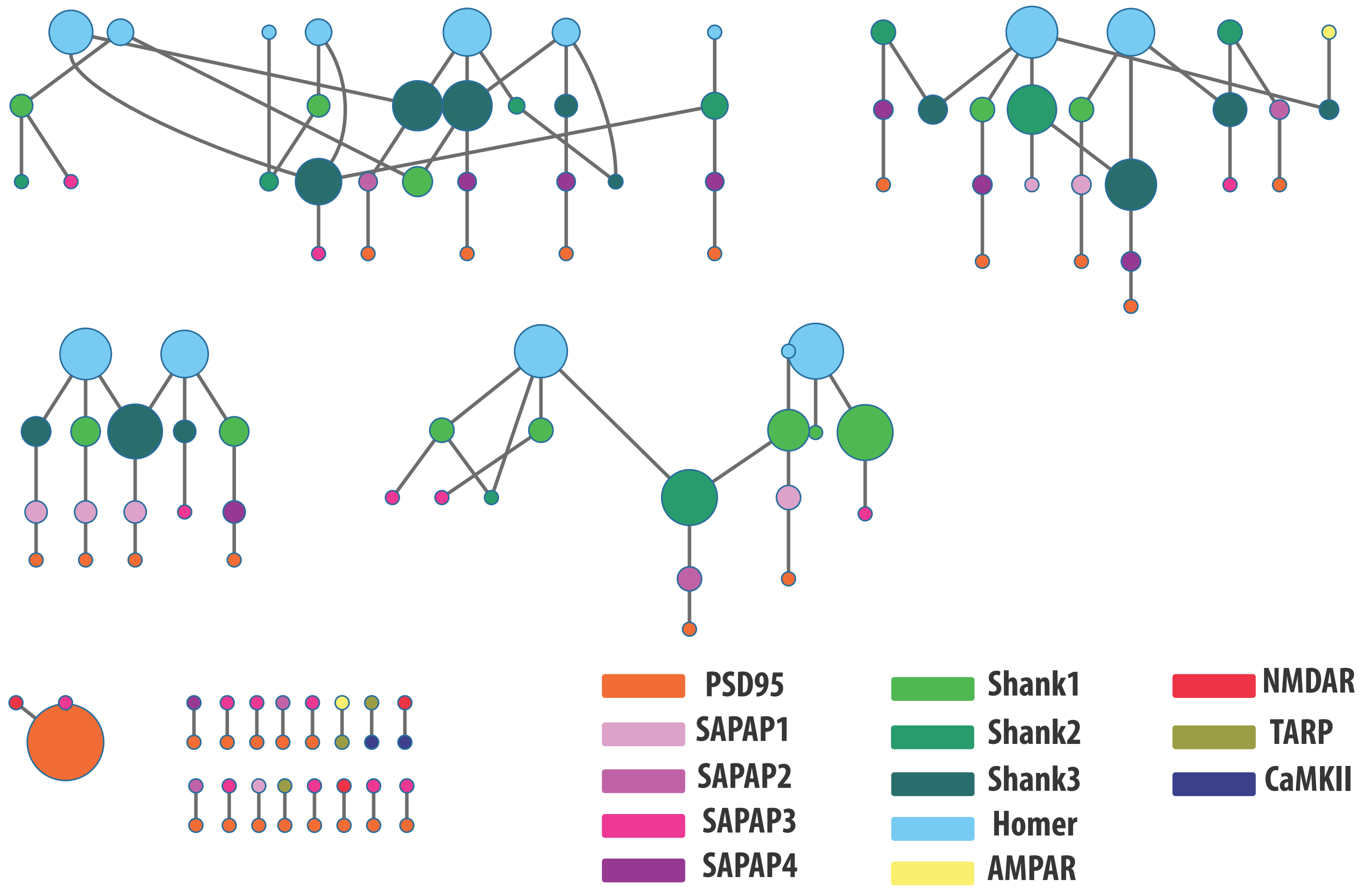


Figure 4.4: Hierarchical representation of the PSD assembly network: the example network from figure 4.3 is presented with a hierarchical layout. The radius of each node is proportional to the betweeness centrality parameter calculated with Cytoscape. Colour scheme is reported at the bottom of the figure.

Relative protein abundances were calculated for each cluster as the molecule number divided by the cluster size to analyse if clusters of different sizes have different protein compositions. For this analysis only clusters containing a number of nodes higher than four (cluster size > 4) were considered and the Shank and SAPAP isoforms were grouped into their respective families. The analysis revealed that the relative protein composition remains constant with the increase of cluster size (figure 4.5). Moreover, the mean values of the relative protein abundance, as reported in figure 4.6 shows that on average the PSD clusters are composed on average of 1:1 ratio between PSD–95 and SAPAP isoforms as well as 1:1 ratio between Homer and Shanks while Shanks are in a 2:1 ratio with PSD95.

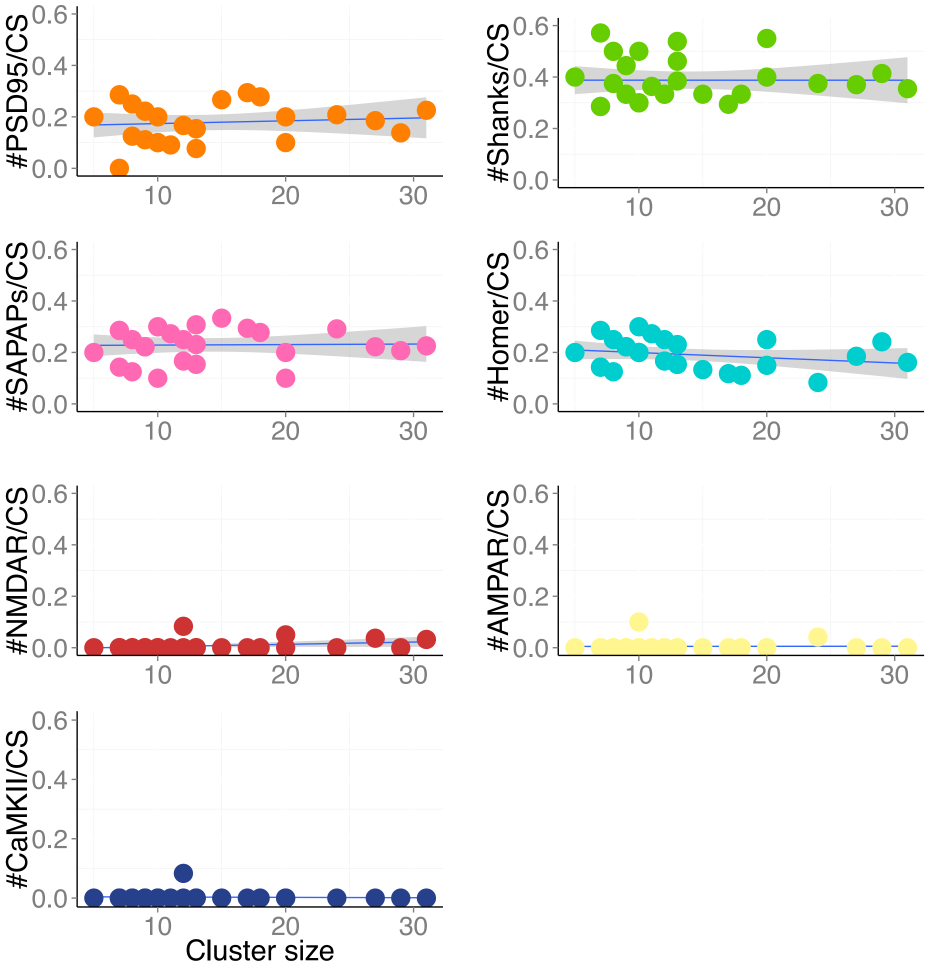


Figure 4.5: Relative abundance of proteins in clusters: for clusters of size higher than 4, the relative protein abundances were calculated, for all the replicates, as the molecule number counted in the cluster divided by the cluster size (CS). The different isoforms of Shank and SAPAP were grouped into a single plot to represent the protein family.

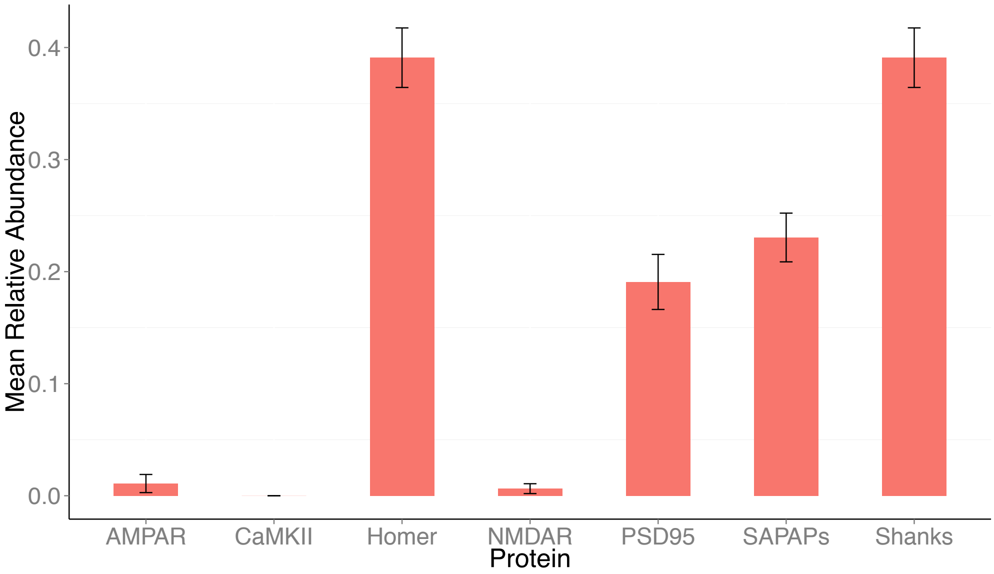


Figure 4.6: Mean protein abundance in the PSD networks: bar plot of the mean relative protein abundance calculated for all the clusters of size higher than four in all the replicates. Error bars on the mean are reported on each bar and represent the standard deviation.

Further analysis on the protein composition showed that there is a tendency for Shank isoforms to be more present in clusters in which PSD95 is less occurring and vice versa, as shown in figure 4.7. This inverse correlation can possibly suggest a *steric competition* between PSD95 and Shank isoforms. Both proteins interact with SAPAP isoforms albeit on different binding sites. It is possible that the binding of PSD95 on SAPAPs prevents Shanks from joining the assembly. Such a scenario leads to a competitive behaviour.

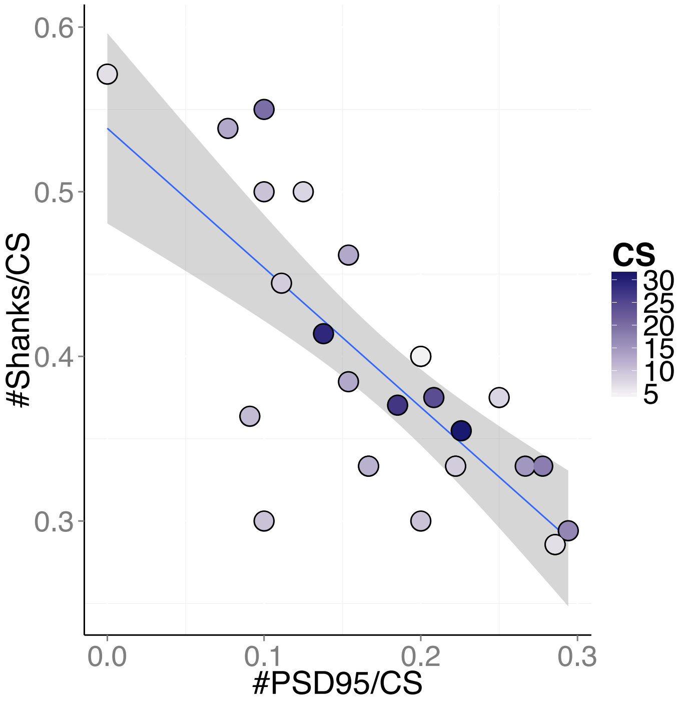


Figure 4.7: Inverse relationship of the quantity of PSD-95 and Shanks incorporated in the PSD assembly: the relative abundance of PSD95 is plotted against the relative abundance of the Shanks isoforms for clusters with size higher than four. Each point in the plot represents a cluster, and its size (CS) is represented with a gradient from light grey for cluster size of 5, to dark blue for a cluster size of 31. Linear correlation R=0.92

4.3.4 Analysis of PSD–95 knockdown simulations

To better understand the structural role of PSD–95 in the formation of the PSD assembly, the molecule number of PSD–95 was lowered to 25 in order to reproduce a knockdown condition (Chen et al., 2011). As for the previous simulations that were mimicking the physiological conditions, protein assemblies were produced during the simulations. An example of one of the output networks is reported in figure 4.8. The clusters produced are visibly smaller than the one obtained in the wild-type conditions. All the produced clusters from both simulation conditions were counted and classified by their cluster size to have a more comprehensive overview of the network topology. The analysis, reported in figure 4.9, confirmed that the produced clusters were of smaller size compared to the wild-type when the presence of PSD–95 was reduced. The majority of the larger clusters produced in the knockdown condition have a size between 5 and 13. This evidence remarks the importance of PSD–95 in the PSD assembly formation and is in agreement with the experimental evidence that reports a patchy loss of PSD in case of PSD–95 knockdown (Chen et al., 2011). Interestingly, the number of clusters that have only two components are also reduced in the knockdown condition. This can be linked to the high number of PSD–95 binary interactions noticed for the wild-type simulations.

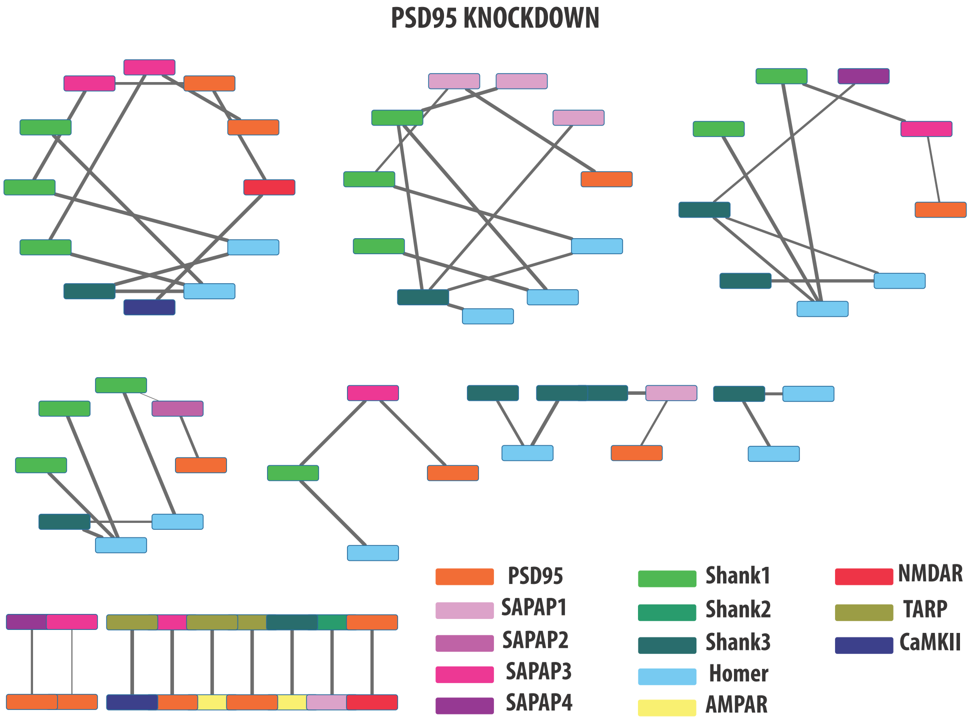


Figure 4.8: PSD-95 knockdown network composition: circular representation of the network organised by species name. Line thickness of the connections is inversely proportional to the time course of the simulation. Colour scheme reported at the bottom of the figure.

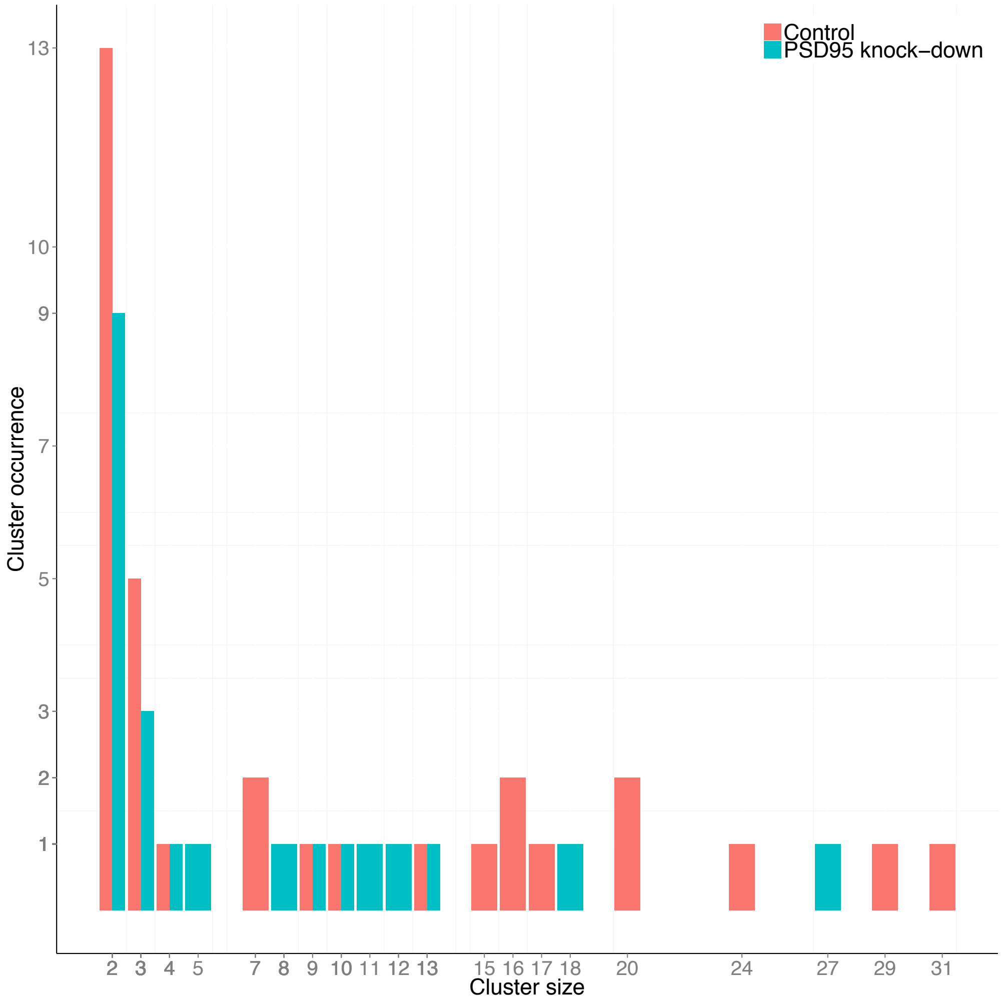


Figure 4.9: Network topology comparison between control and PSD-95 knockdown setups: clusters were counted for all the replicates in both conditions, and classified according to their cluster size. The total number of occurring clusters is reported for the wild-type condition in pink, and the PSD-95 knockdown condition in blue.

Further analysis of the network topology showed a change in the overall connectivity of the network. A parameter that quantifies the connectivity of a network is the *connected component parameter*. In undirected networks, such as the PSD assembly, two nodes are considered connected when they share an edge. All the pairwise connected nodes form a *connected component* within a network. The mean connected component for wild-type simulations and PSD knockdown is presented as a box plot in figure 4.10. Knocking down PSD–95 has the effect of decreasing the connected components of the network, therefore decreasing the overall connectivity. This suggests that PSD–95 has an important role in the overall topology of the PSD assembly in agreement with the experimental evidence (Xu, 2011; Chen et al., 2011).

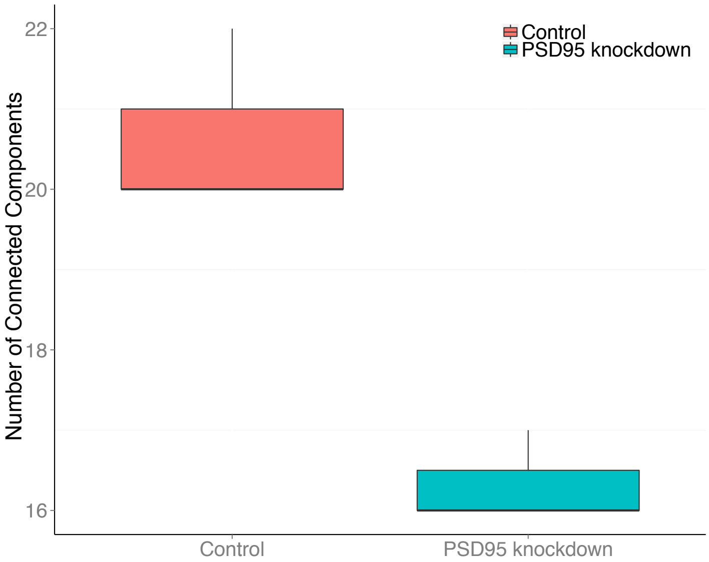


Figure 4.10: Comparison in the network connectivity between wild-type and PSD-95 simulations: box plot representation of the connected components parameter obtained for the replicates in both conditions. The parameter was calculated with NetworkAnalyzer (Doncheva et al., 2012). Statistical difference between the results confirmed by Wilcoxon test p-value 0.02

From the analysis of the networks produced in the PSD–95 knockdown simulations, it is interesting to notice that the layered structure of the clusters remains unchanged. The clusters showed the same hierarchical layout as for the wild-type despite the decrease in size as shown in the example reported in figure 4.11. Isoforms from the SAPAP family as well as Shank family, are always present in the larger clusters together with Homer. Homer and Shank3 still maintain the role of main hubs for these smaller assemblies. The glutamate receptors and CaMKII are still present in very low amounts as for the wild-type simulations.

Analysis of the cluster compositions is reported in figure 4.12. The plot shows the relative abundance of different proteins in the larger clusters for the wild-type and PSD–95 knockdown simulations. Clusters of size below four were not considered in the composition analysis, and Shank and SAPAP isoforms were grouped into their relative families. The analysis shows that as for the wild-type, the relative protein composition is not dependent on the cluster size. Moreover, both conditions share the same relative abundance and protein ratio (see figure 4.6).

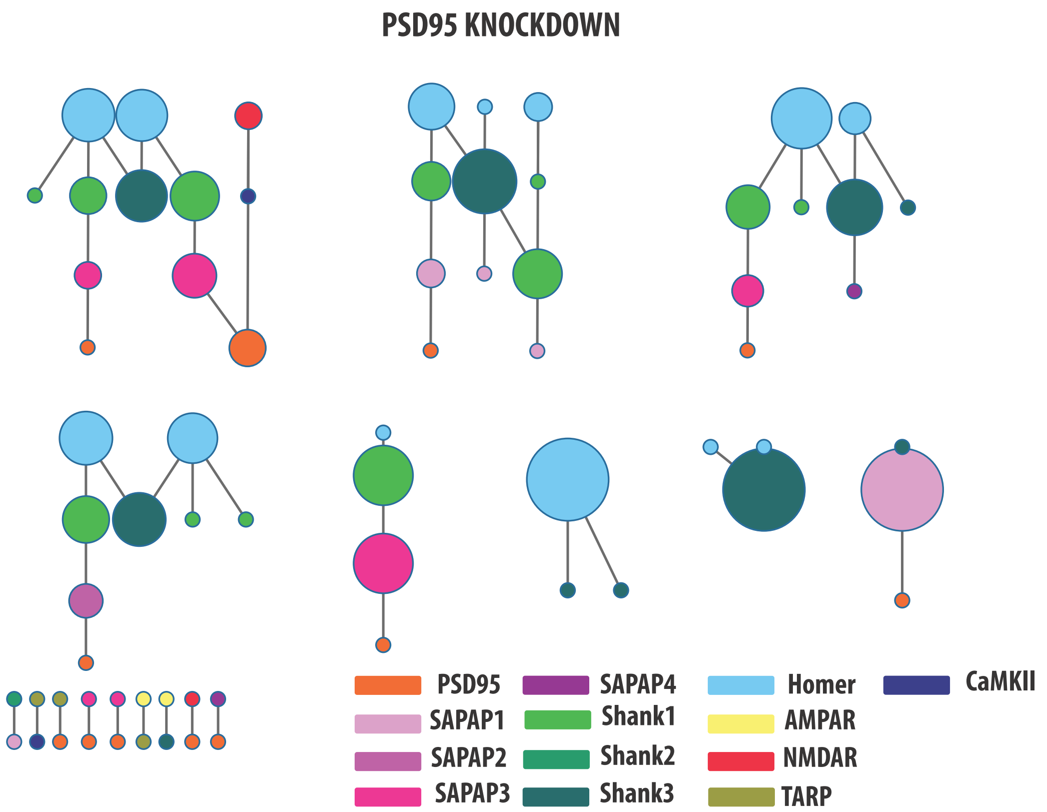


Figure 4.11: PSD-95 knockdown network hierarchical representation: hirarchical representation of the network in figure 4.8. The radius of each node was scaled by the *betweeness centrality* parameter. Colour scheme reported at the bottom of the figure.

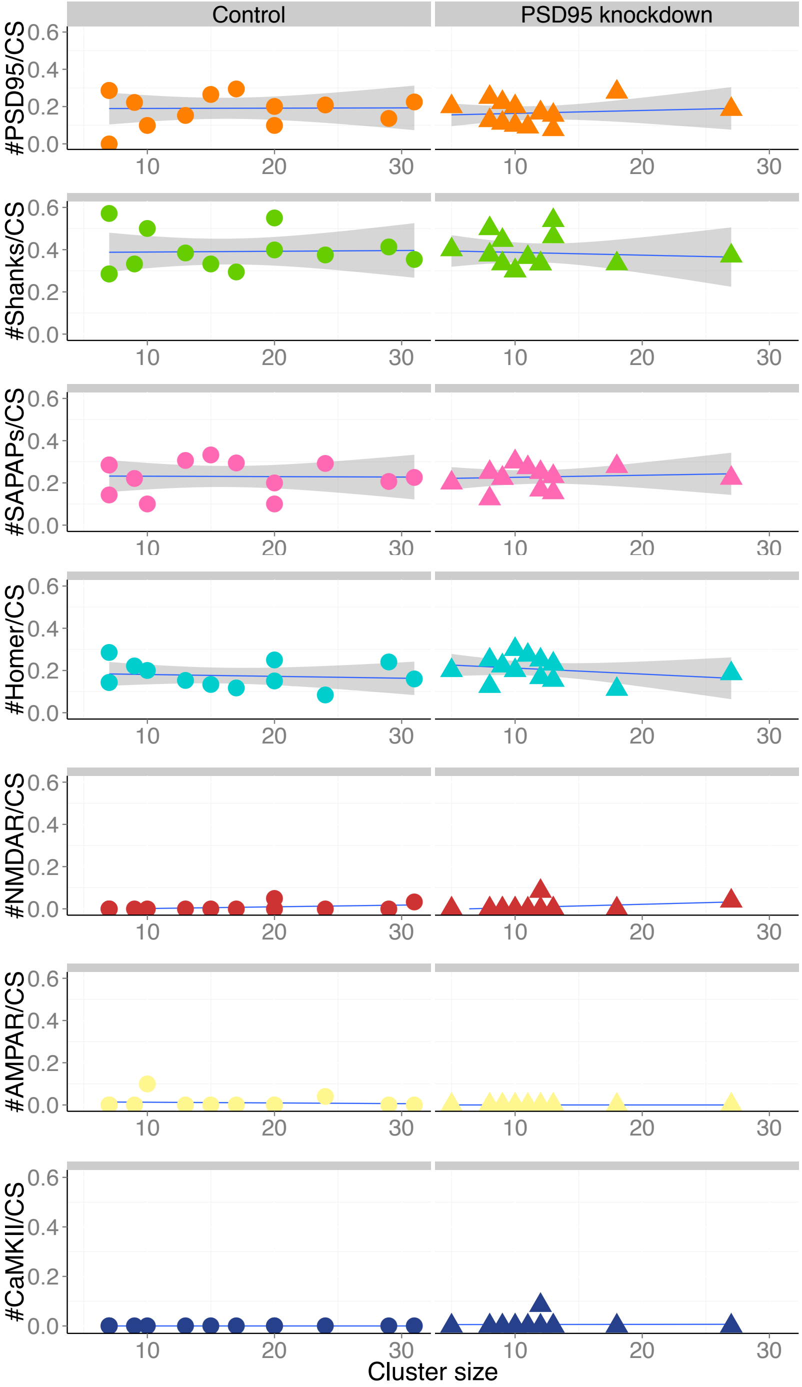


Figure 4.12: Cluster compositions comparison between wild-type and PSD-95 simulations: the protein composition is expressed as relative protein abundance per cluster. This number is calculated as the number of molecules present in a cluster divided by the cluster size.

A clear inverse relationship is found in the wild-type condition (figure 4.13, control condition) when the relative cluster abundance of PSD–95 is compared to the relative abundance of the Shanks isoforms. Interestingly this correlation is weakened by PSD–95 knockdown. This evidence may suggest that the knockdown concentration of PSD–95 is not high enough to cause steric competition between the two proteins, leading to a random incorporation of either protein.

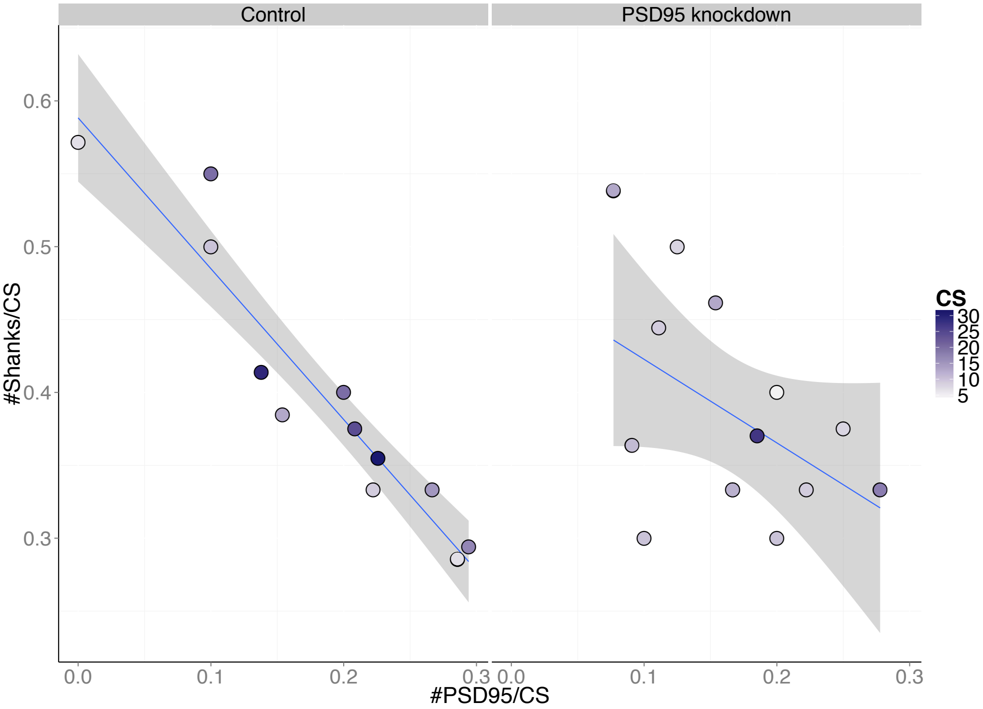


Figure 4.13: Comparison between the relative abundances of PSD-95 and Shanks in wild-type and PSD-95 knockdown simulations: For both conditions, the relative abundance of PSD95 is plotted against the relative abundance of the Shanks isoforms for cluster sizes (CS) higher than four. R^2= 0.879 for the control experiment and R^2= 0.316 for the PSD-95 knockdown experiment. Cluster sizes are represented with the colour gradient.

4.4 Discussion

Due to the high complexity of the PSD assembly, its formation and the role of the proteins involved are still very poorly understood. Computational approaches like the one presented in this thesis, can be an important tool to gain insights into the architecture of the PSD as well as to understand which components are key to the assembly formation.

The crowded nature of the system proved to be very challenging for the developed methodology. I encountered two problems both related to limitations of PhysX, the physics engine used by Unity.
First of all, the Initialiser level (subsection 2.6.2) in which proteins are introduced into the system until the desired stoichiometry is reached and a random starting configuration is generated, had to be merged into the Simulation level. Having a separate step for the generation of the starting configuration is useful not only to ensure a random configuration but also to ensure that each replicate always adopted the exact same starting positions. However the simulation stopped as soon as the output from the Initialiser was loaded into the Simulation level. This was due to the high number of proteins in close proximity to each other, that required memory intense calculation of collision and binding processes, overloading the physics engine. This problem was solved by merging the Initialiser and the Simulation level, allowing proteins to bind during the spawning process while keeping a certain degree of randomness in the starting configuration.

The crowded environment also caused proteins with complex composition to lose the correct relative positions of their corresponding parts. This is likely related to an error during the calculations of collision exit trajectories. When a protein is translated and rotated in space, the physics engine firstly calculates the translational and rotational matrix for the protein root, then a different matrix is calculated for each part present in the hierarchy. These matrices maintain the relative positions and orientations of each child in the hierarchy to their root. Collisions force the physics engine to calculate new matrices. Too many collisions happen at the same time when the environment is too crowded, causing a calculation error. Due to the closed nature of the physics engine it was impossible to fix or to clearly pinpoint the problem. The solution adopted for this study was to reduce the system size, in order to reduce the number of collisions that the physics engine calculates per time step. Both volume and protein numbers were decreased by the same amount, apart from the protein CaMKII that had to be further reduced due to its very high physiological concentration.

Despite the technical difficulties encountered, the method was able to reproduce experimental evidence and to give some biological insights.
The study showed that it is possible to recreate a PSD-like assembly by using only low resolution protein geometries diffusing in space and binding upon collisions between binding sites. The simulated assemblies showed a layered structure similar to the one observed experimentally, with PSD–95 separated from two closely related layers of SAPAP and Shank isoforms (see figure 4.4).

The analysis of the protein composition revealed that PSD–95, Homer, Shank, and SAPAP family members are always present in the simulated assemblies, confirming their important role as scaffolding proteins. Interestingly, the relative protein composition does not change with the assembly diminution, suggesting a fixed ratio that needs to be maintained between the major PSD proteins. This ratio was found to be 1:1 between PSD–95 and SAPAP isoforms and between Homer and Shanks, while Shanks is in a 2:1 ratio with SAPAP (see figure 4.6). It would be of interest to set up new simulations to further explore the implications of this constant ratio between PSD components, for example implementing knockdown simulations for Homer, Shank as well as the SAPAP family.

The glutamate receptors are rarely found in the bigger network clusters, despite being present in high numbers and having their entire protein surface available for binding. This evidence suggests that NMDAR and AMPAR are not a membrane anchoring point for the assembly, but rather that they are trapped into it. CaMKII is also rarely present in the big network clusters. Unlike the glutamate receptors, it is difficult to speculate about its behaviour since its concentration was drastically reduced due to the previously discussed technical problems.

As previously reported, the major class of scaffolding proteins Homer, Shank, PSD–95 and SAPAP are very important for PSD formation but no study has been made in order to understand if these proteins contribute equally to the PSD architecture. The simulation shows that two proteins contribute more than the others to the PSD organisation: Homer and Shank3. Both proteins function as hubs coordinating the protein network. While Homer was already suggested to organise at least a part of the PSD network, Shank3 has never been pointed out so clearly. From a structural point of view, Homer is a very long rod-like protein that contains at each end two separated binding sites. With these characteristics, it is probably not surprising that Homer holds the role of hub in the PSD network. Unlike Homer, Shank3 is smaller and with less binding sites. Shank proteins interact with Homer and the isoforms of the SAPAP family. Unlike the other members of the family, Shank3 is also able to interact with AMPA receptors. Even if AMPARs are rarely present in the larger clusters, it would be interesting to set up a different set of simulations in which Shank3 loses the ability to bind AMPAR. Such simulations, would confirm whether the reason for the different behaviour of Shank3 compared to the other Shank isoforms resides in the ability to bind an extra protein, or in its structure. Moreover, followup knockdown experiments and simulations for Shank3 and Homer would be of great interest.

The crucial importance of PSD–95 for the PSD assembly formation was confirmed by the knockdown simulations. When the amount of PSD–95 in the system is reduced, network clusters appear to be significantly smaller (figure 4.9) and the networks showed an overall decrease in connectivity (figure 4.10). These findings are in accordance with experimental evidence in which the analysed PSDs presented patchy loss after knocking down PSD–95. The lack of presence of both AMPAR and NMDAR in the bigger networks, for the wild-type as well the knockdown case, suggests that PSD–95 is not only an important component for the connectivity of the network, but also as the main membrane anchoring point for the PSD formation.

Interestingly, when the network clusters were analysed for protein composition, no significant change was found in the knockdown case in both composition and protein ratio. This suggests that in the PSD–95 knockdown, the observed reduced size of the clusters could be the result of the assembly necessity to keep the ratio between the proteins fixed. If this is the case, a similar decrease in cluster size should be seen in knockdown experiments for any of the other main components of the bigger clusters. It would be of interest to run other simulations in order to further investigate the topic.

Chapter 5

Conclusions

> It's not over yet. If we don't end it, nothing can start.
>
> *Takigawa Yoshino*
> ZETSUEN NO TEMPEST

The main aim of this work was to develop an accurate and detailed method to simulate protein-protein interactions and large complex formations and use it to study the properties of PSD formation in a dendritic spine. Dendritic spines are the receiving end of individual synapses and they play a key role in the processing of the synaptic information, as explained in detail in chapter 1. The information is passed across the synaptic cleft to the dendritic spine in the form of neurotransmitter releases. The neurotransmitters interact with the ligand-gated ion channels expressed in the postsynaptic membrane and activate them. Subsequently, an ion influx through the channels triggers two major events inside the spine: activation of specific signalling pathways and depolarisation of the postsynaptic membrane. The integration of this signal with the neighbouring spines through time is at the basis of yet elusive learning and memory mechanisms.

In excitatory synapses, an important aspect of this integration is the specific environment in which the ligand gated ion channels and the signalling molecules that interact with them are located: the post synaptic density or PSD. The PSD is a functional specialisation of the postsynaptic membrane and the adjacent cytoplasmic compartment. It is a proteinaceous organelle containing glutamate receptors, their associated signalling molecules as well scaffolding proteins. The PSD size and composition can vary with time and upon stimulation. Stronger synapses which contain a higher number of glutamate receptors also show larger PSDs. The exact structure and composition of the PSD is yet unknown, but the major components have been identified as well as its layered organisation. I needed a simulation platform that allows the study of protein-protein interaction and the formation of large multiprotein complexes in order to study the roles of some of the major components of the PSD assembly and its formation.

There are many simulation platforms available depending on the type of biological process and the level of detail at which the process is considered. Nonetheless, there is still the need for a comprehensive methodology that takes advantage of protein structural information. For this reason I decided to focus on the development of a simulation environment in which protein-protein interactions can be studied with a level of detail that is between an all-atom representation of a protein and a primitive geometry representation for protein-protein interactions studies and proteins assembly formation. The proposed method, presented in chapter 2, takes advantage of the computer animation software Maya, to create a coarse grained representation of the protein surfaces starting from their PDB file structure. Subsequently Unity, a game engine, is used to simulate diffusion and reaction of the proteins models in the simulation environment. The workflow for the generation of the protein models, as explained in section 2.4, generates a low resolution protein surface from the protein structural data. Binding sites of interest are separated from this low polygonal surface and the protein is then reassembled into a hierarchical structure. Once the protein models are completed they can be loaded into Unity and simulated in virtually any environment using the developed Unity extension T.A.R.S.I.D.

I decided to use an agent based approach in T.A.R.S.I.D, having problems in mind such as combinatorial explosion that arise from modelling multiprotein complexes. Protein behaviour is encoded as a set of hierarchical rules: global, local and hyper-local. Global rules affect the entire environment. Local rules affect the single proteins, while hyper-local rules affect the behaviour of the binding site of a protein.

Diffusion was encoded in two different rules: one for the diffusion of cytosolic proteins and one for the diffusion of membrane proteins. Cytosolic proteins are allowed to diffuse in 3D with explicit translation and rotation routines that are mass driven. Encoding the diffusion as mass driven allowed for the precise calculation of the translational and rotational diffusion coefficient of any modelled multiprotein complex. Membrane proteins are allowed to diffuse in 2D on the surface of the simulation container. The diffusion was encoded as mass driven with explicit calculation of translation and rotation, but rotation was only allowed on one axes. One of the feature of the method presented in this thesis is the use of non regular geometries for both proteins and the simulation containers. The diffusion of 2D proteins needs to be independent of the shape of the membrane when a non regular geometry is used as a container. In cases of concave containers, the accumulation of proteins where the surface changes its curvature is a common artefact. This effect seen when 2D diffusion is performed as 3D and subsequently projected back onto the surface. The use of ray-casting together with the local specific orientation of all the membrane proteins, made the implemented 2D diffusion not only independent of the shape of the container, but also independent from its concave nature and its tessellation, as shown in subsection 3.1.4. Validation of the 3D and 2D diffusion reported in section 3.1 showed that both diffusions are accurate and they produce an unbiased random walk.

The binding strategy implemented in T.A.R.S.I.D is based on collision theory: when two proteins collide with their binding sites a binding occurs. Protein binding reactions are considered unidirectional. Therefore one protein is defined as the protein holding the binding rule while the other is considered as target of the binding event. This approach allows the reduction of the number of rules encoded to only one rule for each reaction. Limiting binding events to the collision of binding sites implies that the area of a binding site influences the kinetics of the reaction. This was verified in subsection 3.2.3, where models with bigger binding site areas showed faster kinetics compared to smaller areas. One should be aware of its influence on the binding kinetics, since the definition of a protein binding site is a manual procedure that is influenced by the tessellation of the protein surface and by the quality of the experimental information available.

The binding strategy was validated comparing the geometry of simulated complexes with experimental ones, as reported in section 3.3. The simulated complexes have a much lower resolution compared to methods that used an all-atom representation of proteins as input. This was expected since the protein models used for the simulations have a low resolution and a high resolution simulation was not one of the starting goals. The binding strategy is in agreement with collision theory as shown in section 3.2, when the simulated collision frequency is compared to the theoretical one. Moreover the method is able to discriminate between different types of reactions in terms of different resulting kinetics. The kinetics studies also showed that shape influences the velocity of binding. The protein geometries obtained following the proposed workflow showed different kinetics from their equivalent primitive geometries. This evidence validates the importance of explicitly modelling protein shape since its contribution cannot be predetermined.

In chapter 4, the developed methodology was applied to the study of the PSD assembly formation. The study involved 13 of the major components of the PSD assembly: NMDA and AMPA receptors, PSD–95, Homer, Shank family (Shank1–3), the SAPAP family (SAPAP1–4), the TARP protein Stargazin and CaMKII.

This study aimed to gain insights on the structural importance of these proteins for the PSD as well as confirming that the binding of protein is the main mechanism for the

PSD formation.

The simulations demonstrate that realistic protein geometries diffusing in the simulation environment and a set of binding rules is sufficient to recreate aggregates that resemble the PSD. These aggregates analysed as network clusters showed the same layered structure that is observed in microscopic studies of the PSD. Clusters of different sizes are found in each simulation, indicating that the assembly process starts from binary interactions. These initial clusters grow over time and include more and more proteins. Glutamate receptors, despite being present in large quantities, are not highly represented in the larger clusters. This finding could suggest that the receptors are not the nucleation point for the PSD assembly but instead are trapped by it.

Interestingly two proteins seem to be the hubs of the assembly networks: Homer and Shank3. Homer is a tetrameric scaffolding protein that interacts with Shank proteins and membrane receptor. Shank3 is a scaffolding protein that binds SAPAP proteins and unlike other members of the Shank family also binds to AMPA receptors. Shank proteins also have the ability to dimerise, in a tail to tail manner, with other member of the family. The high capability of these proteins to interact and connect with others makes them the ideal central hubs for the simulated assembly. Knockdown simulations and experiments of both these proteins could elucidate the repercussions on the overall structure of the PSD. In addition, eliminating Shank3 capability to bind AMPA receptors could help to understand whether the reason for the different behaviour of Shank3 compared to the other Shank isoforms resides in the ability to bind an extra protein, or in its structure.

PSD–95 is another scaffolding protein known for interacting with several components in the PSD. It binds NMDAR and to AMPAR via the interaction with Stargazin. PSD–95 also binds to proteins from the SAPAP family, a scaffolding family that interacts with the Shank family. Experiments in which PSD–95 was knocked down showed a PSD with patchy loss, suggesting that PSD–95 has an important role within the PSD structure. To further investigate this phenomenon, I reproduced the knockdown condition by decreasing the PSD–95 concentration alone in the system. The simulated assembly showed clusters of smaller size in comparison to the wild-type condition and in accordance with the experimental evidence. The clusters, despite their smaller size, maintained the same composition and layered structure as the wild-type. This suggests that in the PSD–95 knockdown, the observed reduced size of the clusters could be the result of the assembly necessity to keep the ratio between the proteins fixed. The connectivity of the knockdown networks was lower compared to the wild-type cases, suggesting that the loss of PSD–95 negatively affects the amount of proteins that are able to aggregate together but not the composition of the assembly itself. Altogether these results indicate the importance of PSD–95 as a protein that is required for a formation of a large PSD assembly.

Utilising a different methodology for the study of the PSD, such as a dimensionless single particle simulator, could have never led to the results obtained with the methodology presented in this thesis. Thanks to the use of geometries with finite volumes, the simulations showed steric impediment for binding events that occur later in the simulations. These impediment contributed to the dynamic and composition of the simulated assemblies, as in a more realistic scenario.
Moreover, not taking into account protein shapes and sizes, as well as the binding site areas by using a dimensionless particle simulator again would greatly affect the binding kinetics (as demonstrated in section 3.2) therefore the assemblies composition. It is likely that PSD assemblies, obtained with dimensionless particle simulations, would have a more random composition unlike the simulations presented in chapter 4 that show a fixed ratio between components that is independent from the assembly size.

When the new simulation methodology presented in this thesis was applied to the PSD assembly formation, I encountered two major problems attributable to limitations of the physics engine of Unity, PhysX. The first problem was encountered at the very start of the simulations. The simulation environment is initially populated with the protein models that do not possess binding rules until the desired amount of proteins is reached. Subsequently, position and rotation saved for each protein in the system are loaded into the Simulation in which proteins are allowed to bind. Due to the crowded nature of the PSD assembly, too many collisions occur at the same time at the start of the Simulation, overloading the physics engine. This problem was solved by merging the Initialiser and the Simulation level, allowing proteins to bind during the spawning process. The second problem was also caused by the crowded nature of the environment. Proteins with a complex hierarchy tend to lose the correct relative position and orientation between their parts after numerous collisions. The problem could neither be precisely pinpointed nor solved due to the Closed Source nature of PhysX, and forced me to reduce the size of the system in order to minimise the errors. This illustrates why the use of use of proprietary software programs in science, could be problematic. The advantage of proprietary programs, especially when developed by big companies such as PhysX which is developed by NVIDIA, is that they are stable, constantly curated and updated, and that they provide an appropriate documentation. On the other hand, the possibility to precisely address and further develop a system is of huge importance when a user pushes the capabilities of the system to its maximum, which often occurs in science. Given the opportunity, this work should be implemented in an open platform, with both the physics engine and the game engine being Open Source. Possibly one of the most promising Open Source programs is the 3D animation suite Blender[1]. In the last few years, BioBlender[2] a new plugin for importing PDB files was developed, and thanks to the incorporation of the Open Source physics engine Bullets[3], Blender can now also be used as a game engine. It would be of interesting to test these new capabilities of Blender and compare its performance with Unity and Maya. Another useful feature that could have been implemented on an open physics engine is a geometrical constraint on the protein binding. At the current state, two proteins are allowed to go through the binding process if they collide between two available binding sites. No other limitations are in place, suggesting that a binding can occur even if the contact area between the two is a single mesh vertex. Since upon binding the protein orientations are locked in place, it would be more accurate to put a threshold on the minimum number of vertices that need to be in contact to trigger a binding event. More binding surface area implies that more residues on the binding sites of the two proteins interact. This could assure a more accurate protein complex orientation even by only increasing the threshold from one vertex to two vertices. Utilising such a threshold would be possible only on a system that let the user interfere with the routines that handle collision events, which is not the case for closed systems such as PhysX.

Overall the method presented in this thesis is a good example for modelling biological systems at the *mesoscale*, a scale in between the microscopic and nanoscale. The method performed well considering the discussed limitations. The agent based nature of the system, together with the ability to encode event driven actions, the possibility to any desired geometrical shape, and the accurate diffusion in both 2D and 3D make this method highly flexible. No other available simulation method allows for 2/3D particle

[1] http://www.blender.org

[2] http://www.bioblender.eu

[3] http://bulletphysics.org

based spatial simulation in which proteins are represented as a 3D geometry, allowing therefore excluded volume studies. Moreover this methodology allows the proteins and molecules simulated in the system to interact with each other, allowing protein assemblies studies, like the one proposed in this thesis, but also allowing any protein interaction: activation, inactivation, phosphorylation and so on.

Further applications of this method beyond the PSD assembly study could be complex simulations like the neurotransmitters release from synaptic vesicles into the synaptic cleft, or any other protein signalling pathway.

T.A.R.S.I.D. can be downloaded on sourceforge[4], but soon will be published in the Unity Asset Store[5] as a free Open Source Unity extension. At the moment, the only reactions implemented in this method are protein-protein bindings. It will be of interest to further develop the method by including catalysis. This could be easy achieved by using events driven by collisions as for the protein-protein bindings. Considering for example the case of protein A that is phosphorylated by protein kinase B on its binding site BS; this phosphorylation causes the increase in affinity of protein A for protein C. This very common scenario can be divided into two steps: the phosphorylation reaction and the resulting increase in affinity.

The phosphorylation can be implemented as a state on the binding site BS: the collision of the binding site BS of protein A with the catalytic site of the kinase B will trigger the change in state of BS from unphosphorylated to phosphorylated. The phosphorylation itself can be made reversible either by explicitly modelling the action of a phosphatase or by adding a probability for reversing the state to unphosphorylated. Once the state of BS is phosphorylated, protein A shows an increase in affinity for protein C. In the proposed methodology, the association part of every reaction (k_{on}) is determined by collisions between binding sites, which is dependent on the area of the binding sites involved in the reaction and the diffusion properties of the two proteins. Therefore, an increase in affinity can be implemented as a decrease in the probability of dissociation of the complex AC. Dissociation probabilities can be calculated from the experimentally derived equilibrium constants. Specific simulations for parameter estimation (similar to the one reported in section 3.2) can be done to estimate k_{on} in cases where the dissociation probability k_{off} needs to be precisely calculated. From those simulations, k_{on} that accounts for the

[4] http://sourceforge.net/projects/tarsid

[5] www.assetstore.unity3d.com

area defined as binding site on the protein geometry, can be estimated. Knowing the experimentally derived equilibrium constant and with the obtained k_{on}, is possible to determined the dissociation probability.

There is the need for more comprehensive modelling approaches with the increase of available data from different sources, such as structural data, imaging, interaction and regulatory networks, and expression data. Implementations of the presented methodology on similar platforms (such as Blender) could lead to a more collaborative and long lasting effort to incorporate these new data into simulations.

Having time and resources a more automated pipeline could have been put into place, in which for instant protein geometries are automatically created. In doing so, one should take into account not only the mesh generation process but also the mesh reduction process that can lead to big errors on the protein geometries (as discussed in section 2.4). To reduce the amount of errors, it could be possible to automatically generate several geometries with different setups, and subsequently rank them based on quality check parameters. This process could then be iterated as in a genetic algorithm to select the setup that produces the best result for each protein. Automated routines for the analysis of the results could also be implemented. Ranging from data extraction of the simulation output, to images and plots generations, several automated routines from different sources could be plugged together into a single pipeline. Another improvement to the methodology would be to implement it on a platform that is more suitable for parallelisation. Parallelisation would assure bigger and longer simulations, increasing even more the flexibility of this methodology.

The use of an automated pipeline running on a parallel system would also allow for this methodology to be used for massive scale models such as single cells, or single cell organisms. A lot of effort has been put in recent years to integrate all the knowledge of a system into a single massive model. Due to the lack of data an all the biological processes, but more importantly due to the lack of a system that can handle high level of details on a massive scale, these models are generally implemented as ODEs systems. As previously discussed in this thesis, ODEs are a fast way to encode the dynamic of a biological system, but by far the more accurate. Giving a platform that can simulate single particles with finite geometries in space, like the presented methodology, on a massive scale could lead not only to improve the existing models, but could aid in gaining new biological insight as well as discover new hot spots for drug development.

The use and further development of tools developed within different context, like the one proposed in this thesis, should be welcomed if they can be utilised by the scientific community.

The proposed method is a good example for this knowledge and technology import. It shows how a framework originally developed in the context of the gaming industry can be utilised to execute realistic mesoscale computer simulations of biological systems. The presented attempt to close the gap in the methodology of mesocale modelling in systems biology is highlighting a major landmark for future work in this yet barely explored direction.

Bibliography

Alber, F. et al. (Nov. 2007). “Determining the architectures of macromolecular assemblies”. *Nature* 450.7170, pp. 683–694.

Alcor, D., G. Gouzer, and A. Triller (Sept. 2009). “Single-particle tracking methods for the study of membrane receptors dynamics”. *The European journal of neuroscience* 30.6, pp. 987–997.

Andersen, P (1990). “Synaptic integration in hippocampal CA1 pyramids.” *Progress in brain research* 83, pp. 215–222.

Andersen, P, T. W. Blackstad, and T Lomo (1966). “Location and identification of excitatory synapses on hippocampal pyramidal cells.” *Experimental brain research* 1.3, pp. 236–248.

Andrei, R. M., M. Callieri, M. F. Zini, T. Loni, G. Maraziti, M. C. Pan, and M. Zoppe (2012). “Intuitive representation of surface properties of biomolecules using BioBlender.” *BMC bioinformatics* 13 Suppl 4, S16.

Andrews, S. S. and D. Bray (Dec. 2004). “Stochastic simulation of chemical reactions with spatial resolution and single molecule detail.” *Physical biology* 1.3-4, pp. 137–151.

Andrews, S. S., N. J. Addy, R. Brent, and A. P. Arkin (Mar. 2010). “Detailed simulations of cell biology with Smoldyn 2.1.” *PLoS Computational Biology* 6.3, e1000705.

Autin, L, G Johnson, J Hake, A. Olson, and M Sanner (2012). “uPy: A Ubiquitous CG Python API with Biological-Modeling Applications”. *IEEE Computer Graphics and Applications* 32.5, pp. 50–61.

Banke, T. G. and S. F. Traynelis (Feb. 2003). “Activation of NR1/NR2B NMDA receptors”. *Nature Neuroscience* 6.2, pp. 144–152.

Banker, G, L Churchill, and C. W. Cotman (Nov. 1974). “Proteins of the postsynaptic density.” *The Journal of Cell Biology* 63.2 Pt 1, pp. 456–465.

Bard, L., M. Sainlos, D. Bouchet, S. Cousins, L. Mikasova, C. Breillat, F. A. Stephenson, B. Imperiali, D. Choquet, and L. Groc (Nov. 2010). “Dynamic and specific interaction between synaptic NR2-NMDA receptor and PDZ proteins”. *Proceedings of the National Academy of Sciences of the United States of America* 107.45, pp. 19561–19566.

Baron, M. K., T. M. Boeckers, B. Vaida, S. Faham, M. Gingery, M. R. Sawaya, D. Salyer, E. D. Gundelfinger, and J. U. Bowie (Jan. 2006). “An architectural framework that may lie at the core of the postsynaptic density". *Science (New York, NY)* 311.5760, pp. 531–535.

Bassani, S., A. Folci, J. Zapata, and M. Passafaro (Dec. 2013). "AMPAR trafficking in synapse maturation and plasticity." *Cellular and molecular life sciences : CMLS* 70.23, pp. 4411–4430.

Bats, C., L. Groc, and D. Choquet (Mar. 2007). "The interaction between Stargazin and PSD-95 regulates AMPA receptor surface trafficking." *Neuron* 53.5, pp. 719–734.

Bats, C., D. Soto, D. Studniarczyk, M. Farrant, and S. G. Cull-Candy (May 2012). "Channel properties reveal differential expression of TARPed and TARPless AMPARs in stargazer neurons." *Nature Neuroscience.*

Bayes, A., L. N. van de Lagemaat, M. O. Collins, M. D. R. Croning, I. R. Whittle, J. S. Choudhary, and S. G. N. Grant (Dec. 2010). "Characterization of the proteome, diseases and evolution of the human postsynaptic density." *Nature Neuroscience* 14.1, pp. 19–21.

Beck, M., V. Lucic, F. Forster, W. Baumeister, and O. Medalia (Oct. 2007). "Snapshots of nuclear pore complexes in action captured by cryo-electron tomography." *Nature* 449.7162, pp. 611–615.

Berg, H. C. (1993). *Random Walks in Biology.* Princeton University Press.

Berman, H., K. Henrick, and H. Nakamura (Dec. 2003). "Announcing the worldwide Protein Data Bank." *Nature structural biology* 10.12, p. 980.

Bhalla, U. S. and R Iyengar (Jan. 1999). "Emergent properties of networks of biological signaling pathways." *Science (New York, NY)* 283.5400, pp. 381–387.

Bliss, T. V. and T Lomo (July 1973). "Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path." *The Journal of Physiology* 232.2, pp. 331–356.

Boeckers, T. M., C Winter, K. H. Smalla, M. R. Kreutz, J Bockmann, C Seidenbecher, C. C. Garner, and E. D. Gundelfinger (Oct. 1999). "Proline-rich synapse-associated proteins ProSAP1 and ProSAP2 interact with synaptic proteins of the SAPAP/GKAP family." *Biochemical and biophysical research communications* 264.1, pp. 247–252.

Bourne, J. and K. M. Harris (June 2007). "Do thin spines learn to be mushroom spines that remember?": *Current opinion in neurobiology* 17.3, pp. 381–386.

Bourne, J. N. and K. M. Harris (2008a). "Balancing structure and function at hippocampal dendritic spines". *Annual Review of Neuroscience* 31, pp. 47–67.

Bourne, J. N. and K. M. Harris (2008b). "Balancing structure and function at hippocampal dendritic spines". *Annual Review of Neuroscience* 31, pp. 47–67.

Brakeman, P. R., A. A. Lanahan, R O'Brien, K Roche, C. A. Barnes, R. L. Huganir, and P. F. Worley (Mar. 1997). "Homer: a protein that selectively binds metabotropic glutamate receptors." *Nature* 386.6622, pp. 284–288.

Brandes, U. (June 2001). "A faster algorithm for betweenness centrality*". *The Journal of Mathematical Sociology* 25.2, pp. 163–177.

Brenman, J. E. et al. (Mar. 1996). “Interaction of nitric oxide synthase with the postsynaptic density protein PSD-95 and alpha1-syntrophin mediated by PDZ domains.” *Cell* 84.5, pp. 757–767.

Broderick, G., M. Ru’aini, E. Chan, and M. J. Ellison (2005). “A life-like virtual cell membrane using discrete automata.” *In silico biology* 5.2, pp. 163–178.

Burkhardt, C., M. Muller, A. Badde, C. C. Garner, E. D. Gundelfinger, and A. W. Puschel (July 2005). “Semaphorin 4B interacts with the post-synaptic density protein PSD-95/SAP90 and is recruited to synapses through a C-terminal PDZ-binding motif.” *FEBS letters* 579.17, pp. 3821–3828.

Byrne, M. J., M. N. Waxham, and Y. Kubota (June 2010). “Cellular dynamic simulator: an event driven molecular simulation environment for cellular physiology”. *Neuroinformatics* 8.2, pp. 63–82.

Cajal, S (1892). “A new concept of the histology of neural centers”. *Rev Cienc Med* 18, pp. 457–476.

Cajal, S. R. y (1889). “Conexión general de los elementos nerviosos”. *Med. Pract.* 353.

Cannon, R. C. and G. D’Alessandro (Aug. 2006). “The ion channel inverse problem: neuroinformatics meets biophysics.” *PLoS Computational Biology* 2.8, e91.

Carlin, R. K., D. J. Grab, R. S. Cohen, and P Siekevitz (Sept. 1980). “Isolation and characterization of postsynaptic densities from various brain regions: enrichment of different types of postsynaptic densities.” *The Journal of Cell Biology* 86.3, pp. 831–845.

Chang, C.-W., S.-C. Peng, W.-Y. Cheng, S.-H. Liu, H.-H. Cheng, S.-Y. Huang, and Y.-C. Chang (Nov. 2007). “Studying the protein-protein interactions in the postsynaptic density by means of immunoabsorption and chemical crosslinking”. *Proteomics Clinical applications* 1.11, pp. 1499–1512.

Chen, X., L. Vinade, R. D. Leapman, J. D. Petersen, T. Nakagawa, T. M. Phillips, M. Sheng, and T. S. Reese (Aug. 2005). “Mass of the postsynaptic density and enumeration of three key molecules”. *Proceedings of the National Academy of Sciences of the United States of America* 102.32, pp. 11551–11556.

Chen, X., C. A. Winters, and T. S. Reese (Sept. 2008a). “Life inside a thin section: tomography”. *The Journal of neuroscience : the official journal of the Society for Neuroscience* 28.38, pp. 9321–9327.

Chen, X., C. Winters, R. Azzam, X. Li, J. A. Galbraith, R. D. Leapman, and T. S. Reese (Mar. 2008b). “Organization of the core structure of the postsynaptic density”. *Proceedings of the National Academy of Sciences of the United States of America* 105.11, pp. 4453–4458.

Chen, X., C. D. Nelson, X. Li, C. A. Winters, R. Azzam, A. A. Sousa, R. D. Leapman, H. Gainer, M. Sheng, and T. S. Reese (Apr. 2011). “PSD-95 Is Required to Sustain the Molecular Organization of the Postsynaptic Density”. *The Journal of neuroscience : the official journal of the Society for Neuroscience* 31.17, pp. 6329–6338.

Cheng, D (Mar. 2006). “Relative and Absolute Quantification of Postsynaptic Density Proteome Isolated from Rat Forebrain and Cerebellum”. *Molecular & Cellular Proteomics* 5.6, pp. 1158–1170.

Cheng, D. et al. (June 2006). “Relative and absolute quantification of postsynaptic density proteome isolated from rat forebrain and cerebellum”. *Molecular & cellular proteomics : MCP* 5.6, pp. 1158–1170.

Cho, K. O., C. A. Hunt, and M. B. Kennedy (Nov. 1992). “The rat brain postsynaptic density fraction contains a homolog of the Drosophila discs-large tumor suppressor protein.” *Neuron* 9.5, pp. 929–942.

Choi, U. B., R. Kazi, N. Stenzoski, L. P. Wollmuth, V. N. Uversky, and M. E. Bowen (Aug. 2013). “Modulating the intrinsic disorder in the cytoplasmic domain alters the biological activity of the N-methyl-D-aspartate-sensitive glutamate receptor.” *The Journal of biological chemistry* 288.31, pp. 22506–22515.

Choquet, D (2003). “The role of receptor diffusion in the organization of the postsynaptic membrane”. *Nature Reviews Neuroscience*.

Citri, A. and R. C. Malenka (Jan. 2008). “Synaptic plasticity: multiple forms, functions, and mechanisms.” *Neuropsychopharmacology : official publication of the American College of Neuropsychopharmacology* 33.1, pp. 18–41.

Colbran, R. J. and A. M. Brown (June 2004). “Calcium/calmodulin-dependent protein kinase II and synaptic plasticity”. *Current opinion in neurobiology* 14.3, pp. 318–327.

Collins, M. O., L. Yu, M. P. Coba, H. Husi, I. Campuzano, W. P. Blackstock, J. S. Choudhary, and S. G. N. Grant (Feb. 2005). “Proteomic analysis of in vivo phosphorylated synaptic proteins.” *The Journal of biological chemistry* 280.7, pp. 5972–5982.

Collins, M. O., H. Husi, L. Yu, J. M. Brandon, C. N. G. Anderson, W. P. Blackstock, J. S. Choudhary, and S. G. N. Grant (Apr. 2006). “Molecular characterization and comparison of the components and multiprotein complexes in the postsynaptic proteome”. *Journal of neurochemistry* 97, pp. 16–23.

Connolly, M. L. (Oct. 1983). “Analytical molecular surface calculation”. *Journal of Applied Crystallography* 16.5, pp. 548–558.

Connolly, M. L. (June 1993). “The molecular surface package.” *Journal of molecular graphics* 11.2, pp. 139–141.

Cooper, S., F. Khatib, A. Treuille, J. Barbero, J. Lee, M. Beenen, A. Leaver-Fay, D. Baker, Z. Popovic, and F. Players (Aug. 2010). “Predicting protein structures with a multiplayer online game.” *Nature* 466.7307, pp. 756–760.

Cukor, J., J. Spitalnick, J. Difede, A. Rizzo, and B. O. Rothbaum (Dec. 2009). “Emerging treatments for PTSD.” *Clinical psychology review* 29.8, pp. 715–726.

Cull-Candy, S. G. and D. N. Leszkiewicz (Oct. 2004). “Role of distinct NMDA receptor subtypes at central synapses”. *Science's STKE : signal transduction knowledge environment* 2004.255, re16.

Dani, A., B. Huang, J. Bergan, C. Dulac, and X. Zhuang (Dec. 2010). “Superresolution imaging of chemical synapses in the brain.” *Neuron* 68.5, pp. 843–856.

De Robertis, E and H. S. Bennet (Jan. 1955). “Some features of the submicroscopic morphology of synapses in frog and earthworm.” *The Journal of biophysical and biochemical cytology* 1.1, pp. 47–58.

Deiters, O. and M. J. S. Schultze (1865). *Untersuchungen über Gehirn und Rückenmark des Menschen und der Säugethiere / Nach dem Tode des Verfassers hrsg. von Max Schultze.* Untersuchungen über Gehirn und Rückenmark des Menschen und der Säugethiere /. Braunschweig : Braunschweig Vieweg.

Dingledine, R, K Borges, D Bowie, and S. F. Traynelis (Mar. 1999). “The glutamate receptor ion channels.” *Pharmacological Reviews* 51.1, pp. 7–61.

Dix, J. A. and A. S. Verkman (2008). “Crowding effects on diffusion in solutions and cells”. *Annual review of biophysics* 37, pp. 247–263.

Doncheva, N. T., Y. Assenov, F. S. Domingues, and M. Albrecht (Apr. 2012). “Topological analysis and interactive visualization of biological networks and protein structures.” *Nature protocols* 7.4, pp. 670–685.

Dosemeci, A, T. S. Reese, J Petersen, and J. Tao-Cheng (2000). “A novel particulate form of Ca2+/CaMKII-dependent protein kinase II in neurons”. *The Journal of neuroscience : the official journal of the Society for Neuroscience* 20.9, p. 3076.

Dosemeci, A., A. J. Makusky, E. Jankowska-Stephens, X. Yang, D. J. Slotta, and S. P. Markey (Oct. 2007). “Composition of the synaptic PSD-95 complex.” *Molecular & cellular proteomics : MCP* 6.10, pp. 1749–1760.

Dror, R. O., R. M. Dirks, J. P. Grossman, H. Xu, and D. E. Shaw (2012). “Biomolecular simulation: a computational microscope for molecular biology.” *Annual review of biophysics* 41, pp. 429–452.

Durrant, J. D. and J. A. McCammon (2011). “Molecular dynamics simulations and drug discovery.” *BMC biology* 9, p. 71.

Ehlers, M. D. (Nov. 1999). “Synapse structure: glutamate receptors connected by the shanks.” *Current biology : CB* 9.22, R848–50.

Ehlers, M. D., S Zhang, J. P. Bernhadt, and R. L. Huganir (Mar. 1996). “Inactivation of NMDA receptors by direct interaction of calmodulin with the NR1 subunit”. *Cell* 84.5, pp. 745–755.

Ehlers, M. D., M. Heine, L. Groc, M.-C. Lee, and D. Choquet (May 2007). “Diffusional trapping of GluR1 AMPA receptors by input-specific synaptic activity”. *Neuron* 54.3, pp. 447–460.

Ehrenberg, C. G. (1836). “Bemerkungen uber feste mikroscopische anorganische Formen in den erdigen und derben Mineralien”. *Berichte der Königlichen Preussischen Akademie der Wissenschaften zu Berlin* 1836, pp. 84–85.

Einstein, A (1905). "Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen". *Annalen der Physik* 322.8, pp. 549–560.

Eisenberg, D, E. M. Marcotte, I Xenarios, and T. O. Yeates (June 2000). "Protein function in the post-genomic era." *Nature* 405.6788, pp. 823–826.

El-Husseini, A. E.-D., E. Schnell, S. Dakoji, N. Sweeney, Q. Zhou, O. Prange, C. Gauthier-Campbell, A. Aguilera-Moreno, R. A. Nicoll, and D. S. Bredt (Mar. 2002). "Synaptic strength regulated by palmitate cycling on PSD-95." *Cell* 108.6, pp. 849–863.

Elcock, A. H. (May 2002). "Atomistic simulations of competition between substrates binding to an enzyme." *Biophysical Journal* 82.5, pp. 2326–2332.

Ellis, R. J. (Oct. 2001). "Macromolecular crowding: obvious but underappreciated". *Trends in Biochemical Sciences* 26.10, pp. 597–604.

Erdős, P. and A. Rényi (1959). "On random graphs". *Publicationes Mathematicae Debrecen* 6, pp. 290–297.

Erondu, N. E. and M. B. Kennedy (Dec. 1985). "Regional distribution of type II Ca2+/calmodulin-dependent protein kinase in rat brain." *The Journal of neuroscience : the official journal of the Society for Neuroscience* 5.12, pp. 3270–3277.

Euler, L. (1736). "Methodus universalis series summandi ulterius promota". *Commentarii academiae scientiarum Petropolitanae* 8.1736, pp. 147–158.

Fernández, E., M. O. Collins, R. T. Uren, M. V. Kopanitsa, N. H. Komiyama, M. D. R. Croning, L. Zografos, J. D. Armstrong, J. S. Choudhary, and S. G. N. Grant (2009). "Targeted tandem affinity purification of PSD-95 recovers core postsynaptic complexes and schizophrenia susceptibility proteins." *Molecular Systems Biology* 5, p. 269.

Fifková, E (June 1985). "Actin in the nervous system." *Brain Research* 356.2, pp. 187–215.

Foster, K. A., N. McLaughlin, D. Edbauer, M. Phillips, A. Bolton, M. Constantine-Paton, and M. Sheng (Feb. 2010). "Distinct roles of NR2A and NR2B cytoplasmic tails in long-term potentiation". *The Journal of neuroscience : the official journal of the Society for Neuroscience* 30.7, pp. 2676–2685.

Foster, M. (1897). *A Text Book of Physiology: The central nervous system.* Vol. 3. Macmillan.

Fukata, M., Y. Fukata, H. Adesnik, R. A. Nicoll, and D. S. Bredt (Dec. 2004). "Identification of PSD-95 palmitoylating enzymes." *Neuron* 44.6, pp. 987–996.

Fukata, Y., A. V. Tzingounis, J. C. Trinidad, M. Fukata, A. L. Burlingame, R. A. Nicoll, and D. S. Bredt (May 2005). "Molecular constituents of neuronal AMPA receptors." *The Journal of Cell Biology* 169.3, pp. 399–404.

Gamble, E and C Koch (June 1987). "The dynamics of free calcium in dendritic spines in response to repetitive synaptic input." *Science (New York, NY)* 236.4806, pp. 1311–1315.

Garcia, R. A., K Vasudevan, and A Buonanno (Mar. 2000). “The neuregulin receptor ErbB-4 interacts with PDZ-containing proteins at neuronal synapses.” *Proceedings of the National Academy of Sciences of the United States of America* 97.7, pp. 3596–3601.

Gielen, M., B. Retchless, L Mony, J. Johnson, and P. Paoletti (2009). “Mechanism of differential control of NMDA receptor activity by NR2 subunits”. *Nature* 459.7247, pp. 703–707.

Gillespie, D. (1977). “Exact stochastic simulation of coupled chemical reactions”. *The Journal of Physical Chemistry.*

Goldman-Rakic, P. S., C Leranth, S. M. Williams, N Mons, and M Geffard (Nov. 1989). “Dopamine synaptic complex with pyramidal neurons in primate cerebral cortex.” *Proceedings of the National Academy of Sciences of the United States of America* 86.22, pp. 9015–9019.

Golgi, C. (1883). *Generalità sul sistema nervoso ed istologia del tessuto nervoso.*

Golgi, C. (1885). *Sulla fina anatomia degli organi centrali del sistema nervoso.* S. Calderini.

Golgi, C. (1891). *La rete nervosa diffusa degli organi centrali del sistema nervoso: Suo significato fisiologico.* Istituto Lombardo.

Gray, J. J., S. Moughon, C. Wang, O. Schueler-Furman, B. Kuhlman, C. A. Rohl, and D. Baker (Aug. 2003). “Protein-protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations.” *Journal of Molecular Biology* 331.1, pp. 281–299.

Griffith, L. C., C. S. Lu, and X. X. Sun (Oct. 2003). “CaMKII, an enzyme on the move: regulation of temporospatial localization”. *Molecular interventions* 3.7, pp. 386–403.

Groc, L. (2006). “AMPA and NMDA glutamate receptor trafficking: multiple roads for reaching and leaving the synapse”. *Cell and Tissue Research.*

Groc, L., M. Heine, L. Cognet, K. Brickley, F. A. Stephenson, B. Lounis, and D. Choquet (July 2004). “Differential activity-dependent regulation of the lateral mobilities of AMPA and NMDA receptors.” *Nature Neuroscience* 7.7, pp. 695–696.

Harris, K. M. and J. K. Stevens (Aug. 1989). “Dendritic spines of CA 1 pyramidal cells in the rat hippocampus: serial electron microscopy with reference to their biophysical characteristics.” *The Journal of neuroscience : the official journal of the Society for Neuroscience* 9.8, pp. 2982–2997.

Harris, K. M., F. E. Jensen, and B Tsao (July 1992). “Three-dimensional structure of dendritic spines and synapses in rat hippocampus (CA1) at postnatal day 15 and adult ages: implications for the maturation of synaptic physiology and long-term potentiation.” *The Journal of neuroscience : the official journal of the Society for Neuroscience* 12.7, pp. 2685–2705.

Hayashi, M. K., H. M. Ames, and Y. Hayashi (Aug. 2006). “Tetrameric hub structure of postsynaptic scaffolding protein homer”. *The Journal of neuroscience : the official journal of the Society for Neuroscience* 26.33, pp. 8492–8501.

Hayashi, M. K., C. Tang, C. Verpelli, R. Narayanan, M. H. Stearns, R.-M. Xu, H. Li, C. Sala, and Y. Hayashi (Apr. 2009). “The postsynaptic density proteins Homer and Shank form a polymeric network structure”. *Cell* 137.1, pp. 159–171.

Hebb, D. O. (1949). *The organization of behavior.* New York: Wiley.

Hepburn, I., W. Chen, S. Wils, and E. de Schutter (2012). “STEPS: efficient simulation of stochastic reaction-diffusion models in realistic morphologies.” *BMC systems biology* 6, p. 36.

Herculano-Houzel, S. (June 2012). “The remarkable, yet not extraordinary, human brain as a scaled-up primate brain and its associated cost.” *Proceedings of the National Academy of Sciences of the United States of America* 109 Suppl 1, pp. 10661–10668.

Hodgkin, A. L. and A. F. Huxley (Aug. 1952). “A quantitative description of membrane current and its application to conduction and excitation in nerve.” *The Journal of Physiology* 117.4, pp. 500–544.

Hoelz, A., E. W. Debler, and G. Blobel (2011). “The structure of the nuclear pore complex.” *Annual review of biochemistry* 80, pp. 613–643.

Hollmann, M and S Heinemann (1994). “Cloned glutamate receptors.” *Annual Review of Neuroscience* 17, pp. 31–108.

Holmes, W. R. and W. B. Levy (May 1990). “Insights into associative long-term potentiation from computational models of NMDA receptor-mediated calcium influx and intracellular calcium concentration changes.” *Journal of neurophysiology* 63.5, pp. 1148–1168.

Hoops, S., S. Sahle, R. Gauges, C. Lee, J. Pahle, N. Simus, M. Singhal, L. Xu, P. Mendes, and U. Kummer (Dec. 2006). “COPASI–a COmplex PAthway SImulator.” *Bioinformatics (Oxford, England)* 22.24, pp. 3067–3074.

Humphrey, W, A Dalke, and K Schulten (Feb. 1996). “VMD: visual molecular dynamics.” *Journal of molecular graphics* 14.1, pp. 33–8–27–8.

Husi, H., M. A. Ward, J. S. Choudhary, W. P. Blackstock, and S. G. N. Grant (July 2000). “Proteomic analysis of NMDA receptor-adhesion protein signaling complexes”. *Nature Neuroscience* 3.7, pp. 661–669.

Hwang, H., T. Vreven, J. Janin, and Z. Weng (Nov. 2010). “Protein-protein docking benchmark version 4.0.” *Proteins* 78.15, pp. 3111–3114.

Jeong, H, B Tombor, R Albert, Z. N. Oltvai, and A. L. Barabási (Oct. 2000). “The large-scale organization of metabolic networks.” *Nature* 407.6804, pp. 651–654.

Jingsong, C. (2013). “Research on the Application Subdivision Pattern in Triangular Mesh Simplification”. *link.springer.com*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 19–24.

Johnson, G. T., L. Autin, D. S. Goodsell, M. F. Sanner, and A. J. Olson (Mar. 2011). “ePMV embeds molecular modeling into professional animation software environments.” *Structure (London, England : 1993)* 19.3, pp. 293–303.

Johnson, M. S., N Srinivasan, R Sowdhamini, and T. L. Blundell (1994). “Knowledge-based protein modeling.” *Critical reviews in biochemistry and molecular biology* 29.1, pp. 1–68.

Kandel, E. R. and L Tauc (1963). “Prolonged increase in the efficiency of an efferent pathway of an isolated ganglion after the coupled activation of a more effective tract.” *Journal de physiologie* 55, pp. 271–272.

Karplus, M and J Kuriyan (May 2005). “Molecular dynamics and protein function.” *Proceedings of the National Academy of Sciences of the United States of America* 102.19, pp. 6679–6685.

Kasai, H., T. Hayama, M. Ishikawa, S. Watanabe, S. Yagishita, and J. Noguchi (July 2010). “Learning rules and persistence of dendritic spines”. *The European journal of neuroscience.*

Kennedy, M. B. (Oct. 2000). “Signal-processing machines at the postsynaptic density.” *Science (New York, NY)* 290.5492, pp. 750–754.

Kennedy, M. B., M. K. Bennett, and N. E. Erondu (Dec. 1983). “Biochemical and immunochemical evidence that the ”major postsynaptic density protein” is a subunit of a calmodulin-dependent protein kinase.” *Proceedings of the National Academy of Sciences of the United States of America* 80.23, pp. 7357–7361.

Kennedy, M. B., H. C. Beale, H. J. Carlisle, and L. R. Washburn (June 2005). “Integration of biochemical signalling in spines”. *Nature Reviews Neuroscience* 6.6, pp. 423–434.

Kharazia, V. N. and R. J. Weinberg (Nov. 1997). “Tangential synaptic distribution of NMDA and AMPA receptors in rat neocortex”. *Neuroscience Letters* 238.1-2, pp. 41–44.

Kim, E, M Niethammer, A Rothschild, Y. N. Jan, and M. Sheng (Nov. 1995a). “Clustering of Shaker-type K+ channels by interaction with a family of membrane-associated guanylate kinases.” *Nature* 378.6552, pp. 85–88.

Kim, E, M Niethammer, A Rothschild, Y. N. Jan, and M. Sheng (Nov. 1995b). “Clustering of Shaker-type K+ channels by interaction with a family of membrane-associated guanylate kinases.” *Nature* 378.6552, pp. 85–88.

Kim, E, S Naisbitt, Y. P. Hsueh, A Rao, A Rothschild, A. M. Craig, and M. Sheng (Feb. 1997). “GKAP, a novel synaptic protein that interacts with the guanylate kinase-like domain of the PSD-95/SAP90 family of channel clustering molecules.” *The Journal of Cell Biology* 136.3, pp. 669–678.

Kim, E. and M. Sheng (Oct. 2004). “PDZ domain proteins of synapses”. *Nature Reviews Neuroscience* 5.10, pp. 771–781.

Kim, J. H., D Liao, L. F. Lau, and R. L. Huganir (Apr. 1998). “SynGAP: a synaptic RasGAP that associates with the PSD-95/SAP90 protein family”. *Neuron* 20.4, pp. 683–691.

Kotaleski, J. H. and K. T. Blackwell (Apr. 2010). “Modelling the molecular mechanisms of synaptic plasticity using systems biology approaches”. *Nature Reviews Neuroscience* 11.4, pp. 239–251.

Kreienkamp, H.-J. (2008). “Scaffolding proteins at the postsynaptic density: shank as the architectural framework.” *Handbook of experimental pharmacology* 186, pp. 365–380.

Kruger, J, P Kipfer, P Kondratieva, and R Westermann (Nov. 2005). “A Particle System for Interactive Visualization of 3D Flows”. *IEEE Transactions on Visualization and Computer Graphics* 11.6, pp. 744–756.

Langevin, P. (1908). *On the theory of Brownian motion.* CR Acad. Sci.(Paris).

Le Novère, N. et al. (Jan. 2006). “BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems.” *Nucleic acids research* 34.Database issue, pp. D689–91.

Leaver-Fay, A. et al. (2011). “ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules.” *Methods in enzymology* 487.19, pp. 545–574.

Lee, H.-J. and J. J. Zheng (2010). “PDZ domains and their binding partners: structure, specificity, and modification”. *Cell communication and signaling : CCS* 8, p. 8.

Lim, I. A., D. D. Hall, and J. W. Hell (June 2002). “Selectivity and promiscuity of the first and second PDZ domains of PSD-95 and synapse-associated protein 102.” *The Journal of biological chemistry* 277.24, pp. 21697–21711.

Lintott, C., K. Schawinski, S. Bamford, A. Slosar, K. Land, D. Thomas, E. Edmondson, K. Masters, R. C. Nichol, and M. J. Raddick (2011). “Galaxy Zoo 1: data release of morphological classifications for nearly 900 000 galaxies”. *Monthly Notices of the Royal Astronomical Society* 410.1, pp. 166–178.

Lorensen, W. E. and H. E. Cline (1987). “Marching cubes: A high resolution 3D surface construction algorithm”. *SIGGRAPH Comput. Graph.* 21.4, pp. 163–169.

Love, P. J., P. V. Coveney, and B. M. Boghosian (Aug. 2001). “Three-dimensional hydrodynamic lattice-gas simulations of domain growth and self-assembly in binary immiscible and ternary amphiphilic fluids.” *Physical review E, Statistical, nonlinear, and soft matter physics* 64.2 Pt 1, p. 021503.

Lv, Z., A. Tek, F. Da Silva, C. Empereur-mot, M. Chavent, and M. Baaden (2013). “Game on, science - how video game technology may help biologists tackle visualization challenges.” *PLoS ONE* 8.3, e57990.

Maciel, A., T. Halic, Z. Lu, L. P. Nedel, and S. De (Sept. 2009). “Using the PhysX engine for physics-based virtual surgery with force feedback”. *The International Journal of Medical Robotics and Computer Assisted Surgery* 5.3, pp. 341–353.

Magrane, M. and U. Consortium (2011). “UniProt Knowledgebase: a hub of integrated protein data.” *Database : the journal of biological databases and curation* 2011, bar009.

Maguire, E. A., D. G. Gadian, I. S. Johnsrude, C. D. Good, J Ashburner, R. S. Frackowiak, and C. D. Frith (Apr. 2000). “Navigation-related structural change in the hippocampi of taxi drivers." *Proceedings of the National Academy of Sciences of the United States of America* 97.8, pp. 4398–4403.

Malek Mansour, M and F. Baras (1992). "Microscopic simulation of chemical systems". *Physica A: Statistical Mechanics and its Applications* 188.1, pp. 253–276.

Malinow, R. and R. C. Malenka (2002). "AMPA receptor trafficking and synaptic plasticity". *Annual Review of Neuroscience* 25, pp. 103–126.

Marangoni, A. G. (Apr. 2003). *Enzyme Kinetics.* A Modern Approach. John Wiley & Sons.

Maxwell, J. C. (1860). "V. Illustrations of the dynamical theory of gases.—Part I. On the motions and collisions of perfectly elastic spheres". *Philosophical Magazine Series 4* 19.124, pp. 19–32.

McCloskey, M. A. and M. M. Poo (Jan. 1986). "Rates of membrane-associated reactions: reduction of dimensionality revisited." *The Journal of Cell Biology* 102.1, pp. 88–96.

McGuffee, S. R. and A. H. Elcock (Mar. 2010). "Diffusion, crowding & protein stability in a dynamic molecular model of the bacterial cytoplasm." *PLoS Computational Biology* 6.3, e1000694.

Megias, M, Z Emri, T. F. Freund, and A. I. Gulyas (2001). "Total number and distribution of inhibitory and excitatory synapses on hippocampal CA1 pyramidal cells." *Neuroscience* 102.3, pp. 527–540.

Minezaki, Y., K. Homma, and K. Nishikawa (May 2007). "Intrinsically disordered regions of human plasma membrane proteins preferentially occur in the cytoplasmic segment." *Journal of Molecular Biology* 368.3, pp. 902–913.

Moreira, I. S., P. A. Fernandes, and M. J. Ramos (Jan. 2010). "Protein-protein docking dealing with the unknown." *Journal of computational chemistry* 31.2, pp. 317–342.

Morris, R. G., E Anderson, G. S. Lynch, and M Baudry (Mar. 1986). "Selective impairment of learning and blockade of long-term potentiation by an N-methyl-D-aspartate receptor antagonist, AP5." *Nature* 319.6056, pp. 774–776.

Morrison, A., M. Diesmann, and W. Gerstner (June 2008). "Phenomenological models of synaptic plasticity based on spike timing." *Biological cybernetics* 98.6, pp. 459–478.

Mulkey, R. M. and R. C. Malenka (Nov. 1992). "Mechanisms underlying induction of homosynaptic long-term depression in area CA1 of the hippocampus." *Neuron* 9.5, pp. 967–975.

Naisbitt, S, E Kim, J. C. Tu, B Xiao, C Sala, J Valtschanoff, R. J. Weinberg, P. F. Worley, and M. Sheng (July 1999). "Shank, a novel family of postsynaptic density proteins that binds to the NMDA receptor/PSD-95/GKAP complex and cortactin". *Neuron* 23.3, pp. 569–582.

Nehring, R. B., E Wischmeyer, F Doring, R. W. Veh, M. Sheng, and A Karschin (Jan. 2000). "Neuronal inwardly rectifying K(+) channels differentially couple to PDZ proteins of the PSD-95/SAP90 family." *The Journal of neuroscience : the official journal of the Society for Neuroscience* 20.1, pp. 156–162.

Niazi, M. and A. Hussain (Aug. 2011). "Agent-based computing from multi-agent systems to agent-based models: a visual survey". *Scientometrics* 89.2, pp. 479–499.

Nicoll, R. A., S. Tomita, and D. S. Bredt (Mar. 2006). "Auxiliary subunits assist AMPA-type glutamate receptors." *Science (New York, NY)* 311.5765, pp. 1253–1256.

Niethammer, M, E Kim, and M. Sheng (Apr. 1996). "Interaction between the C terminus of NMDA receptor subunits and multiple members of the PSD-95 family of membrane-associated guanylate kinases." *The Journal of neuroscience : the official journal of the Society for Neuroscience* 16.7, pp. 2157–2163.

Nooren, I. M. A. and J. M. Thornton (July 2003). "Diversity of protein-protein interactions." *The EMBO journal* 22.14, pp. 3486–3492.

Novák, B. and J. J. Tyson (Dec. 2008). "Design principles of biochemical oscillators." *Nature reviews Molecular cell biology* 9.12, pp. 981–991.

O'Keefe, J and J Dostrovsky (Nov. 1971). "The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat." *Brain Research* 34.1, pp. 171–175.

O'Keefe, J. and L Nadel (1978). "The hippocampus as a cognitive map". *Clarendon Press, Oxford.*

Oliveira, R. F., A. Terrin, G. Di Benedetto, R. C. Cannon, W. Koh, M. Kim, M. Zaccolo, and K. T. Blackwell (2010). "The role of type 4 phosphodiesterases in generating microdomains of cAMP: large scale stochastic simulations." *PLoS ONE* 5.7, e11725.

Opazo, P., S. Labrecque, C. M. Tigaret, A. Frouin, P. W. Wiseman, P. De Koninck, and D. Choquet (July 2010). "CaMKII triggers the diffusional trapping of surface AMPARs through phosphorylation of stargazin." *Neuron* 67.2, pp. 239–252.

Pacold, M. E., S Suire, O Perisic, S Lara-Gonzalez, C. T. Davis, E. H. Walker, P. T. Hawkins, L Stephens, J. F. Eccleston, and R. L. Williams (Dec. 2000). "Crystal structure and functional analysis of Ras binding to its effector phosphoinositide 3-kinase gamma." *Cell* 103.6, pp. 931–943.

Palade, G. E. (1954). "Electron microscope observations of interneuronal and neuromuscular synapses". *Anatomical Record* 118.2, pp. 335–336.

Paoletti, P. and J. Neyton (Feb. 2007). "NMDA receptor subunits: function and pharmacology". *Current opinion in pharmacology* 7.1, pp. 39–47.

Peng, J., M. J. Kim, D. Cheng, D. M. Duong, S. P. Gygi, and M. Sheng (May 2004). "Semiquantitative proteomic analysis of rat forebrain postsynaptic density fractions by mass spectrometry". *The Journal of biological chemistry* 279.20, pp. 21003–21011.

Petersen, J, X Chen, and L Vinade (2003). "Distribution of postsynaptic density (PSD)-95 and Ca2+/calmodulin-dependent protein kinase II at the PSD". *The Journal of ...*

Pettersen, E. F., T. D. Goddard, C. C. Huang, G. S. Couch, D. M. Greenblatt, E. C. Meng, and T. E. Ferrin (Oct. 2004). "UCSF Chimera–a visualization system for exploratory research and analysis." *Journal of computational chemistry* 25.13, pp. 1605–1612.

Pi, H. J., N. Otmakhov, F. El Gaamouch, D. Lemelin, P. De Koninck, and J. Lisman (Aug. 2010). "CaMKII control of spine size and synaptic strength: role of phosphorylation states and nonenzymatic action." *Proceedings of the National Academy of Sciences of the United States of America* 107.32, pp. 14437–14442.

Pierce, B. G., Y. Hourai, and Z. Weng (Sept. 2011). "Accelerating Protein Docking in ZDOCK Using an Advanced 3D Convolution Library". *PLoS ONE* 6.9, e24657.

Purkinje, J. E. (1837). "Neueste Untersuchungen aus der Nerven und Hirn Anatomie". *Bericht über die Versammlung deutscher Naturforscher und Aertze in Prag.* Pp. 177–180.

R Development Core Team, R (Jan. 2013). "R: A Language and Environment for Statistical Computing". GPL–2 — GPL–3.

Racca, C., F. A. Stephenson, P. Streit, J. D. B. Roberts, and P. Somogyi (2001). "NMDA Receptor Content of Synapses in Stratum Radiatum of the Hippocampal CA1 Area". *Results and problems in cell ...* 20.7, pp. 2512–2522.

Rall, W (Nov. 1959). "Branching dendritic trees and motoneuron membrane resistivity." *Experimental Neurology* 1, pp. 491–527.

Rall, W (Mar. 1962). "Theory of physiological properties of dendrites." *Annals of the New York Academy of Sciences* 96, pp. 1071–1092.

Rao, C. V., D. M. Wolf, and A. P. Arkin (Nov. 2002). "Control, exploitation and tolerance of intracellular noise." *Nature* 420.6912, pp. 231–237.

Rigaut, G, A Shevchenko, B Rutz, M Wilm, M Mann, and B Seraphin (Oct. 1999). "A generic protein purification method for protein complex characterization and proteome exploration." *Nature biotechnology* 17.10, pp. 1030–1032.

Rizzo, A., A. Hartholt, B. Rothbaum, J. Difede, C. Reist, D. Kwok, A. Leeds, J. Spitalnick, T. Talbot, and T. Adamson (2014). "Expansion of a VR Exposure Therapy System for Combat-Related PTSD to Medics/Corpsman and Persons Following Military Sexual Trauma". *Medicine Meets Virtual Reality 21: NextMed/MMVR21* 196, p. 332.

Rost, B., G. Yachdav, and J. Liu (July 2004). "The PredictProtein server." *Nucleic acids research* 32.Web Server issue, W321–6.

Rostaing, P., E. Real, L. Siksou, J.-P. Lechaire, T. Boudier, T. M. Boeckers, F. Gertler, E. D. Gundelfinger, A. Triller, and S. Marty (Dec. 2006). "Analysis of synaptic ultrastructure without fixative using high-pressure freezing and tomography". *The European journal of neuroscience* 24.12, pp. 3463–3474.

Roy, A., A. Kucukural, and Y. Zhang (2010). "I-TASSER: a unified platform for automated protein structure and function prediction." *Nature protocols* 5.4, pp. 725–738.

Rubin, J. and M. Wechselberger (July 2007). "Giant squid-hidden canard: the 3D geometry of the Hodgkin-Huxley model." *Biological cybernetics* 97.1, pp. 5–32.

Sanabria, H, M. Swulius, S. J. Kolodziej, and J Liu (2009). "βCaMKII regulates actin assembly and structure". *Journal of Biological ...*

Sasaki, Y. F. et al. (Apr. 2002). "Characterization and comparison of the NR3A subunit of the NMDA receptor in recombinant systems and primary cortical neurons." *Journal of neurophysiology* 87.4, pp. 2052–2063.

Sauro, H. M. and B. N. Kholodenko (Sept. 2004). "Quantitative analysis of signaling networks." *Progress in biophysics and molecular biology* 86.1, pp. 5–43.

Saxton, M. J. (Feb. 1994). "Anomalous diffusion due to obstacles: a Monte Carlo study." *Biophysical Journal* 66.2 Pt 1, pp. 394–401.

Saxton, M. J. (Mar. 1996). "Anomalous diffusion due to binding: a Monte Carlo study." *Biophysical Journal* 70.3, pp. 1250–1262.

Schacter, D. L. and E. Tulving (1994). *Memory systems 1994*.

Schaff, J, C. C. Fink, B Slepchenko, J. H. Carson, and L. M. Loew (Sept. 1997). "A general computational framework for modeling cellular structure and function." *Biophysical Journal* 73.3, pp. 1135–1146.

Schiegg, A, W Gerstner, R Ritz, and J. L. van Hemmen (Sept. 1995). "Intracellular Ca2+ stores can account for the time course of LTP induction: a model of Ca2+ dynamics in dendritic spines." *Journal of neurophysiology* 74.3, pp. 1046–1055.

Schnell, S and T. E. Turner (June 2004). "Reaction kinetics in intracellular environments with macromolecular crowding: simulations and rate laws." *Progress in biophysics and molecular biology* 85.2-3, pp. 235–260.

Schreiber, G, G Haran, and H. X. Zhou (Mar. 2009). "Fundamental Aspects of Protein-Protein Association Kinetics". *Chemical reviews* 109.3, pp. 839–860.

Scoville, W. B. and B Milner (Feb. 1957). "Loss of recent memory after bilateral hippocampal lesions." *Journal of neurology, neurosurgery, and psychiatry* 20.1, pp. 11–21.

Sessoms-Sikes, S, Y Honse, and D Lovinger (2005). "CaMKII [alpha] enhances the desensitization of NR2B-containing NMDA receptors by an autophosphorylation-dependent mechanism". *Molecular and Cellular ...*

Shannon, P., A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker (Nov. 2003). "Cytoscape: a software environment for integrated models of biomolecular interaction networks." *Genome research* 13.11, pp. 2498–2504.

Sharpe, J, C. J. Lumsden, and N Woolridge (2008). "In Silico: 3D Animation and Simulation of Cell Biology with Maya and MEL - Jason Sharpe, Charles John Lumsden, Nicholas Woolridge - Google Books".

Shaw, D. E., R. O. Dror, J. K. Salmon, J. P. Grossman, K. M. Mackenzie, J. A. Bank, C. Young, M. M. Deneroff, B. Batson, and K. J. Bowers (2009). "Millisecond-scale molecular dynamics simulations on Anton". Pp. 1–11.

Sheng, M. and E Kim (June 2000). "The Shank family of scaffold proteins". *Journal of cell science* 113 (Pt 11), pp. 1851–1856.

Sheng, M. and E. Kim (Nov. 2011). “The Postsynaptic Organization of Synapses.” *Cold Spring Harbor perspectives in biology.*

Shoichet, B. K. and I. D. Kuntz (Sept. 1991). “Protein docking and complementarity.” *Journal of Molecular Biology* 221.1, pp. 327–346.

Shontz, S. M. and D. M. Nistor (2013). “CPU-GPU Algorithms for Triangular Surface Mesh Simplification”. *link.springer.com.* Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 475–492.

Shouval, H. Z., M. F. Bear, and L. N. Cooper (Aug. 2002). “A unified model of NMDA receptor-dependent bidirectional synaptic plasticity.” *Proceedings of the National Academy of Sciences of the United States of America* 99.16, pp. 10831–10836.

Smoluchowski, M. V. (1916). “Drei Vortrage uber Diffusion, Brownsche Bewegung und Koagulation von Kolloidteilchen”. *Zeitschrift fur Physik* 17, pp. 557–585.

Song, S, K. D. Miller, and L. F. Abbott (Sept. 2000). “Competitive Hebbian learning through spike-timing-dependent synaptic plasticity.” *Nature Neuroscience* 3.9, pp. 919–926.

Spacek, J and M Hartmann (1983). “Three-dimensional analysis of dendritic spines. I. Quantitative observations related to dendritic spine and synaptic morphology in cerebral and cerebellar cortices.” *Anatomy and Embryology* 167.2, pp. 289–310.

Squire, L. R. (Apr. 1992). “Memory and the hippocampus: a synthesis from findings with rats, monkeys, and humans.” *Psychological review* 99.2, pp. 195–231.

Stein, V., D. R. C. House, D. S. Bredt, and R. A. Nicoll (July 2003). “Postsynaptic density-95 mimics and occludes hippocampal long-term potentiation and enhances long-term depression.” *The Journal of neuroscience : the official journal of the Society for Neuroscience* 23.13, pp. 5503–5506.

Sugiyama, Y., I Kawabata, and K Sobue (2005). “Determination of absolute protein numbers in single synapses by a GFP-based calibration technique”. *Nature Methods.*

Sumioka, A., D. Yan, and S. Tomita (June 2010). “TARP phosphorylation regulates synaptic AMPA receptors through lipid bilayers.” *Neuron* 66.5, pp. 755–767.

Takeuchi, M, Y Hata, K Hirao, and A Toyoda (1997). “SAPAPs”. *Journal of Biological ...*

Tolle, D. P. and N. Le Novère (2010). “Meredys, a multi-compartment reaction-diffusion simulator using multistate realistic molecular complexes.” *BMC systems biology* 4, p. 24.

Tomita, M et al. (Jan. 1999). “E-CELL: software environment for whole-cell simulation.” *Bioinformatics (Oxford, England)* 15.1, pp. 72–84.

Tovchigrechko, A. and I. A. Vakser (July 2006). “GRAMM-X public web server for protein-protein docking.” *Nucleic acids research* 34.Web Server issue, W310–4.

Triller, A. and D. Choquet (Aug. 2008). “New concepts in synaptic biology derived from single-molecule imaging”. *Neuron* 59.3, pp. 359–374.

Trommald, M and G Hulleberg (Jan. 1997). “Dimensions and density of dendritic spines from rat dentate granule cells based on reconstructions from serial electron micrographs.” *The Journal of comparative neurology* 377.1, pp. 15–28.

Tu, J. C., B Xiao, J. P. Yuan, A. A. Lanahan, K Leoffert, M Li, D. J. Linden, and P. F. Worley (Oct. 1998). “Homer binds a novel proline-rich motif and links group 1 metabotropic glutamate receptors with IP3 receptors.” *Neuron* 21.4, pp. 717–726.

Tu, J. C. et al. (July 1999). “Coupling of mGluR/Homer and PSD-95 complexes by the Shank family of postsynaptic density proteins.” *Neuron* 23.3, pp. 583–592.

Uchino, S., H. Wada, S. Honda, Y. Nakamura, Y. Ondo, T. Uchiyama, M. Tsutsumi, E. Suzuki, T. Hirasawa, and S. Kohsaka (May 2006). “Direct interaction of post-synaptic density-95/Dlg/ZO-1 domain-containing synaptic molecule Shank3 with GluR1 alpha-amino-3-hydroxy-5-methyl-4-isoxazole propionic acid receptor.” *Journal of neurochemistry* 97.4, pp. 1203–1214.

Upadhyay, S. K. (Nov. 2010). *Chemical Kinetics and Reaction Dynamics.* Springer Verlag.

Uversky, V. N. (June 2013). “The most important thing is the tail: multitudinous functionalities of intrinsically disordered protein termini.” *FEBS letters* 587.13, pp. 1891–1901.

Valtschanoff, J. G. and R. J. Weinberg (Feb. 2001). “Laminar organization of the NMDA receptor complex within the postsynaptic density”. *The Journal of neuroscience : the official journal of the Society for Neuroscience* 21.4, pp. 1211–1217.

Van Kampen, N. G. (1992). *Stochastic processes in physics and chemistry.* Vol. 1. Elsevier.

Vullhorst, D., J. Neddens, I. Karavanova, L. Tricoire, R. S. Petralia, C. J. McBain, and A. Buonanno (Sept. 2009). “Selective expression of ErbB4 in interneurons, but not pyramidal cells, of the rodent hippocampus.” *The Journal of neuroscience : the official journal of the Society for Neuroscience* 29.39, pp. 12255–12264.

Waldeyer-Hartz, W. von (1891). *Ueber einige neuere Forschungen im Gebeite der Anatomie des Centralnervensystems.* G. Thieme.

Walikonis, R. S., O. N. Jensen, M Mann, D. W. Provance, J. A. Mercer, and M. B. Kennedy (June 2000). “Identification of proteins in the postsynaptic density fraction by mass spectrometry.” *The Journal of neuroscience : the official journal of the Society for Neuroscience* 20.11, pp. 4069–4080.

Wang, H., F. Qiao, and B. Zhou (Dec. 2013). “Multi-Feature Metric-Guided Mesh Simplification”. *link.springer.com.* New Delhi: Springer India, pp. 535–542.

Wei, X., W. Li, K. Mueller, and A. Kaufman (2002). “Simulating fire with texture splats”. *Visualization, 2002 VIS IEEE*, pp. 227–234.

Wenthold, R. J., R. S. Petralia, I. I. Blahos J, and A. S. Niedzielski (Mar. 1996). “Evidence for multiple AMPA receptor complexes in hippocampal CA1/CA2 neurons.” *The Journal of neuroscience : the official journal of the Society for Neuroscience* 16.6, pp. 1982–1989.

White, D. A., A. K. Buell, T. P. J. Knowles, M. E. Welland, and C. M. Dobson (Apr. 2010). “Protein aggregation in crowded environments.” *Journal of the American Chemical Society* 132.14, pp. 5170–5175.

Whitford, K. L., P. Dijkhuizen, F. Polleux, and A. Ghosh (2002). “Molecular control of cortical dendrite development.” *Annual Review of Neuroscience* 25, pp. 127–149.

Wigstrom, H, B Gustafsson, Y. Y. Huang, and W. C. Abraham (Feb. 1986). “Hippocampal long-term potentiation is induced by pairing single afferent volleys with intracellularly injected depolarizing current pulses.” *Acta physiologica Scandinavica* 126.2, pp. 317–319.

Willows, A. O. and G Hoyle (Dec. 1969). “Neuronal network triggering a fixed action pattern.” *Science (New York, NY)* 166.3912, pp. 1549–1551.

Wiltgen, M. and G. P. Tilz (2009). “Homology modelling: a review about the method on hand of the diabetic antigen GAD 65 structure prediction.” *Wiener medizinische Wochenschrift (1946)* 159.5-6, pp. 112–125.

Wolf, D. M. and A. P. Arkin (Apr. 2003). “Motifs, modules and games in bacteria.” *Current opinion in microbiology* 6.2, pp. 125–134.

Wolfram, S (1984). “Cellular automata as models of complexity”. *Nature* 311.5985, pp. 419–424.

Xiao, B, J. C. Tu, and P. F. Worley (June 2000). “Homer: a link between neural activity and glutamate receptor function.” *Current opinion in neurobiology* 10.3, pp. 370–374.

Xu, D, Y Xu, and E. C. Uberbacher (July 2000). “Computational tools for protein modeling.” *Current protein & peptide science* 1.1, pp. 1–21.

Xu, W. (Apr. 2011). “PSD-95-like membrane associated guanylate kinases (PSD-MAGUKs) and synaptic plasticity.” *Current opinion in neurobiology* 21.2, pp. 306–312.

Young, M. and P. Carroad (1980). “Estimation of diffusion coefficients of proteins”. *Biotechnology and Bioengineering.*

Zhang, W, L Vazquez, M Apperson, and M. B. Kennedy (Jan. 1999). “Citron binds to PSD-95 at glutamatergic synapses on inhibitory neurons in the hippocampus.” *The Journal of neuroscience : the official journal of the Society for Neuroscience* 19.1, pp. 96–108.

Zhou, H.-X., G. Rivas, and A. P. Minton (June 2008b). “Macromolecular Crowding and Confinement: Biochemical, Biophysical, and Potential Physiological Consequences *”. *Annual review of biophysics* 37.1, pp. 375–397.

Zhou, H.-X., G. Rivas, and A. P. Minton (2008a). “Macromolecular crowding and confinement: biochemical, biophysical, and potential physiological consequences.” *Annual review of biophysics* 37, pp. 375–397.

Zimmerman, S. B. and S. O. Trach (Dec. 1991). “Estimation of macromolecule concentrations and excluded volume effects for the cytoplasm of Escherichia coli.” *Journal of Molecular Biology* 222.3, pp. 599–620.