Università
Ca'Foscari
Venezia

Corso di Laurea magistrale

in Science and Technology of Bio and Nanomaterials

(LM-53 - Scienza e ingegneria dei materiali)

Final Thesis

A network topology approach to the relation between painful disorders and mutations in sodium channel proteins

Supervisor
Ch. Prof. Achille Giacometti

Assistant supervisor
Ch. Prof.ssa Marta Simeoni

Graduand
Alberto Antonio Toffano
Matricolation number
840040

Academic Year
2018 / 2019

CA' FOSCARI UNIVERSITY OF VENICE

MASTER'S DEGREE THESIS

A network topology approach to the relation between painful disorders and mutations in sodium channel proteins

Author:
Alberto TOFFANO

Supervisors:
Dr. Achille GIACOMETTI
Dr.ssa Marta SIMEONI

A thesis submitted in fulfillment of the requirements for the degree of Science and Technology of Bio and Nanomaterials

June 20, 2019

CA' FOSCARI UNIVERSITY OF VENICE

Abstract

Science and Technology of Bio and Nanomaterials
Department of Molecular Sciences and Nanosystems

A network topology approach to the relation between painful disorders and mutations in sodium channel proteins

by Alberto TOFFANO

The purpose of this thesis was to combine a number of different computational techniques to understand a major problem in neurosciences. Notably we discuss the treatment of a membrane protein expressed in the human peripheral nervous system and how mutations within its primary sequence may lead to the onset of painful neuropathies. To address this question, it is necessary to get structural information relating to each mutation. To obtain such data in a short time, a computational approach was adopted, hinging upon homology modeling. Using three models with a known structure of homologous proteins, structural models were produced for each mutation. The generated set of models was analyzed through graph-theory complemented by machine learning techniques, looking for a common patterns relating specific mutations to patological diseases, and able to discriminate them from mutations that do not alter the correct functionality of the protein. Our results suggest that the use of the graph kernel techniques and Dominant set clustering are the optimal tools to identify common topological patterns among pain-related mutations in over 90% of the studied models.

Acknowledgements

I would first like to thank my thesis advisors University Researcher Simeoni Marta of the Faculty of Computer Science at Ca' Foscari and Professor Giacometti Achille of the Faculty of Science and Technology of Bio and Nanomaterials at Ca' Foscari, Venice. Only the patience and the time devoted to me by my supervisors made it possible to write this paper. I would also like to thank the experts who were involved in the validation survey for this thesis project: Marchi Margherita, Salvi Erika and Lauria Giuseppe of Institute of Neurology 'Carlo Besta', Milan. Without their contribution to the collection of data and their availability for meetings and clarifications, this thesis could not have been written. A special thanks also to the colleagues who helped me with their computer knowledge, contributing to the writing of programs, otherwise impossible for me. Thanks to Giacomo Chiarot, Daniele Crosariol and Fabio Rosada.

I would like to thank my friends and colleagues, without whom I certainly wouldn't have shared so many good moments and difficult exam sessions. I owe everyone a lot, which cannot be described here in a few lines. Probably the best companions I could meet, thank you Valeria, Virginia, Giovanna, Francesca, Aurelio, Andrea, Davide, Giulia and Nicola. How can I forget my roommates Ana, Anna, Annaclaudia e Caterina? who had the patience to put up with me for these years or my work colleagues? the people met were really many and unfortunately it is impossible for me to be able to thank everyone adequately. Finally, I must express my gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study.
Thank you.

Alberto

Contents

Chapter 1

Introduction

The application of machine learning and graph theory techniques to neuroscience has witness a resurgence of interest in the last decade due the new tools that became recently available [66]. However, their application to the field of painful neurophaties has been very scanty, also because of the limited experimental results available in this area. The aim of the present thesis is to show how a combination of different and complementary computational techniques, including computational biology and network analysis, can help to shed new light to this field.

Painful neuropathies nowadays afflict millions of people around the world and their mechanisms are still unclear to a large extent. Partially for this reason, available therapies are not able to satisfactorily alleviate those suffering from these pathologies [8]. The factors leading to the emergence of these problems are complex and multifacets (environmental factors, genetic predisposition and diet to name a few). However, a common point is the way of communicating pain, more in detail, it is known [43] that the pain stimulus is mediated by a class of membrane proteins denoted as the voltage-gated sodium channels. To this family belongs NaV1.7, which is of extreme interest to humans. It is highly expressed in the peripheral nervous system, mediating the signal between the peripheries of the body and the central nervous system (the brain). It has been speculated [22] that gain-of-function mutations are in direct connection with the onset of painful neuropathies, such as inherited erythromelalgia (IEM), paroxysmal extreme pain disorder (PEPD) and small fibre neuropathy (SFN). Investigating how these mutations as compared to others that do not involve functional alterations, modify the structure and the kinetics of this protein, can be of great help to understand what are the mechanics that govern these problems. A classical pathway [64] to derive structural information on each single mutation would require the use of adequate and modified cell cultures where over-express the modified gene of interest, protein purification and concentration for subsequent structural analyzes by NMR, XDR or cryo-EM. This process, however, is both time consuming and expensive, hence a guideline toward an optimal experimental probe would be desirable. Computational techniques can be of great help in this respect as they offer increasingly reliable tools for the development of representative models of protein structures. In particular, a homology modeling technique [38], based on the observation that proteins having the same biological function tend to preserve a common structure, has proven to be a very effective tool that can be used in several different biological context. The

underlying idea is that, given the structure of a homologous protein, it is possible to generate the structural model of a protein for which only the function and the primary sequence are known, by aligning the common amino acid traits, thus obtaining a rough model presumably having a geometry close to that of the real native structure, This model can then be refined by energy minimization to determine which is the set of coordinates of the native state that minimizes the free energy of the system. In this way it is possible to generate in a relatively short time a set of models representative of each mutation, that can subsequently be subjected to further analysis. In this thesis, we built upon this idea and tried to determine whether there was a pattern linking gain-of-function mutations to each other compared to another set of models representing mutations not directly related to functional disorders. This idea was patterned after a previous study by a research group based at the Carlo Besta Neurological Institute in Milan [32], that recently tackled this issue using data coming from their own clinical counterpart.

To reach this goal, we exploited graph-theory complemented by machine learning techniques, by representing proteins through their topological graphs. Here a protein can be seen as a set of amino acids (nodes) connected to each other by edges reflecting their internal complex interactions, and hence can be represented by a complex network. First, this analysis approach allows for reducing the complexity of a three-dimensional system to a two-dimensional representation that preserves the topological information. And on the other hand, it helps identifying common patterns impossible or hardly visible in their three dimensional structure.

The basis of our study relies on past work by the Carlo Besta research group, but we significantly improved it both from the methodological viewpoint and the extension of the studied cases. Briefly, in [32], the authors generated a set of models representative of some gain-of-function mutations and mutations not directly related to neuropathies, using the 3RVY homologous protein, deposited structure of the voltage-dependent sodium channel, as template *Arcobacter bultzeri*). The models were then subjected to an energy minimization to allow them to converge to their native structures and then transformed into the corresponding Residue Interaction Networks (RINs). The RINs so obtained were then compared on the basis of the calculation of some metrics (such as Betweenness Centrality, Edge Betweenness, Degree, Clustering Coefficient, Closeness Centrality and Eccentricity). On this basis, it was then speculated that the betweenness centrality index is the most sensitive to pain-related mutations.

In this work, we used the same mutations used by Dimos et al [32], applied to three models generated by using three different templates. The first template was the same used in [32] and was meant as a benchmark. The other two templates derived from two other homologous proteins, in order to produce models deriving from templates with greater identities with the starting sequences and to probe an additional space of conformations. All the structures generated have been subjected to energy minimization and transformed into their corresponding RINs, the metrics have been calculated and

the betweenness centrality index resulted again to be the most sensitive index for the distinction between gain-of-function mutations and mutations not involved in alteration of functionality, as stated in the reference paper. An important novelty of our study was the use of an additional route that has been explored by resorting to the use of Graph Kernels [35, 67], in particular the Weisfeiler-Lehman kernel [63], to compare the whole RINs topology, rather than using the punctual evaluation of a metric. A successive application of the Dominant Set clustering method [10] allowed us to evaluate the ability of the kernel methods to discriminate between gain-of-function mutations and mutations not involved in alteration of functionality. The results obtained are encouraging and seem to recognize in this approach an instrument of greater selectivity and reliability.

The thesis is organized as follows: Chapter 2 gives a brief introduction on proteins in general, their role inside the peripheral nervous system, how they propagate the signals and how they are related to the onset of painful neuropathies. Chapter 3 describes the computational tools employed in the thesis and shows the workflow followed in this study. Chapter 4 presents two main related works: the one by Dimos et al [32], and another closely related work that addressed the same issues with a different approach. Chapter 5 illustrates the main results of the thesis and finally Chapter 6 draws some concluding remarks.

Chapter 2

Insight into the Context

This Chapter aims to introduce the reader to the roles of proteins in the human pheriperal nervous system. We start with a brief introduction to proteins and their structure, and proceed afterwords by illustrating the tools that have been our eyes into this world, emphasizing their merits and limits. We then introduce the human's peripheral nervous system, the role of proteins in this system and how they are connected to the onset of neuropathic pain disorders.

2.1 Proteins

Proteins play essential roles in most biological processes, they could be involved in chemical reactions as enzymes; others like hemoglobin and myoglobin are involved in transport and storage processes. Also some proteins allow and median comunication between cells or are involved in control of growth and differentiation of cells. Proteins are composed of twenty amino acids, folding into unique three-dimensional structures that are strictly related to their biological functions. The onset of malfunctions resulting from the development of diseases or alterations in the amino acid sequence in sensitive traits, can cause fatal complications. Thus understanding the structures of proteins and their related functions in various biological mechanisms are important subjects of studies. Proteins are polymers made of monomers former from 20 amino acids. These mono-mers share a similar structure, all have a central carbon with functional groups on the sides: amino and carboxylic groups. This central carbon is called alpha (it is still part of the polypeptide chain) and following the Greek alphabet are then named the other carbon atoms constituting the side chain of each residue (if present). C_α also constitutes a chiral center for all amino acids, except in the special case of glycine whose side chain is made up of a hydrogen. Outside of these common points, the properties of its side chain characterize the behavior of each amino acid. For example, valine and leucine have apolar side chains that give these amino acids a hydrophobic character. A different case is that of arginine and lysine, which have charged side chains, and have a basic character at physiological pH. A protein polypeptide chain is generated by joining amino acids end-to-end through peptide bonds. In general, proteins can be classified into three types: fibrous, membrane or globular. The classes of fibrous proteins contain collagen, a-keratin and other proteins involved in many structural roles.

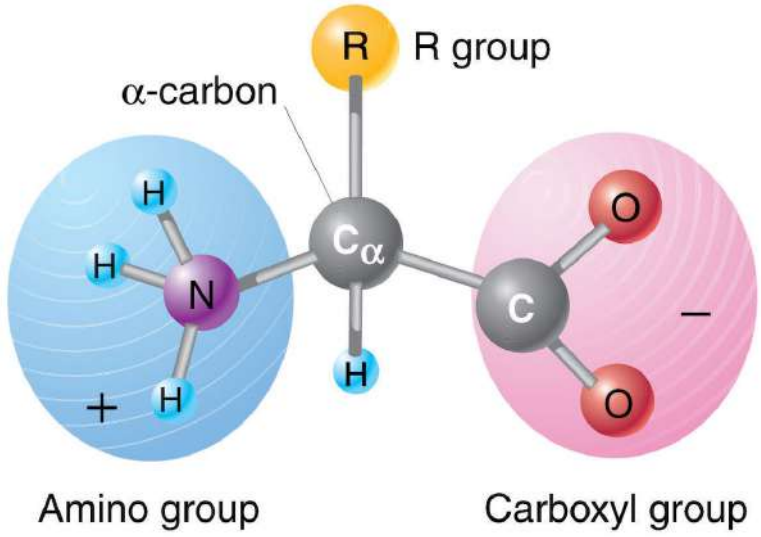

FIGURE 2.1: Amino acid scaffold [59].

As in tendon and bone formation or compose hair and skin. Membrane proteins reside in cellular membranes, where they mediate the exchange of the molecules and information across cellular boundaries. Most proteins in the cytoplasm of cells are soluble in aqueous environment and adopt compact globular morphology. These globular proteins are the catalysis for virtually all biochemical reactions in living cells. The basic structure of the peptide bond is shown in the figures 2.1 and 2.2.

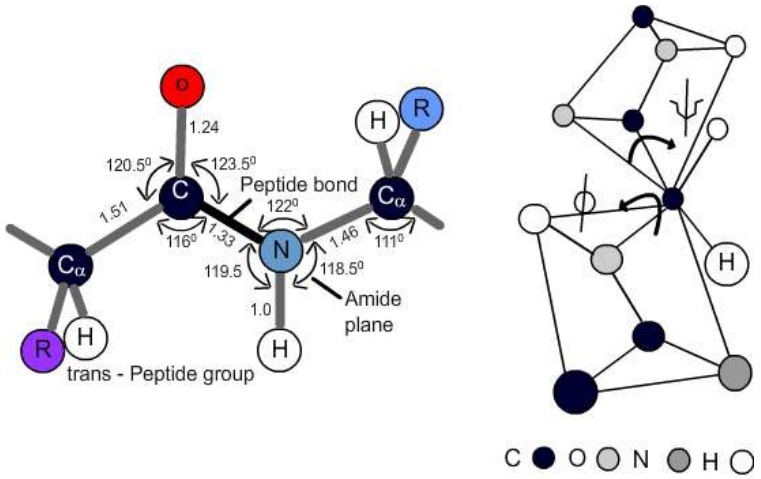

FIGURE 2.2: Peptide bond [45].

The peptide bond by its nature brings with it a hybrid character between single and double bond. This results in a certain rigidity in the plane of the bond, allowing rotation only around the C_α. The angles of rotation have been called ϕ and ψ, around N-C_α and C_α-C bond, respectively (figure 2.3).

As mentioned before, the constituent elements of proteins are 20 amino acids, defined by the properties of their side chain and can be divided into

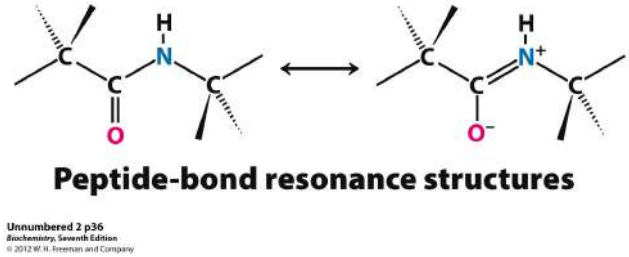

FIGURE 2.3: planarity due to peptide-bond [Biochemistry, Seventh Edidtion, 2012 W.H. Freeman and Company].

two main groups: hydrophobic and polar (figure 2.4). It is these characters of the side chains that guide the protein to rearrange itself in its native structure and to determine its role within the biological sector; in particular, it is the hydrophobicity that characterizes the side chains that play a pivotal role in protein folding [11].

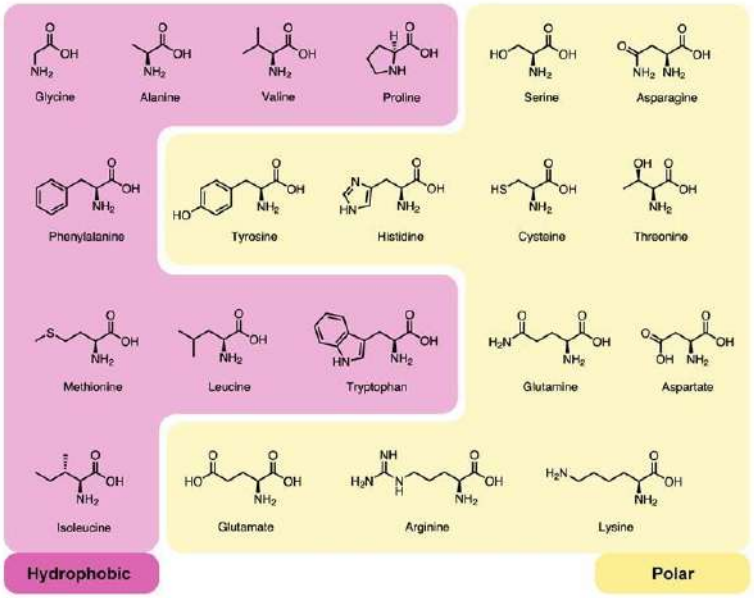

FIGURE 2.4: Amino acids list, classified based on their polar or hydrophobic behavior [55].

The central problem associated with forming of the hydrophobic core from the protein side chains is that the main chain of the protein is highly polar, but it must be also buried in the interior of the protein. This problem is solved in a very elegant way by the formation of regular secondary structure within the interior of the protein molecule. The origin of these structures can be tracked back to the presence of hydrogen bond donor and acceptor, NH and CO. Thus, by forming regular hydrogen bonds between NH and CO, the protein backbone can be neutralized in the protein interior. Such a secondary structure is usually one of two types: α-helices or β-sheets.

2.1.1 α-Helices

The most abundant types of secondary structures are α-helices. The first one to deduce this (and also the other main secondary structure, β-sheet) was Linus Pauling. He and his coworker were able to predict their existance already one decade before the structures of entire proteins were first revealed by x-ray crystallography [21]. An example of α-helix is reported in the figure 2.5(left). This configuration is characterized to have 3.6 residues each turn, that means that there is one residue every 100 degree of rotation. Each residue is translated 1.5 Å along the helix axis, which gives a vertical distance of 5.4 Å between structurally equivalent atoms in a turn. The hydrogen bonds between residues are established between each i-th element of the sequence and the i+4-th residual in cascade. The α-helix above described is not the only helical-like structure, in biological life were reported other two structures: 3_{10}-helix and π-helix. These structures differ from α-helix because they are stabilized by a hydrogen bond between i, i+3 and i, i+5 residues respectively.

2.1.2 β-Sheet

Beta sheets are built up from beta strands which are normally from 5 to 10 residues long. The side chains in a beta strand point alternatively up and under the beta sheet. Usually, beta sheets have their beta strands either parallel or antiparallel, in some samples, they have both, but they are less common (figure 2.5 right).

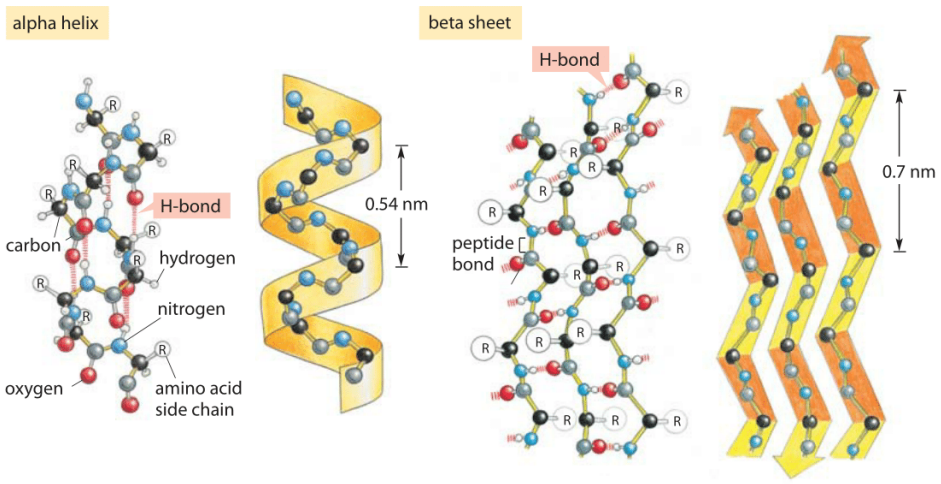

FIGURE 2.5: Representation of α-helix and β-sheet [41].

Since it was first proposed that the amino acid sequence is sufficient to determine the three-dimensional folded structure of the protein, significant efforts have been done to investigate the fundamental protein folding mechanism and the physical driving force on protein folding from an amino acid sequence. As previuos, the driving force to protein folding is the hydrophobic character of side chain. The explanation of this is to be sought in how

these chains interact with the surrounding environment. Hydrophobic interactions pack non-polar residues to minimize unfavorable contact with water. This process can be seen as a ball rolling down a mountain, the bottom of the mountain represents its free energy minimum (figure 2.6).

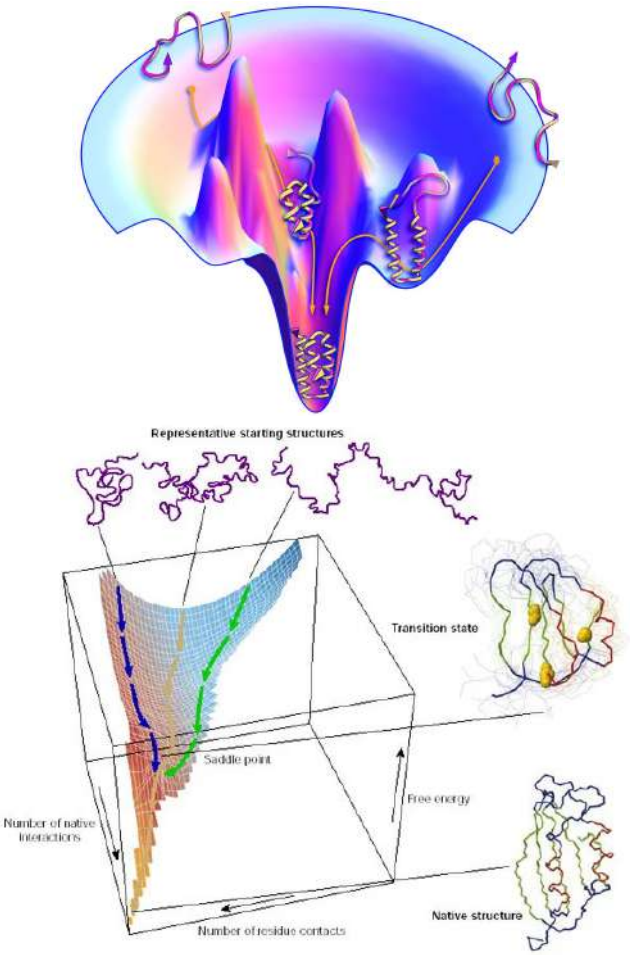

FIGURE 2.6: example of energy landscape of conformation[17][18].

2.2 Overview on physical tools for determining the protein structure

Tertiary structure determination of biomolecules at atomic resolution provides essential insights into the function of bioactive molecules. X-ray crystallography and nuclear magnetic resonance (NMR) spectroscopy have been the primary methods over the past few decades to obtain high-resolution

structures. More recently, the rapid technological growth of cryo-electron microscopy has seen this technique emerge as a third major approach to solve bio-macromolecular structures at atomic resolution. The area of application of these techniques is to be considered complementary. Information obtainable through NMR spectroscopy, in some cases cannot be achieved via X-ray diffraction or cryo-EM. As an example solution NMR offers a number of distinct features for structural biology studies: 1)Dynamics of protein folding, structural fluctuations, internal mobility and chemical exchange of target molecules can be investigated over a wide range of timescales. 2) Studies of protein-protein or protein–ligand interactions can be performed under physiological conditions. The affinity and the location of the interaction sites between the target protein and its binding partner molecules can be determined accurately and sensitively even if the interaction is very weak. A great successful application of NMR to study biomolecular dynamics has been carried out on intrinsic disordered proteins (IDPs) such as zinc finger proteins that usually do not yield crystals, which provide sufficient quality X-ray diffraction data for high-resolution structure determination. Moreover, these proteins are too small or too flexible to obtain strong contrast images by modern cryo-EM analysis. [64] Nonetheless, these experimental tools have limitations in determining the structures of some proteins and are very time consuming and expensive. For example, some proteins are very difficult to crystallize, which hampers the structure determination by x-ray crystallography. NMR spectroscopy also has limitations, for example, in that currently it is applicable only to proteins with less than about 30 kDa. When the molecular weight of the target protein exceeds this level measurement and assignment of the protein NMR signals become difficult owing to the increasing degeneration and line-broadening of the signals. One other example is the structure determination of membrane proteins. Membrane proteins are located in the lipid bilayer and of importance in the transport of the proteins across the membrane and many other processes. These membrane proteins have very different environment from that of other soluble proteins. While other cellular proteins have polar environment, which is aqueous, membrane proteins reside in the lipid bilayer which is hydrophobic. Thus, the structure determination of the membrane proteins by conventional experimental tools is particulary challenging. With the developments in genome project, the identification of the protein sequences has been accelerated, but the speed of the structure determination and functional assignments has been much slower. [29] In light of these technological achievements, and thanks to their applications in the medical context to date, extensive databases are available to the scientific community that links mutations along the amino acid sequence with the onset of human diseases. It remains an obstacle to find a reliable way of probing and easily describing the changes resulting from mutations. Computational tools (CT) can help to overcome this obstacle, in an area that underpins a broad spectrum of disciplines, including chemistry and biochemistry, catalysis, materials science, nanoscience, energy and environmental science, and geosciences. Taking advantage of the tools made available by technological progress, we focused this study on the search of points in common

between the mutations associated with neuropathies and those that do not involve functional changes, in the context of a membrane protein shown to be closely linked to these disorders. This protein belongs to the class of ion channels. Many conditions such as epilepsy, pain syndromes, cystic fibrosis, and many others have been attributed to mutations that lead to alteration of ion channel function in some way. In order to be able to offer some kind of drug-based treatment for these conditions in the future, an improvement of molecular understanding of the effects of the mutations will provide the most rational route to a successful. Outcome of this objective, to achieved this we decided to resort to the use of tools such as homology modelling and molecular dynamics (MD) simulations. [43][37]

2.3 Peripheral Nervous System

The peripheral nervous system (PNS) consists of all neurons that exist outside the brain and spinal cord. This includes long nerve fibers containing bundles of axons as well as ganglia made of neural cell bodies. The peripheral nervous system connects the central nervous system (CNS) made of the brain and spinal cord to various parts of the body and receives input from the external environment as well. [16]

There are two types of cells in the peripheral nervous system, carrying information to (sensory nervous cells) and from (motor nervous cells) the central nervous system. Cells of the sensory nervous system send information to the CNS from internal organs or from external stimuli. Motor nervous system cells carry information from the CNS to organs, muscles, and glands. [5]

In our body, peripheral nerve cells of different shape and thickness connect the brain to the rest of the body and allow it to decode signals coming from the outside world (see figures 2.7, 2.8 and 2.9). Most of these neurons are unidirectional: either send messages to the brain about what happens in or around the periphery, or they reply to a brain's signal outward to the muscles and other cells. Nevertheless, human's anatomy including least-evolved small diameter neurons called C-fibers still working in ancient and undisciplined ways. As an example, they are bidirectional: in addition to encoding and transmitting messages inward to the spinal cord and the brain when they recognized dangerous stimuli, they also convey signals outward to a wide range nearby cells throughout the body. Ascertained their role, then it is easily acceptable how peripheral neuropathies which damage small fibers lead to a bewildering array of symptoms: chronic wide-spread pain, dizziness, weakness, nausea and more [34]. Dorsal root ganglion (DRG) are clusters of sensory neurons, each projecting a single bifurcated axon toward both the dorsal spinal cord as well as peripheral targets (skin, muscle, viscera). In the periphery, axon terminals detect both noxious and innocuous thermal, chemical, and mechanical stimuli, and transduce these signals in the form of all-or-nothing action potentials toward the spinal cord. The type and threshold of stimulus necessary to initiate afferent signals, as well as the speed and fidelity with which these signals are processed, depends on the

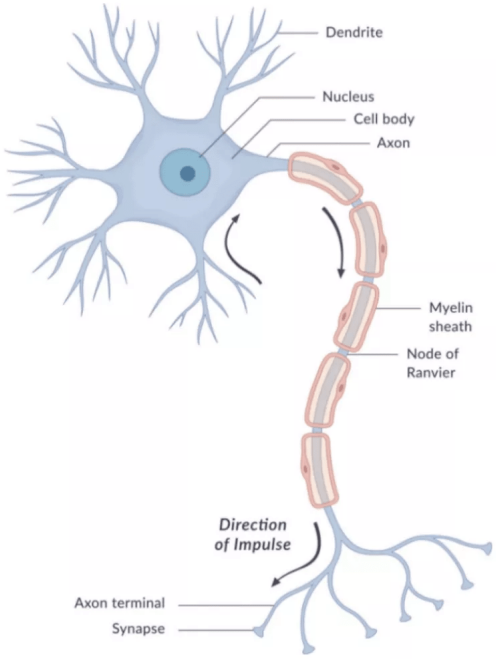

FIGURE 2.7: example of neuron, the basic unit of nervous system [48].

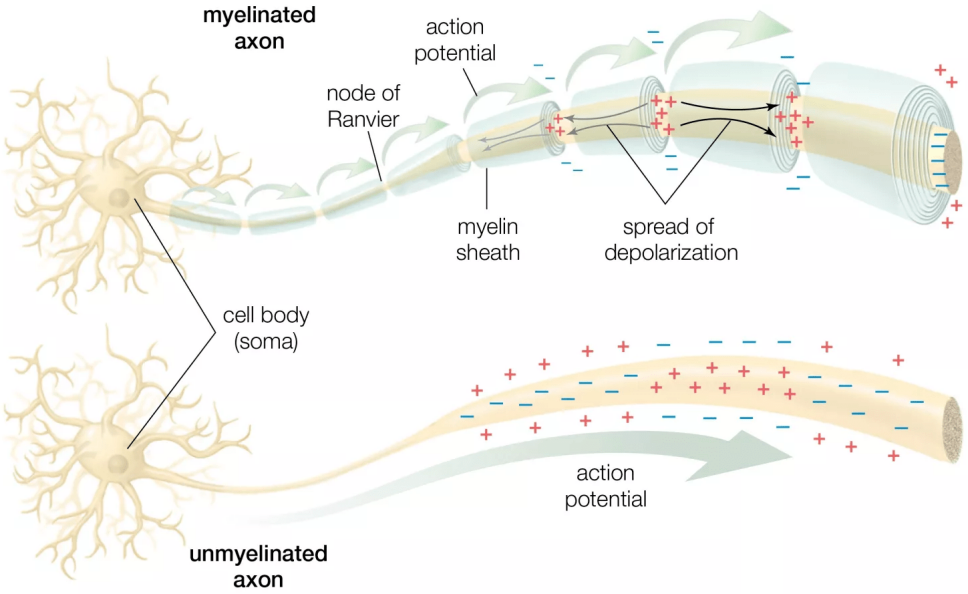

FIGURE 2.8: Myelinated and Unmyelinated axons [54].

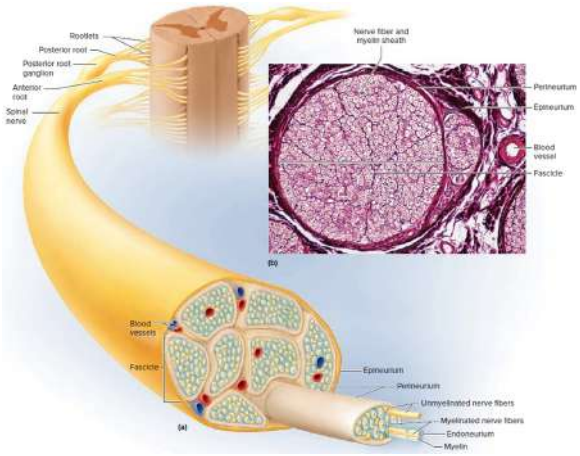

FIGURE 2.9: Anatomy of nerve fibers [4].

type of sensory neuron. Small-diameter neurons with unmyelinated axons have relatively high stimulus thresholds and are called nociceptors (or C-fibers). Small to medium-diameter neurons with thinly myelinated axons, have lower thresholds and higher conduction velocity are also referred to as nociceptors, but specifically as Aδ-fibers. The largest-diameter sensory neurons with relatively thick myelin are low threshold and called proprioceptors since they are largely responsible for afferent feedback from muscles for position, movement, and reflex. Under normal conditions, nociceptive DRG neurons exhibit relatively low levels of spontaneous spike activity. However, inflammation due to disease or injury may lead to increased intrinsic activity and increased sensitivity to external stimuli. This sensitization of peripheral nociceptors significantly contributes to the manifestation of chronic pain.[7]

2.4 Voltage Gated Sodium Channels

Peripheral nerve cells are capable of performing the functions described above through the presence of a special class of proteins at the level of their membrane. This class of proteins are **Voltage-gated sodium channels** (Na_V Channels, VGSCs), is a group of heteromeric integral transmembrane glycoproteins, the human genome contains ten structurally related sodium channel genes encoding for the alpha-subunits which share more than 50% amino acid sequence homology.

Their function, and the associated conformational change is precisely dictated by the potential variations between inside and outside of the membrane. When the cell membrane is depolarized by a few millivolts, above a threshold level, sodium channel allowed the influx of sodium ions before rapidly inactivating (see figure 2.10). Is this influx of sodium ions that generate the action potential indispensable for transduction and transmission of the road range of somatosensory signals (temperature, touch, smell, proprioception and pain) [25]. The ion-conducting aqueous pore is contained entirely within the α subunit that have a size of $\sim$260 kDa, and the essential elements

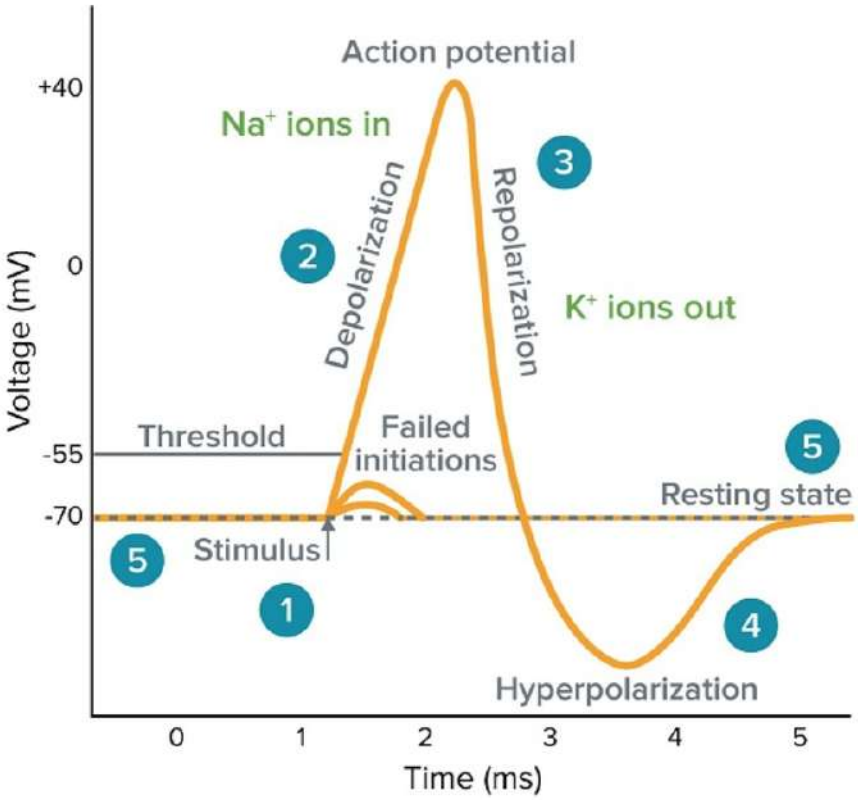

FIGURE 2.10: The Action Potential [2].

of sodium-channel function: channel opening, ion selectivity and rapid inactivation can be demonstrated when α subunits are expressed alone in heterologous cells. Coexpression of the β subunit is required for full reconstitution of the properties of native sodium channels, as these auxiliary subunits modify the kinetics and voltage-dependence of the gating [71]. The α subunits structure comprises four homologues domains I-IV (see figure 2.11), each domain having six transmembrane helices(S1-S6). The voltage sensor is located in the S4 segments, which contain positively charged amino-acid residues (mostly arginine) in every third position. A re-entrant loop between helices S5 and S6 is embedded into the transmembrane region of the channel to form the narrow, ion-selective filter at the extracellular end of the pore. Upon depo-

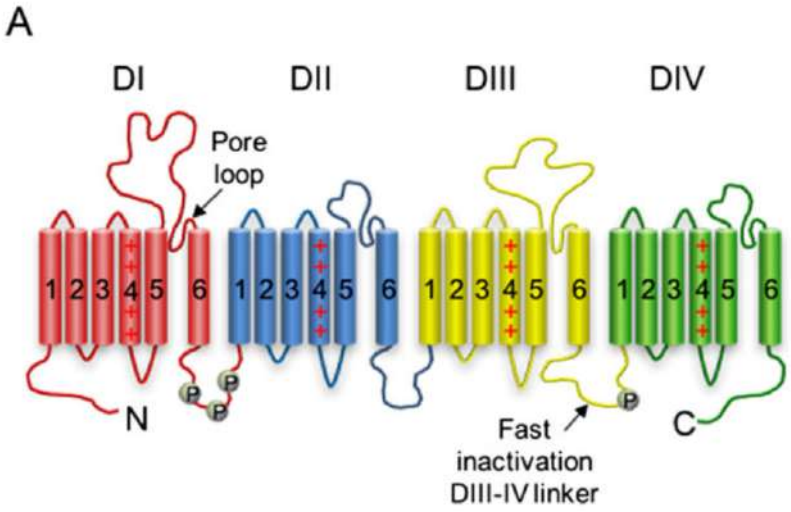

FIGURE 2.11: Single Chain [25].

larisation of a cell, the S4 regions in domains I–IV move rapidly and induce a conformational change in the protein which opens the ion channel pore. The entry of sodium ions through the pore leads to the upstroke of the action potential in excitable cells. Inactivation of the channel then follows as a highly conserved trio of amino acids located in the intracellular loop between domains III and IV moves into and occludes the channel pore, leading to the downstroke of the action potential [28].
In an evolutionary analysis, Voltage-dependent sodium channel genes have been identified in a variety of animals, including flies, leeches, squid and jellyfish, as well as mammalian and non-mammalian vertebrates. The biophysical properties, pharmacology, gene organization, and even intron splice sites of invertebrate sodium channels are largely similar to the mammalian sodium channels, adding further support to the idea that the primordial sodium channel was established before the evolutionary separation of the invertebrates from the vertebrates. This is an extremely important point thanks to

this strong correlation between the amino acid sequences and the physiological function performed, the homology modeling work is based on, and which will be discussed in more detail below. The human's Na_V family comprises nine homologous α subunits ($Na_V1.1$ - $Na_V1.9$), which overall molecular structures are highly conserved, but they are encoded by different genes. An evolutionary classification divide the genes into four groups, based on their subdivision between the chromosomes.

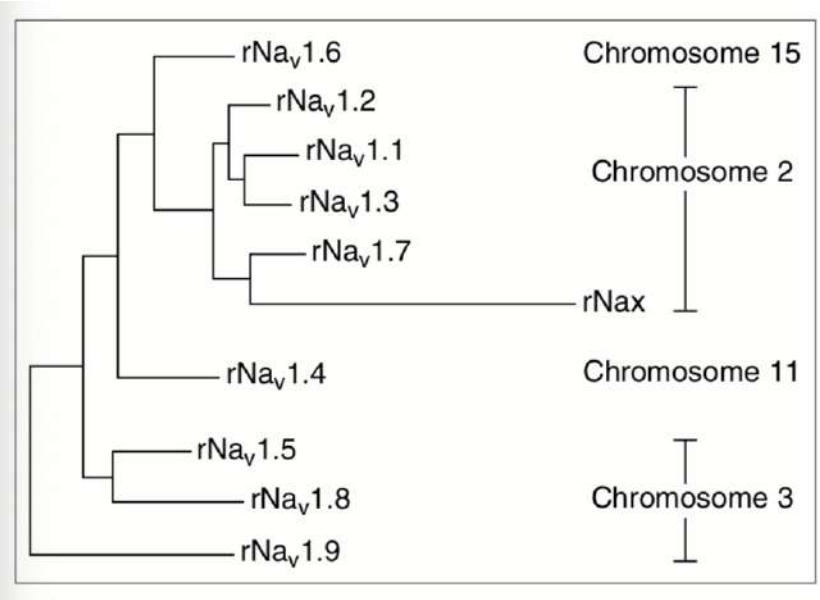

FIGURE 2.12: Phylogenetic tree [71].

A first group composed by genes encoding $Na_V1.1$, $Na_V1.2$, $Na_V1.3$ and $Na_V1.7$ are located in chromosome 2 both in human and in rodent, these group share high similarities in sequence, are broad express in neurons and have high sensibility at nanomolar concentration of the neurotoxin tetrodotoxin, that easily block the channel (see figure 2.12). A second cluster of genes encoding $Na_V1.5$, $Na_V1.8$ and $Na_V1.9$ is located on human chromosome 3, they share approximately 75% identical in amino-acid sequence to the group of channels on chromosome 2. This 25% in difference includes changes such as increased resistance to tetrodotoxin, as example, $Na_V1.5$ the principal cardiac isoform has a single amino-acid changed, from phenylalanine to cysteine, in the pore region of I domain and it is responsible for 200-fold reduction in tetrodotoxin sensitivity, relative to the channels encoded on chromosome 2.
At the corresponding position in channels $Na_V1.8$ and $Na_V1.9$ the residue is serine, and this change results in even greater resistance to tetrodotoxin. These two channels are preferentially expressed in peripheral sensory neurons. The last two isoforms $Na_V1.4$ and $Na_V1.6$ share greater than 85% sequence identity and similar functional properties to the chromosome 2 encoded channels, including tetrodotoxin sensitivity in the nanomolar concentration range, but they are encoded in others chromosomes. $Na_V1.4$ is located in chromosome 11 and $Na_V1.6$ in chromosome 15.

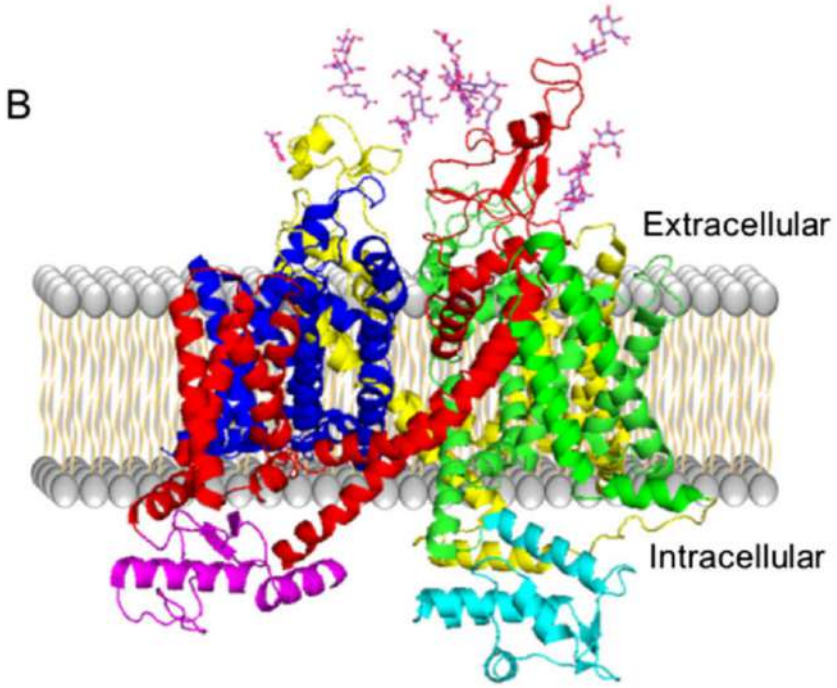

FIGURE 2.13: 3D structure of Sodium Channel Protein [25].

2.5 Neuropathic Pain Disorders

Pain disorders severely afflicts about half a billion people around the world, but medicine has no seen the remarkable progress in the treatment that other areas such as cardiovascular disease or cancer have undergone. Despite a substantial investments by the pharmaceutical industry, little progress has been made in developing novel efficacious and safe analgesics. One reason for this is that we know very little about the mechanisms that underlie different sort of pain.

Our limited knowledge of the types of sensory neurons and the subtypes of VGSCs involved in humans conditions, let alone the central mechanisms that modulate pain. We just know that VGSCs are strictly related to pain sensation, but the expression and role of Na_V subtypes in native sensory neurons are unclear. Moreover our way to develop new drugs are strongly based on animals models, the translational gap from rodents and humans has been blamed for the failure in developing pain medications. This gap becomes clear if we look at the distinct ratio of expression of different Na_V subtypes. The ratio of Na_V 1.7 expression in hDRG (human Dorsal Root Ganglion) is about 50% of the total VGSCs, and that value is much higher than in the mDRG (mouse DRG)(20%). By context, the mDRG has much higher ratio of Na_V 1.8 (45%) than in humans (15%), and a critical role of Na_V 1.8 in the pathogenesis of neuropathic pain has been well documented in rodents. It is reasonable to postulate that the high ratio of expression of Na_V 1.7 in humans DRG neurons is associated with an enhanced role of this subtype

in human pain condictions compared with the other subtypes. This notion is supported by human genetic showing that both most pain-related conditions (loss-of-function and gain-of-function) are found in SCN9A gene, while fewer mutations have been found in SCN10A (encoding Na_V 1.8), SCN11A (encoding Na_V 1.9) and SCN8A (encoding Na_V 1.6). Given this, blocking peripheral nerves as a route to treating many different type of pain is attractive. Nerve block has been used for decades as an effective treatment for most pain conditions and relies upon suppressing the electric signals carried out by VGSCs. [22][14]

Chapter 3

Computational Methods

3.1 Protein structure prediction

Owing to the significant improvement in genome sequencing technologies and efforts, the genomic sequences of a large number of organisms have been now determined. As of August 2015, 187 million sequences from 500000 organisms have been deposited in Genbank database. [6] Among them, 50 million sequences have been translated into protein amino acid sequences and stored in the UniProtKB/TrEMBL database. However sequences alone do not provide insight into what each protein does in living cells, and the three-dimensional structure of these is often important for interpreting their biological behavior. Structural biology techniques such as Nuclear magnetic resonance (NMR), X-ray crystallography and Cryo-EM (Electron Microscope) provide the most accurate characterization of the protein structure. However, because of the technical difficulties associated with cost and time, the gap between the number of protein sequences and that of protein structures is rapidly expanding. As of August 2015, there are only 100000 protein structures solved and listed in Protein Data Bank (PDB), compared with 50 million protein sequences in UniProt. Therefore, solved structures only account 0.2% of the known sequences. One promising approach to reduce this gap comes from computational chemistry modeling that can generate high-resolution structural models for the sequences that can be conveniently used by the science community.[33]

3.1.1 A brief introduction to modeling techniques

Traditionally, computational approaches for protein structure prediction have been categorized into three classes:

- Homology modelling
- Threading
- *Ab Initio* modelling

Homology modellig

Homology modeling is the most powerful method for predicting the tertiary structure of proteins in cases where a query protein has sequence similarity

to a protein with known atomic structure. These methods are based on the observation that structures are more conserved than sequences. Therefore, an accurate molecular model of a protein may be constructed by assigning a conformation that is based on sequence alignment, followed by model building and energy minimization. Reliable models could be obtain with a sequence similarity over 30%.

Threading

This method differs from homology modeling substantially because it tries to align the sequence with a structure, while in homology modeling there is an alignment between sequences.

Ab Initio

This technique is often used when there are no homologous structures with which to make a comparison. Not having a starting geometry, to compare with more demanding computational simulations capable of probing the space of the conformations in search of the coordinates that minimize potential energy. [40]

3.2 Swiss-Model

Homology modeling is become an important technique in structural biology, significantly contributing to narrowing the gap between known protein sequences and experimentally structures determined. This technique relies on evolutionary related structures (templates) to generate a structural model of protein of interest (target). With more experimentally determined structures of protein becoming available (in Protein Data Bank, PDB), it has been observed that interacting interfaces are often conserved among homologous protein. A significant contribution to this trend originates from the continuous progress of structure determination technologies, including recent developments of Electron Microscopy (EM) based methods, which are particularly suited for the large macromolecular complex. Swiss-Model is an online server able to generate a structural model in a completely automated way, through the homology modeling protocols [46]. Protein models were generated by following these main steps:

- **INPUT DATA**: Users can provide query sequence in FASTA [3] or write the amino acid sequence as a plain text. When the target protein is heteromeric, it's necessary to specify each subunit.
- **Template Search**: The sequence uploaded were used to search an evolutionary related protein structure by mean BLAST (Basic Logical Alignment Search Tool). This tool compare one or more query sequences to sequence database and calculates the statistical significance of matches, looking for the closest homologous protein. BLAST can be used to infer functional and evolutionary relationships between sequences as well as help identify members of gene families [44].

- **Template Selection**: Typically the research for a candidate produces more than a possible reference template, from which the most accurate candidate has to been chosen using a scoring function. Various aspects of the protein must be taken into account to evaluate the models, compatibility of amino acid residue with their local environment, knowledge-based statistical potential and evolutionary information as well as energy-based methods. Top-ranked templates are listed in a tabular form with a descriptive set of features.

- **Model Building**: For each template structures, a protein model is automatically produced by copy conserved atom coordinates as defined by target-template alignment. The insertion and deletion regions are modeled as if they were loops, maintaining an atomic-resolution of the residue. As final step, small structural distortions, unfavourable interactions or clashes introduced during the modelling process are resolved by energy minimizzation. Energy minimization are preformed using OpenMM library and CHARMM27 force field for parametrization.

- **Model Quality Estimation**: The last step followed by Swiss-Model involves the evaluation of the quality of the product model, relying on the QMEAN scoring function. QMEAN uses several statistical descriptor, three are based on energy potential based on geometry of model at different scales. Local geometry is evaluated by a torsion angle potential calculated over three consecutive amino acids, a secondary structure specific distance-dependent pairwise residue level potential is used to assess long-range interactions and a solvatation potential is implemented to describe the burrial status of residues. Finally two terms are added to describe the agreement of the terms just mentioned [69][50][49].

3.3 FG-MD: Energy minimization

Swiss-Model is a template based method, actually the most accurate method in protein structure forecast. As mentioned before, model are built by aligning the query sequences to a single protein template and then copying the structure information from the template to the aligned regions. Therefore, final structural models are often closer to the template than to their native structures. Although is efficient for removing steric clashes and unfavorable torsional angles, MD simulations without restraints often drive structure away from the native state. FG-MD use a multiple templates approach to reshape the energy landscape from golf-course-like to funnel-like ones and drive the energy minimization closer to native state; CASP 8 and CASP 9 (The Critical Assessment of protein Structure Prediction, [42]) refinement experiments have shown how FG-MD was among the very few methods that could consistently bring the initial model closer to the native state [30].

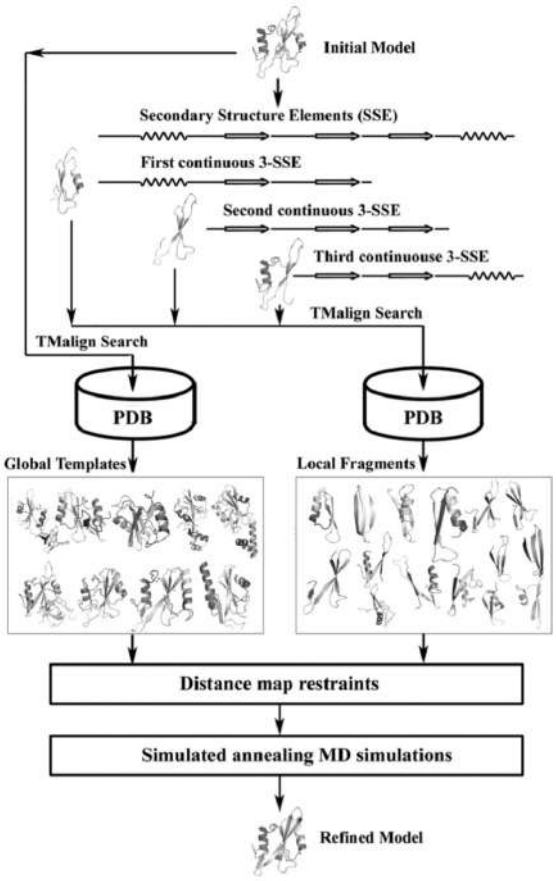

FIGURE 3.1: The protocol includes three stages of identification of fragment structures from the PDB, molecular dynamics refinement simulation guided by fragmental restraints, and final model selection [30].

3.3.1 FG-MD Refinement Protocol

The workflow followed by FG-MD are summed in figure 3.1. Minimization start from a target protein structures, the sequence was split into separated secondary structure elements (SSEs). The substructures consisting of three consecutive SSEs are used as probes to look for counterparts in the PDB library. The top 20 templates with the highest TM-score (A.2) were used to collect spatial restraints. The final refined models were selected on the basis of the sum of Z-score of hydrogen bonds, Z-score of the number of clashes and Z-score of FG-MD energy. All procedure is completely automated and FG-MD server is freely available at `http://zhanglab.ccmb.med.umich.edu/FG-MD`.
FG-MD force field is composed of four terms (the values of the distances shown below are dimensionless and normalized to 1 Å):

- **Distance Map Restraints:**
  The C_α distance maps were collected from three sources of initial models, global structure templates and fragmental structure templates, and it is written as:

$$E(r_{(ij)}) = \begin{cases} k_1(r_{(ij)} - r^1_{(ij)})^2 + k_2(r_{(ij)} - r^2_{(ij)})^2 + k_3(r_{(ij)} - r^3_{(ij)})^2 & \text{if } r_{(ij)} \leq 15 \\ 0 & \text{if } r_{(ij)} > 15 \end{cases}$$

  where $r_{(ij)}$ is the distance (in Å) between ith and jth C_α atoms, $r^1_{(ij)}$, $r^2_{(ij)}$, $r^3_{(ij)}$ instead are the distance maps from the initial model, global structure template and fragment template. k_1, k_2 and k_3 are the corresponding energy constants with values equal to 0.5, 0.5 and 2.0 $kcal/mole$ restrictions. The threshold value at 15Åis a heuristically determined value.

- **Explicit Hydrogen Binding:**
  H-bonding potential is defined as:

$$E_{HB}(d_{ij}, \alpha, \beta) = \begin{cases} k_4(d_{ij} - d_0)^2 + k_5(\alpha - \alpha_0)^2 + k_6(\beta - \beta_0)^2 & \text{if } d_{ij} \leq 3.0 \\ 0 & \text{if } d_{ij} > 3.0 \end{cases}$$

  Where d_{ij} is the distance (in Å) between hydrogen of the donor and oxygen of the acceptor, α is the angle of N-H-O and β is the angle of C-O-H (figure 3.2).

  The values of d_0, α_0 and β_0 are derived from the statistics average of high-resolution experimental structures dataset. Autors found that values were $d_0 = 1.95 \pm 0.17$ Å, $\alpha_0 = 160.0 \pm 12.2°$ and $\beta_0 = 150.0 \pm 17.5°$. k_4, k_5 and k_6 are the energy constants with values equal to 2.0, 0.5 and 0.5 respectively ($kcal/mole$).

- **Repulsive Potential:**
  This C_α repulsive potential was designed to quickly relax structural patterns with different clashes and is defined as follows:

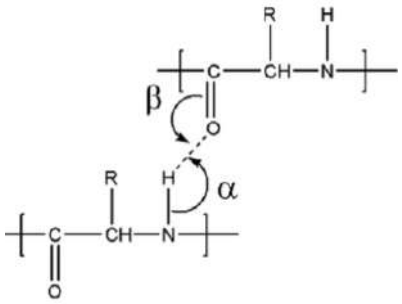

FIGURE 3.2: The definition of backbone hydrogen bond [30].

$$E(r_{(ij)}) = \begin{cases} k(3.6 - r_{(ij)}) & \text{if } r_{(ij)} \leq 3.6 \\ 0 & \text{if } r_{(ij)} > 3.6 \end{cases}$$

Where the energy constant $k = 200 Kcal/mole$.

- **AMBER99 Force Field:**
  Here is reported the standard AMBER99 force field [68]:
  split

$$E_{AMBER} = \sum_{bonds} K_r(r - r_{eq})^2 + \sum_{angles} K_\theta(\theta - \theta_{eq})^2 + \sum_{dihedrals} V_n/2[1 + \cos(n\phi - \gamma)] + \sum_{i<j}[A_{ij}/R_{ij}^2 - B_{ij}/R_{ij}^6 + q_i q_j/\varepsilon R_{ij}] \tag{3.1}$$

  where r, θ and ϕ are bond length, bond angle and torsion angle respectively; r_{eq}, θ_{eq} and γ are the respectively equilibrium values. K_r, K_θ and V_n are the force constants (for bond length, bond angle and torsion angle). A_{ij}, B_{ij} are Lennard-Jones parameters, q_i and q_j are the partial charge of atom i and j. R_{ij} is the distance between atom pair i and j.

3.4 QMEANBrane

Once the protein models are generated, a fundamental task is to evaluate their goodness. Being able to discriminate the quality of a model allows to choose the best candidate among a series of alternatives. In this sense, various techniques have been developed to address this question. However, in the case of membrane proteins, the use of these methods clashes with their chemical-physical properties that favorable interactions are opposite to the case of soluble proteins. And these knowledge-based methods have been calibrated on soluble proteins, thus the perform poorly when applied to membrane proteins. QMEANBrane exploits the increasing availability of deposited high definition membrane protein structures to adapt knowledge-based methods to this class of proteins. It is known that the properties of

membrane proteins are strongly influenced by their interaction with phospholipid tails, but a clear division into a membrane region and a soluble region does not adequately reflect the variation in molecular properties along the membrane axis. To capture these differences, QMEANBrane divides the study into three parts: an interface zone consisting of all those residues whose C_α are at a distance of 5Åfrom the defined membrane plane (see figure 3.3). A membrane region enclosed by all those residues that are more than 5Åbetween the two planes, finally, a region of soluble protein consisting of the remaining amino acids [26].

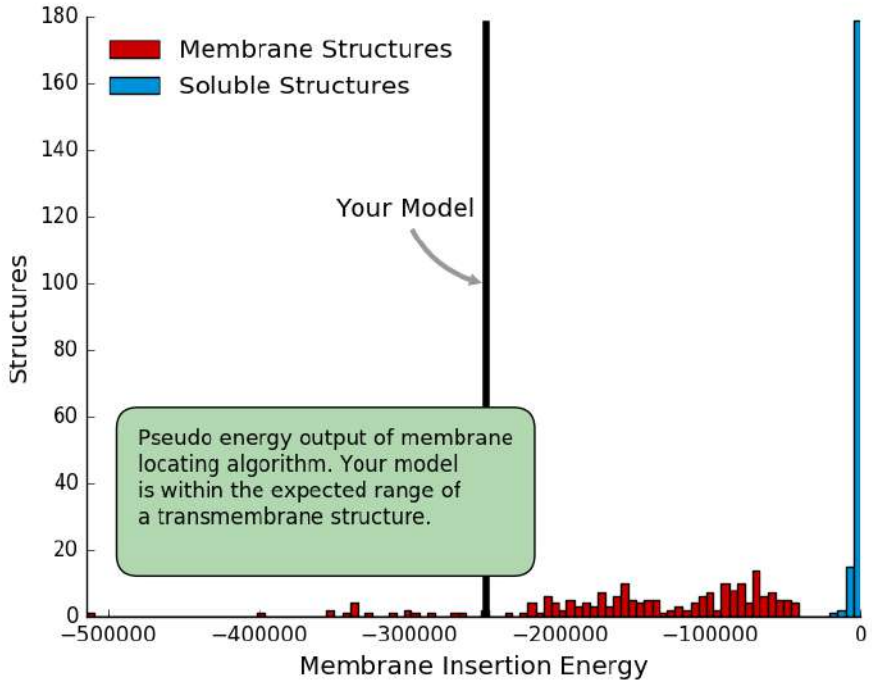

FIGURE 3.3: Example of result obtain using QMEANBrane, relative to WT (6A90 template) after FG-MD minimization.

3.5 RING2.0

RING2.0 is the tool that has been implemented to represent the models generated in the form of graphs. What is a graph? It is nothing more than a mathematical representation of a system, through which the nodes identify elements or characteristics of the system, and the connections between two elements or more elements are highlighted through the arcs that connect them.

3.5.1 definition of graph

A graph is formally defined as *G=(V,E)*, where V={1,...n} is a finite set of nodes (or vertices) and *E: V x V* is the set of edges which connect the nodes. With

the name n it means the total number of nodes, while m identifies the total number of arcs (see figure 3.4).

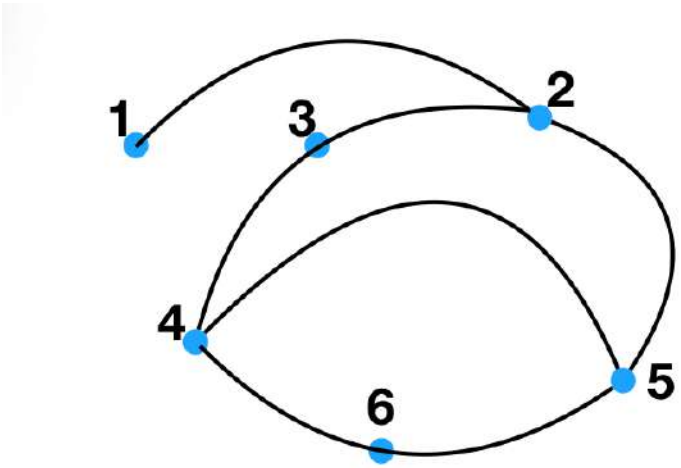

FIGURE 3.4: Example of graph.

These interactions, in addition to a graphic representation, can also be reported in matrix form: The adjacency matrix A of graph G is defined as:

$$[A_{ij}] = \begin{cases} 1 & \text{if } (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

where v_i and v_j are nodes of G. Moreover, a sequence of nodes v_i, v_j, $..v_d$ of length *k-1* is defined as path w such that $(v_{i-1}, v_i) \in E$ for $1 < i \leq k.wisapathif v_i \neq v_j$ iff $i \neq j \forall i,j \in 1, ...k$.

RING-2.0 algorithm generates the graph in two steps. The first identifies a list of residue-residue pairs eligible to undergo an interaction based merely on distance measurements. The second characterizes every contact by identifying the specific type of interaction. The first step identifies interacting pairs with different strategies according to the user choice, Network Policy parameter in the web server interface, and reflecting different cases.

- **Closest:** all atoms of the residue pair are considered to measure the distance. This option is convenient for PDBs with good resolution for which is safe to consider sidechain coordinates.
- **Lollipop:** the distance is calculated between the mass centers of the two interacting residues. Moreover the algorithm checks that the sidechains are not pointing in opposite direction.
- C_α**:** the distance is calculated between C-alpha atoms.
- C_β**:** the distance is calculated between C-beta atoms.

This step produces a list of all retrieved interactions, labeling them as generic Inter-Atomic Contact (IAC). Now an algorithm identify which kind

of interaction exists between pairs and allow the users to decide the cardinality by means of the Interaction Type parameter (see figure 3.5):

- **One:** RING reports only one interaction per residue pair (the most energetic).
- **Multiple:** RING reports multiple interactions per residue pair but only one interaction per interaction type.
- **All:** Were listed all the interactions.
- **No specific:** This second step is skipped and were provided only generic IAC interactions.

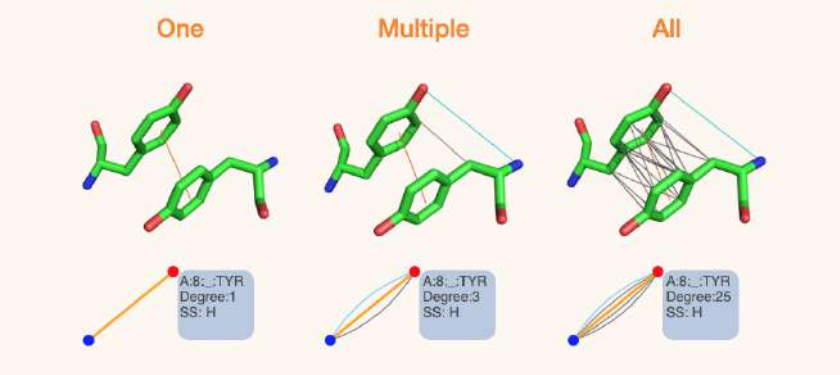

FIGURE 3.5: Visual representation of how edges were built depending on the chosen Interaction Type parameter [52]

3.5.2 Definiction of Interaction Type

RING2.0 is a freely available tool able to identifies 6 different type of interaction, plus a generic interaction (IAC) that simply indicates a generic contact based on a distance cutoff.

- **Hydrogen bond (HBOND):** they are determined on the basis of the DSSP method (A.4). The H-bond between atom pair exist if it respects these two rules: Distance between Donor (D) and Acceptor (A) is less or equal to 3.5 (or 5.5 relaxed) Å. The angle H-D-A (θ) is less or equal to 63° (figure 3.6).
- **Van der Waals (VDW):** are identified by simply measuring the distance between the surface of two atoms. The distance threshold is 0.5 (or 0.8 relaxed) Å. The pairs of atoms considered valid in establishing this type of bond are C-C and C-S, plus the special case of oxygen and nitrogen in the residues of Glutamine and Aspargine (figure 3.7).
- **Disulfide bridges:** This is a covalenta kind bond, and is subject to strong spatial constraints. RING-2.0 identifies disulfide bridges when the distance between sulfur atoms of cysteine pairs is lower or equal to 2.5 (or 3.0 relaxed) Å(figure 3.8).

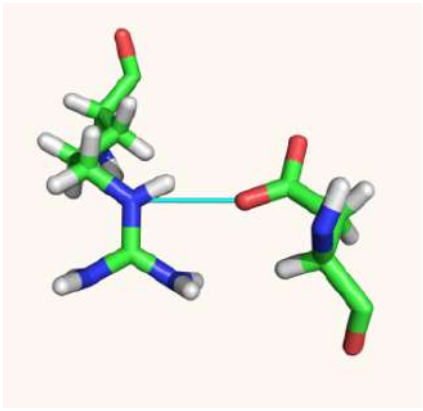

FIGURE 3.6: Representation of the hydrogen bond [52].

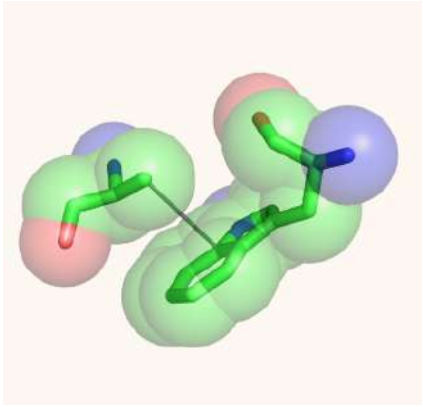

FIGURE 3.7: Representation of Van der Waals bond [52].

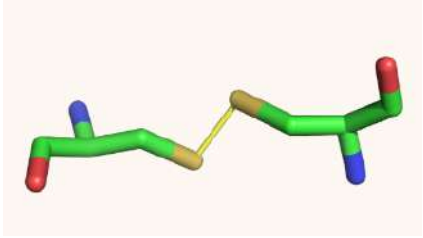

FIGURE 3.8: Representation of disulfide bond [52].

- **Ionic bridges:** occurs when the center of mass of the charged groups are at an equal or lower distance 4.0 (or 5.0 relaxed) Å(figure 3.9).

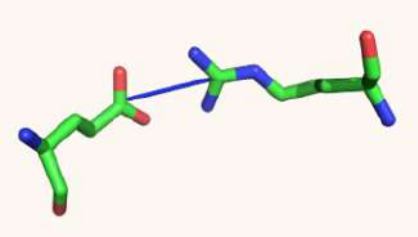

FIGURE 3.9: Representation of disulfide bond [52].

- $\pi - \pi$ **Stacking (PIPISTACK):** this interaction is evalueted only between aromatic residues (His, Tyr, Trp, Phe). RING2.0 takes one ring plane as reference and calculates the strength of the force based on the angle between the planes of the two rings (figures 3.10 and 3.11).

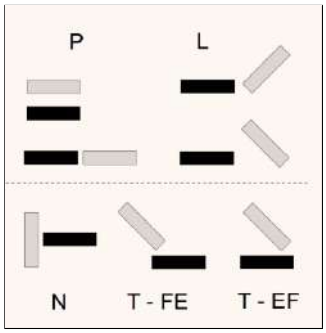

FIGURE 3.10: Possible angles between the rings.

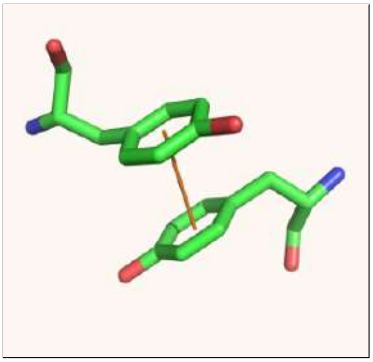

FIGURE 3.11: Image of $\pi - \pi$ stacking.

- π**-cation (PICATION):** this interaction occurs between positively charged residues (Arg, Lys) and the electronic clouds of aromatic rings. In this sense, Histidine is not taken in account because it can act both as cation and as π-system. The conditions for which RING2.0 considers the establishment of the bond are two:the distance between the center of mass of the positively charged residue and any atom of the π-system must be less than 5.0 (or 7.0 relaxed) Åand the angle between the distance vector and the ring plane has to guarantee that the mass center of the cation lies above (or below) the ring area (figure 3.12).

3.6 Cytoscape

Cytoscape is an open source software project for integrating biomolecular interaction networks with high-throughput expression data and other molecular states into a unified conceptual framework. The central organization on which Cytoscape rests are the graphs, in perfect continuity with what is

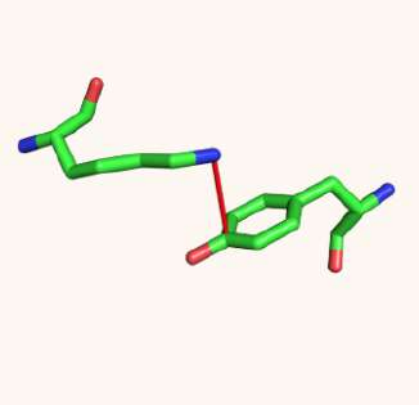

FIGURE 3.12: Representation of π-cation bond [52].

produced through the use of RING2.0. This tool responds to the growing demand for a method of analysis due to the vastness of models that have been developed to simulate biochemical reactions and gene transcription kinetics, cellular physiology, and metabolic control. Such models promise to transform biological research by providing a framework to systematically interrogate and experimentally verify knowledge of a pathway, manage the immense complexity of hundreds or potentially thousands of cellular components and interactions (how to represent the dense connections between protein residues that give life to a biologically active structure). Furthermore reveal emergent properties and unanticipated consequences of different pathway configurations (how to identify which biomolecule can represent a good candidate in the development of increasingly selective drugs). The core of Cytoscape is extensible through a straightforward plug-in architecture, allowing rapid development of additional computational analyses and features [60]. To facilitate the analysis, two plug-ins were used to carry out the analysis on the network metrics.

- **RINalyzer** One of the main features of RINalyzer is the computation and illustration of a comprehensive set of well-known topological network centrality measures (based on shortest paths, current flows, or random walks) for relating spatially distant residue nodes, and discovering crucial residues and their long-range interaction paths in protein structures. In particular, this tool was used to determine the metrics based on a shortest path approach [19].

- **CytoNCA** CytoNCA provides multiple centrality calculations (Betweenness and Closeness Centrality, Degree Centrality, Eigenvector Centrality, Local Average Connectivity-based Centrality, Network Centrality, Subgraph Centrality, Information Centrality) for both weighted and unweighted networks, gives various forms of visualization analysis, and quantitatively evaluates the computation results [65].

It was decided to use both tools to verify the consistency of the analytical results.

3.7 Pattern recognition through Kernel Methods

The search for recurring patterns between data sets is as old as science, and in the same way the automated search for these schemes is as old as computing. Pattern analysis deals with the problem of detecting relations in data, starting from the assumed that the data mean the output of any kind of observation, measurement or recording apparatus. This therefore includes images in digital format, sequences of DNA, pieces of text, time series, records of commercial transactions or, like in our case, vectors describing the state of a physical system. Once in possession of this data set, the ways through which valuable knowledge can be obtained must be explored. Referring to knowledge as something more abstract, at the level of relations between and patterns within the data. Such knowledge can enable us to make predictions about the source of the data or draw inferences about the relationships inherent in the data. At the beginning, the automated approaches to this type of problem have had to come up against various practical obstacles, that only in the recent period have they been protagonists of two important turns: In the mid-80s the field of pattern analysis underwent a nonlinear revolution allowing, as the name suggests, to address the study of systems characterized by non-linear patterns; albeit with heuristic algorithms and incomplete statistical analysis. Finally, in the mid-1990s, the approach to kernel-based methods made it possible to analyze non-linear relationships with the accuracy that was previously reserved only for linear patterns. Through patterns, we can understand any relations, regularities or structure inherent in some source of data. By detecting significant patterns in the available data, a system can anticipate predictions about new data coming from the same source. There are many important problems that can only be solved using this approach, problems ranging from bioinformatics to text categorization, from image analysis to web retrieval [61]. In our case, the models were represented through graphs, from which we then tried to extrapolate some congruent patterns with their biological behavior. To do this, two kernels methods were used:

- **Shortest-path kernels** From a completely general point of view, the graph-kernels are based on the comparison of substructures of graphs, which can be: paths, subtrees and cyclic patterns. The first essential step in a shortest-path kernel appraoch is to transform the original graphs into shortest-paths graphs ($G \rightarrow S$). S is a graph composed of the same set of nodes of G, but unlike G has an arc connecting each pair of nodes of G connected by a path (the weight of this arc is proportional to the shortest path connecting the two nodes).

  - **Definition of Shortest Path Graph Kernel** *Taken two graphs $G_1$and G_2 transformed into their correspondents S_1 and S_2. It is possible to define the shortest path kernel between $S_1 = (V_1, E_1)$ and $S_2 = (V_2, E_2)$ as*:

$$k_{ShortestPath}(S_1, S_2) = \sum_{e_1 \in E_1} \sum_{e_2 \in E_2} k_w^{(l)}(e_1, e_2) \tag{3.3}$$

  where $k_w^{(l)}$ is a positive define kernel on edge walks of length 1 [9].

- **Weisfeiler-Lehman kernel** Weisfeiler-Lehman approach follows another way to probe if two graphs share common patterns. As with the shortest path, even with this approach, the graphs undergo a transformation before a kernel is applied to them. Weisfeiler-Lehman takes a graph and maps it to a sequence of graphs, whose nodes are defined in such a way as to capture and label the local topology of the starting graph. With this method it is necessary to introduce some tricks to the previously described definition of graph: A graph G is defined by the triplet (V, E, l), where V, E are the vertices and edges of graph and l: $V \rightarrow \Sigma$ is a function the assigns an alphabet labels ($\in \Sigma$) to nodes in the graph. This method provides for probing the topology by studying the surroundings of the nodes, through the neighboring vertices. A neighborhood of a node $N(v)$ is defined as the set of nodes to which it is directly connected by an arc ($N(v) = \{v' \mid (v, v') \in E\}$). The concept of subtree must therefore be introduced: a subtree is a subgraph of graph, which has no cycles (a path of length k-1 in which $v_1 = v_k$) and it is rooted by designed a node as a root of the subtree. Then the height of the subtree is defined as the maximum distance between the root and any other node belonging to the subtree (see figures 3.13 and 3.14) [63].

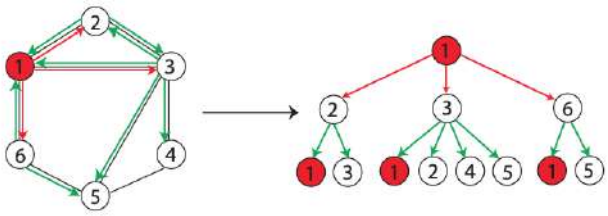

FIGURE 3.13: Subtree pattern of a graph with height 2 and rooted in node 1 [63].

The idea behind this algorithm is to increase the labels of each node, based on the local topology, this process is repeated until G and G' differ. If the same new labels are kept, the two graphs are isomorphic.

– **Definition of Weisfeiler-Lehman Kernel** *It is called a Weisfeiler-Lehman sequence of graphs of height h as:*

$$\{G_0, G_1, ..G_i, ..G_h\} = \{(V_0, E_0, l_0), (V_1, E_1, l_1), ..(V_i, E_i, l_i), ..(V_h, E_h, l_h)\} \tag{3.4}$$

where G_0 corresponds to the original graph, G_i identifies the generic contract graph after i itterations and h is the total number of itterations improved.
We then define the kernel between two graphs G and G' as:

$$k_{WL}^{(h)} = k(G_0, G_0') + k(G_1, G_1') + .. + k(G_h, G_h') \tag{3.5}$$

where $G_0, ..G_h$ and $G'_0, ..G'_h$ are the graphs sequences of G and G′ respectively.

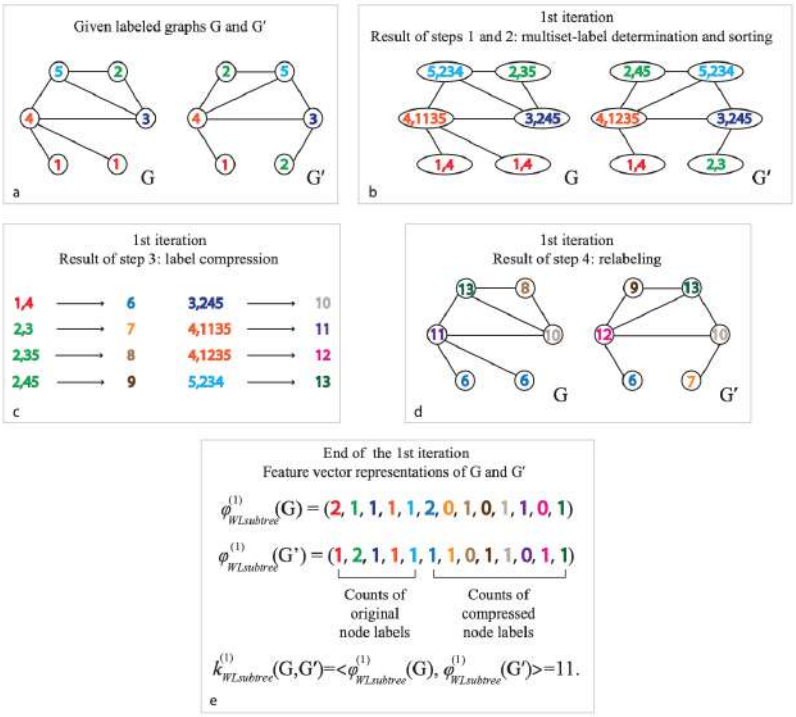

FIGURE 3.14: Rappresentation of the Weisfeiler-Lehman subtree kernel with one itteration (h=1) for two graphs. In this case the graphs are not isomorphic, they are already different after the first itteration. [63].

3.7.1 Dominant-set Clustering

Dominant-set clustering framework is an approach that tries to answer at the problem of organizing a set of elements in clusters, in such a way that each group satisfies an internal homogeneity and expresses an external inhomogeneity between the groups. The algorithm behind this approach does not require any assumption underlying the data representation (cluster spectra: does not require that the elements to be clustered have a representation as points of a vector space), it also does not require prior knowledge of the number of clusters (as it is able to determine them in sequence). Finally it also allows to determine if there are overlaps between the clusters [10].

The schedule followed to produce this thesis is briefly summarized in the pipeline below (see figure 3.15):

- The isoform 1 classified with the code **NP_002968.1** has been selected as Wild Type (WT) sequence, in agreement with **NCBI** (National Center For Biotechnology Information).

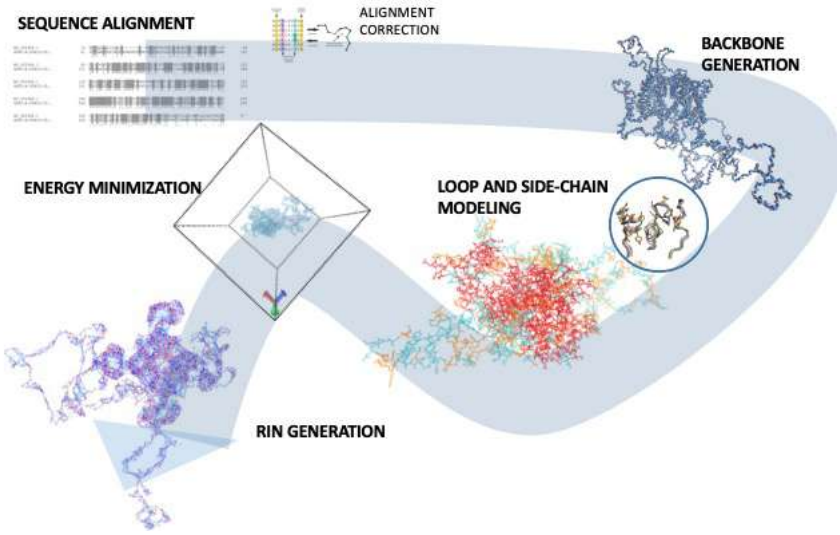

FIGURE 3.15: Rappresentation of pipeline followed.

- it has been prepared an excel file, and taking in account the mutations reported by Dimos et al.([32]); they have been created manually 85 copies of WT sequence with a singolar puntual mutation.
- Homology modelling of each sequences has been produced using **Swiss-Model** Server (https://swissmodel.expasy.org/). using three different templates:
  - A first template produced by Dimos et al, based on the cryo-SEM structure of ortholog protein of *Acrobacter bultzeri*.
  - A second template, from eukaryotic kingdom, it has been chosen for its higher closeness to human.
  - A last template, evolutionarily distant from humans, but interesting because it is representative of the sodium channel in the open state.
- Subsequently, energy minimization of each structures were preformed using online tool **FG-MD** (Fragment-Guided MD simulation).
- Minimized structures obtained in this way, were upload to **RING2.0** server to convert them in their corresponding RIN (Residue Interaction Network).
- RINs taken as representative of the structures were analyzed using **Cytoscape** and kernel-method approaches.

Chapter 4

Related Works

The aim of this thesis was to test and improve the computational pipeline proposed by Dimos et al ([32]), their work is reported below. Other publications have been considered and a work that has better explored the potential of computational approaches is shown at the end of this chapter.

4.1 Reference Work

This research group is based at IRCCS Foundation "Carlo Besta" Neurological Institute in Milan. Among the research areas followed by the institute of neurology, there are clearly the study of the onset of forms of painful neuropathies linked to gene expression. In particular, mutations in the SCN9A gene, which codes for the α-subunit of the voltage-gated sodium channel NaV1.7, have been speculated to be related with these pathological forms. Dimos et al used homology modeling to build an atomic model of NaV1.7 and a network-based theoretical approach, which can predict interatomic interactions and connectivity arrangements, to investigate how pain-related NaV1.7 mutations may alter specific interatomic bonds and cause connectivity rearrangement, compared to benign variants and polymorphisms. The analysis of networks, and in particular, of how mutations change the topology of these networks have been calculated by determining the following metrics: betweenness centrality (B_{ct}), degree (D), clustering coefficient (CC_{ct}), closeness (C_{ct}) and eccentricity (E_{ct}) where calculated for each graph. Finally, the determined values were compared by taking the same value relative to the WT model as a benchmark ($\Delta_{value} = mutant_{value} - WT_{value}$).

4.1.1 Methods followed

The pipeline followed is briefly summarized in the figure 4.1, essentially the work was divided into two main blocks. In a first part, representative protein models of both the WT sequence and all the considered mutations were generated. In the second block, the structures have been converted into graphs showing the topology of the structures, representing as graph nodes the amino acids, and the arcs as the interactions existing between the residues.

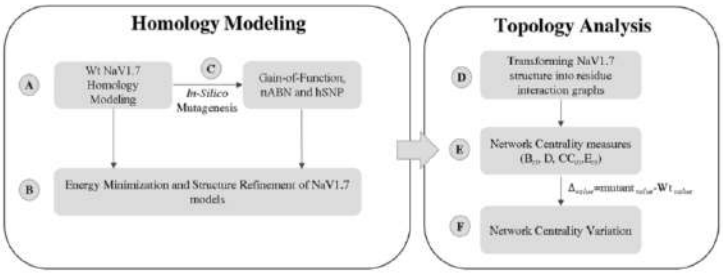

FIGURE 4.1: NaV1.7 computational protocol overview. A NaV1.7 WT homology modelling of based on the bacterial NavAb sodium channel template. B Energy minimization and structure refinement of the protein structure with YAMBER force field and FG-MD server. C *In-silico* mutagenesis for pathogenetic and control group (nABN/hSNPs) mutations. D Transforming NaV1.7 structure into residue interaction graphs. The construction of inter-residue network was based on interatomic bonds (hydrophobic, hydrogen bonds, salt-bridges, cation-π and π-π stacking interactions) using the commands "ListIntAtom" and "ListIntBo" via YASARA software. The *de novo* network construction for each mutant and WT models is achieved considering the predicted binary interatomic bonds. E-F. Network centrality calculation and their relative variation between mutant and WT ($\Delta_{value} = mutant_{value} - WT_{value}$) [32]

- **Homology Modeling of NaV1.7 WT sequence** The representative model of the closed conformational state was generated starting from the *Arcobacter bultzeri* (NaVAb) template and using NM_002977.3 as a human sequence, through the MEMOIR server [20]. Gap region (269-340, DI) between template-target alignment and interdomain loop regions (416-726, DI-DII; 967- 1175, DII-DIII; 1458-1498, DIII-DIV) were excluded from *in-silico* mutagenesis. The template used and the WT sequence share an identity of 28% respectively for DI, DIII and DIV while it is 24% for DII (overall 27% sequence identity). *Ab-initio* analysis was then applied to extend a missing trait of the PD propeller S6, using the tool I-TASSER online. The structure thus generated was subjected to an energy minimization, using first YAMBER force field (from YASARA software [36]) and then with FG-MD.

- ***In-Silico* Mutagenesis of NaV1.7 pathogenic and control mutations** Subsequently, mutations found in IEM, PEPD and SFN patients have been introduced in silico. Of these mutations the gain-of-function character has been demonstrated through electrophysiology assay. To increase the number of control variants, the missense SNPs identified among the SCN9A homologous genes that share the identity of the 90% nucleotide sequence using the NCBI HomoloGene database were added. finally, to converge to models closer to the native state of each structure, each mutated-model was subjected to the same energy minimizations applied for the WT.

- **Graph representation of NaV1.7 structures** NaV1.7 models were converted in their mathamtical graphs by indentifying interatomic bonds between the residues. The nature of interatomic interactions considered were: hydrophobic iterations, hydrogen bonds, saline-bridges, π-cation and π-π stacking. And an edge was placed to represent them if they came to be established between two amino acids *i* and *j* at a maximum distance of 5 Å.

- **Calculation of network metrics** The calculation of the metrics was performed by viewing the networks with Cytoscape and using the NetworkAnalyzer plugin. Below is a brief description of the metrics considered, defining G=(N, E) a graph where N is a set of nodes and E is a set of the edges:

  - **Betweenness Centrality (B_{ct})**: B_{ct} (of a generic n node) is defined as the fraction of shortest pathways between all pair of nodes of network that go through that n node.

$$B_{ct}(n) = \sum_{c \neq n \neq t} \frac{\sigma_{ct}(n)}{\sigma_{ct}} \tag{4.1}$$

    where s, t $\in$ N, $\sigma_{ct}(n)$ is the total number of shortest paths from s to t that n lies on, while σ_{ct} indicates the number of shortest paths between s and t. This metric can highlight the importance of a node, showing how this node acts as a bridge between the various topological connections of the network.
  - **Edge Betweenness (EB_{ct})**: Similarly EB_{ct} shad light on the importance of an edge, quantity that is expressed by the equation:

$$EB_{ct}(e) = \sum_{n_i \in N} \sum_{n_j \in N \setminus (n_i)} \frac{\sigma_{n_i n_j}(e)}{\sigma_{n_i n_j}} \tag{4.2}$$

    where $\sigma_{n_i n_j}$ indicates the total number of shortest paths between n_i and n_j, while $\sigma_{n_i n_j}(e)$ indicates the number of shortest paths between n_i and n_j which pass through e $\in$ E.
  - **Degree (D)**: D(n) simply identifies the number of nodes directly connected to n.
  - **Clustering Coefficient (CC_{ct})**: CC_{ct} identifies if there are well connected nodes, in detail the fraction of triangles around a node between the total number of possible triangles. Is define by:

$$CC_{ct}(n) = \frac{2e_n}{k_n(K_n - 1)} \tag{4.3}$$

    where k_n is the number of neighbors of n and e_n is the number of connected pairs between all neighbors of n.
  - **Closeness Centrality (C_{ct})**: C_{ct} it measure the sum of inverted distances (how far), to all other nodes in the graph. In view of the

lower distance between a node and the rest of the network, the more this node is important. It is define as:

$$C_{ct}(n) = \frac{1}{average(d(n,m))} \tag{4.4}$$

– **Eccentricity (E_{ct})**: E_{ct} measures the distances between a node n and the most distance node m, if $E_{ct}(n)$ is low means the n close to all the other nodes.

$$E_{ct}(n) = max[d(n,m)] \tag{4.5}$$

4.1.2 Results obtained

In assembling the structure of WT NaV1.7 in the closed state, a homology-modeling approach was applied. Due to the presence of some gaps between the reference template and the human sequence, some areas have been excluded from the final model, while an important missing treatise in the alignment (belonging to the helix S6 of the Pore Domain), has been modeled through an *ab initio* approach. The model obtained was analyzed with RAMPAGE server ([39]), resulting in 88.5% of residues in most favored region, 9% in allowed region (90 residues) and 2.5% in outlier region (25 residues). The work previously carried out shows how the gain-of-function mutations significantly alter the bio-physical properties of the canal, without however highlighting what are the variations between the underlying interatomic bonds. To do this the authors have resorted to a network-based approach, highlighting how in particular the betweenness centrality is able to discriminate between pain-related mutations from non-pathogenic mutations and isomorphisms.

This work shows how $\Delta B_{ct}(B_{ct(mutant)} - B_{ct(WT)})$ values tend to be significantly higher in NaV1.7 pain-related mutations than in control groups, as B_{ct}represents the influence that the shortest communication pathways have on the overall interatomic connections. Nodes with high B_{ct} value could efficiently integrate signals (e.g. energy) and the reduction of B_{ct} value caused by single amino acid substitutions suggests that the signaling transfer capability of the network is decreased. Conversely, the increase of B_{ct} value suggests that a mutated node could facilitate the load transfer through the shortest communication pathways. Therefore, changes in ΔB_{ct} reflect increased or decreased potential for connectivity of amino acid within the protein and provides numerical values about how single amino acid substitutions might act as a bottleneck for specific nodes linking different parts of the network. Furthermore, previous studies have shown that there may be an allosteric type of communication through residues having high values of B_{ct}. More in detail, the authors of this paper, using $\Delta B_{ct} \pm 0.26$ as cut-off value, were able to cluster 43 of the 53 control mutations and 23 of 30 gain-of-function mutations with a specificity of 83% (figure 4.2).

Disease	Mutation	Amino acid Properties	Channel Part	ΔB_{ct}
IEM	I136V	$=H_{\varphi}O$	VSD ($S1;D_I$)	0.12
	S211P	Polar → $H_{\varphi}O$	VSD ($S4;D_I$)	-1.09
	F216S	$H_{\varphi}O$ → polar	VSD ($S4;D_I$)	-1.71
	L823R	$H_{\varphi}O$ → charged	VSD ($S4;D_{II}$)	1.23
	W1538R	$H_{\varphi}O$ → charged	VSD ($S2\text{-}S3;D_{IV}$)	0.18
	I234T	$H_{\varphi}O$ → polar	S4-S5 (D_I)	2.33
	S241T	=Polar	S4-S5 (D_I)	0.34
	I848T	$H_{\varphi}O$ → polar	S4-S5 (D_{II})	-5.83
	L858H	$H_{\varphi}O$ → charged	S4-S5 (D_{II})	-1.85
	L858F	$=H_{\varphi}O$	S4-S5 (D_{II})	-1.74
	G856D[a]	$H_{\varphi}O$ → charged	S4-S5 (D_{II})	-0.55
	A863P	$=H_{\varphi}O$	S4-S5 (D_{II})	-0.32
	P1308L	$=H_{\varphi}O$	S4-S5 (D_{III})	0.04
	V1316A	$=H_{\varphi}O$	S4-S5 (DI_{III})	0.36
	A1632E[b]	$H_{\varphi}O$ → charged	S4-S5 (D_{IV})	0.27
	N395K	polar → charged	Pore ($S6;D_I$)	5.32
	V400M	$=H_{\varphi}O$	Pore ($S6;D_I$)	-0.68
	V872G	$=H_{\varphi}O$	Pore ($S5;D_I$)	-2.48
	F1449V	$=H_{\varphi}O$	Pore ($S6;D_{III}$)	-0.51
	A1746G	$=H_{\varphi}O$	Pore ($S6;D_{IV}$)	1.40
SFN	R185H	=charged	VSD ($S2\text{-}S3;D_I$)	0
	I228M	$=H_{\varphi}O$	VSD ($S4;D_I$)	2.04
	I739V	$=H_{\varphi}O$	VSD ($S1;D_{II}$)	0.54
	M1532I	$=H_{\varphi}O$	VSD ($S2\text{-}S3;D_{IV}$)	0.15
	M932L	$=H_{\varphi}O$	Loop Pore (D_{II})	0.46
PEPD	V1298D	$H_{\varphi}O$ → charged	S4-S5 (D_{III})	-0.81
	V1298F	$=H_{\varphi}O$	S4-S5 (D_{III})	-0.004
	V1299F	$=H_{\varphi}O$	S4-S5 (D_{III})	0.07
	G1607R	$H_{\varphi}O$ → charged	S4-S5 (D_{III})	0.62
	M1627K	$H_{\varphi}O$ → charged	S4-S5 (D_{IV})	1.22

FIGURE 4.2: Representation of ΔB_{ct} NaV1.7 mutations relative to IEM, PEPD and SFN.Reporting the physical-chemical changes resulting from the mutation and their location along the peptide chain.

4.1.3 Conclusion

The authors conclude by stating that gain-of-function mutations significantly alternate the connections between the channel residues and by proposing to consider B_{ct} as a marker of pathogenic shift in the mutant channels. In any case experimental evaluations will be necessary to clarify their biological meaning [32].

4.2 Other work

In strict correlation with what has been done, documentation of a work concerning the voltage-gated calcium channels (VGCCs) was presented. As voltage-gated sodium channels (VGSCs) also VGCCs are essential for maintaining the normal physiological functions of the human body, are still embedded in plasmatic membrane and they replay up to depolarization signals changing their conformation to open state and selectively allowing the influx of calcium ions. Moreover VGCCs and VGSCs also share structural similarities, in both cases the essential subunit consists of the rewinding of a single polypeptide chain, which is organized into 4 domains (each domain has 6 transmembrane helices of which the first 4, S1-S4, form the sensitive voltage domain and the last two constitute the pore domain S5-S6) ([24]). In mammals, there are 10 structures classified as VGCCs, and in particular the authors of this work, have focused their attention on one of them: Cav1.2, whose inhibition has proved to be a strategic key against hypertension and myocardial ischemia. Several classes of small molecules, such as dihydropyridine, benzothiazepine, and phenylalkylamine have shown inhibitory capacities towards Cav, but the modulation mechanism is still unclear. As in our case study, they tried to shed light on this issue thorough homology modeling approach, furthermore, they then used the models produced for dynamic analysis through a molecular dynamics approach. They used as template the resolved structure of the Cav1.1 of the rabbit (defined at near atomic resolution of 3.6 Åand deposite in PDB with code:5GJV), as representative of the closed state.

4.2.1 Methods

The protein sequence related to the 4 transmembrane domains plus the connection sections was recovered from the UniProt database and aligned with the cryo-EM structure of Cav1.1 of rabbit, showing an identity around 65% (see figure 4.3). Authors built 5 models using I-TASSER modelling package, which they subsequently checked with: PROCHECK (as tool to evaluate stereochemical quality) and WHATS-CHECK (as probe of bond geometry). The selected model at the end of structure evaluation, has been embedded in 20Åthick palmitoyl-oleoyl-phosphati-dylcholine (POPC) lipid bilayer, as representative of the membrane environment. This step was performed using CHARMM-GUI membrane builder [31] setting 570 POPC molecules, system was fully solvated below and above with 66,951 TIP3P water molecule and was ionized with 195 Ca^{2+} and 369 Cl^{-} to reach the ionic concentration of 150

mM. Moreover, they produced a model representative of open state starting from deposited structures of NaV1.4 of eel (PDB code:5XSY) as a template, embedding the obtained systyem in 543 POPC lipid molecules, solvated by 95,530 TIP3P water molecules and was ionized with 277 Ca^{2+} and 532 Cl^{-} to reach the ionic concentration of 150 mM. All molecular dynamics simulation were performed using as force field CHARMM36 and as computational software NAMD-2.9 package. The simulations were conducted under these condictions:

- cutoff of 12Åfor the Van der Waals interactions
- cutoff of 10Åfor electrostatic interactions.
- the temperature has been set at 303.15K with 1 atm of pressure using Langevin thermostat.

First, the restrained system was subjected to an energy minimization and equilibrated with six steps: 2-steps of NVT dynamics using 1 fs time step and 4-steps of NPT dynamics performed with 2 fs time step. Finally a classical molecular dynamics simulation of 100 ns was conducted without any restrictions. In an attempt to trigger the channel opening, after the 100 ns simulation, an external electric field was introduced along the Z-axis, calibrating the electric field in order to generate a voltage potential of -40mV. The simulation with the effective field was carried out for 200 ns.

Percent identity matrix	Human $Ca_V1.2$	Rabbit $Ca_V1.1$	Eel $Na_V1.4$	Bac Ca_VAb
Human $Ca_V1.2$	100%	65.08%	18.26%	13.80%
Rabbit $Ca_V1.1$	65.08%	100%	20.79%	16.10%
Eel $Na_V1.4$	18.26%	20.79%	100%	16.10%
Bac Ca_VAb	13.80%	16.10%	16.10%	100%

FIGURE 4.3: Identity ratio between the sequences taken into consideration: Humans Cav1.2, Rabbit Cav 1.1, Eel Nav1.4 and CavAb(the bacterial homologue of *Arcobacter bultzeri*, the same species form which Dimos et al taken the reference template for generate their model).

4.2.2 Results and Discussions

The five models produced have been subjected to various evaluation tools, among which: Ramachandran Plot assessment (figure 4.4) also used in the work of this thesis. A comparison between the reliability of the models produced is therefore possible in the first generalization. Bearing in mind that different proteins have been modeled, which however have different analogies that unite them, the performances of the two tools used in homology modeling can be compared (I-TASSER vs SWISS-MODEL). The results are reported in 4.1, in particular, results of the WTM3 and WT6A models take

Models	**Ramachandran plot (%)**		
	Favored	Allowed	Disallowed
No.1	77.3	19.3	3.4
No.2	79.7	17.1	3.2
No.3	77.7	19.6	2.6
No.4	74.1	22.4	3.5
No.5	75.7	21.7	2.6
WTM3	78.5	17.1	4.4
WT6A	81.0	13.8	5.2

TABLE 4.1: Ramachandran plot evaluation results for the 5 models produced by [24] and the 2 WT (WT MOESM3-based and WT 6A90-based) we produced.

into account the structures in their integrity, therefore also include the connection sections between the transmembrane helices. Moreover, the simulation with applied electic field improved to mimic the condition that bring to the open state of the channel have not shown any ion influx along the pore. They then resorted to a steered molecular dynamics approach, which were able to identified several key residues that govern the calcium ion binding and channel gating proccess. Steered molecular dynamics was carried out by applying an external force on the calcium atoms, imposing a constant speed of v=0.00004Å/fs and having an elastic constant of k=4kcal/molÅalong the Z direction for a maximum displacement of 60Å. A total of 11 replicas were recorded, and all of them have shown a huge barrier when the ion reaches the internal gate region of the channel. Three sites associated with high energy have been address to be potential binding sites in the selectivity filter (figure 4.5):

- Site 1) formed by four deprotonated Glu residues (Glu363 (DI), Glu706 (DII), Glu1135 (DIII), and Glu1464 (DIV)) from the four transmembrane domains.
- Site 2) due to the coordination of the ion with a Glu residue from site 1 and to a residual Thr of site 3.
- Site 3) formed by four Thr residues (Thr361 (DI), Thr704 (DII), Thr1133 (DIII), and Thr1462 (DIV)).

4.2.3 Conclusion

The authors of this article reported for the first time an atomistic model of the voltage-gated calcium channel enriching it with a molecular dynamics analysis. Although they applied an electric field in order to mimic the conditions that lead to the opening of the channel, they failed to register any passage of ions along the channel; but using a steered molecular dynamics

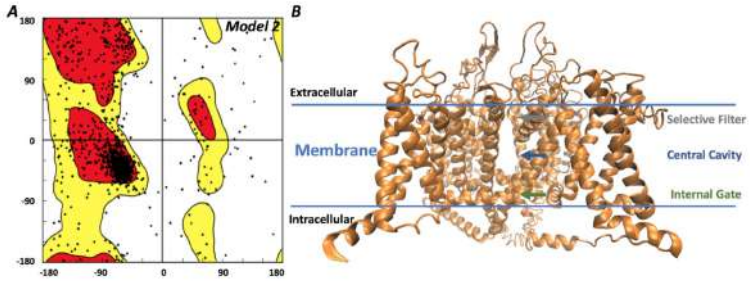

FIGURE 4.4: A)The Ramachandran plot of their No.2 model and B) rappresentation of the channel embedded in the plasmatic membrane [24]

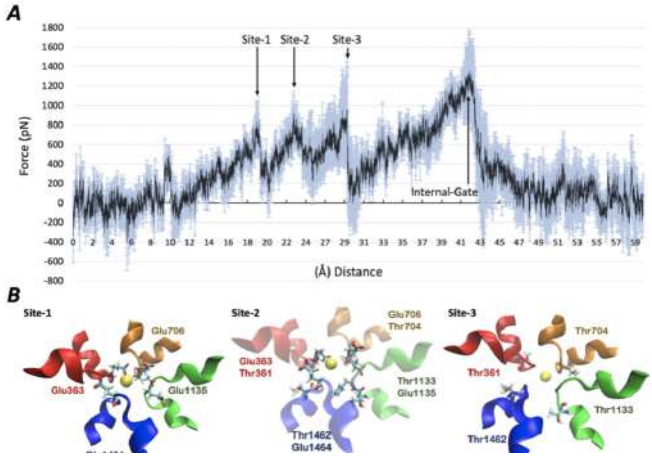

FIGURE 4.5: A) Force profile of ion registered along Z-axis B) graphic reconstruction of binding sites [24].

approach, they were able to highlight three possible sites involved in ion selectivity. Concluding that a molecular dynamics approach could be useful for understanding the characteristics of the open state.

Chapter 5

Results

As previously discussed, the family of dependent voltage sodium channels is involved in different levels of communication and amplification of signals through localized nerve fibers throughout the body. Remarkably, mutations in the sodium channel that is expressed at the level of the preferred nervous system (NaV1.7) is itself proved to be closely related to the onset of neuropathies. To shed light on which is the connection between mutations and the onset of neuropathies, it would be crucial to have a structural model representative of each mutation. The production of these models with experimental techniques, which involve the grafting of the modified genetic trait in an organism suitable for its expression, the over-expression, and purification for the measurements of the protein; even for a small sample set it would require a large investment of time and money. For this reason, it was decided to adopt computational models. In particular, since there are available orthologous structures deposited to the one under examination, it was decided to develop the models using homology modeling techniques.

In accordance with the work carried out by Dimos et al, NP_002968.1 was chosen as the reference sequence. From the NCBI site, it is classified as the first isoform of the transcript variant 1 (NM_002977.3: identification code for the chromosomal sequence)[27]. This was the starting point on which the mutations were carried out, taking as a reference the same mutations considered by Dimos et al [32].

5.0.1 Deepening to the reasons that led to the choice of the NP002968.1 sequence as WT

To characterize the chosen sequence, it was compared with the isoforms made available by the UniProt site ([15]), which reports 4 isoforms (IDs: Q15858, Q15858-2, Q15858-3, Q15858-4) for this protein [15] (From the NCBI site, it is classified as the first isoform of the transcript variant 1 (NM_002977.3) [27]). Using the align tool made available by the same site, there is an identity among the sequences of 99.296%, whose differences reside substantially in two points. First of all, the WT sequence considered presents a deletion concerning the isoform Q15858, which involves the absence of a stretch of 11 amino acids between 648-658 (ref Q15858). The second and more relevant point, the presence of an arginine replacing a tryptophan in the 1150 position (W1150R). Department of Neurology and Center for Neuroscience and Regeneration Research of Yale University School of Medicine highlighted how

the replacement of R with W involves a shifts activation voltage dependence 7.9 to 11mV in a depolarizing direction (based on the presence or absence of β subunits). This mutation results in the doubling of the signal frequency in small DRG neurons, thus making this isomorphism closer to gain-of-function mutations. [23].

A total of 30 mutations closely correlated with the onset of painful-related syndromes in humans and 55 mutations that did not show clear links with neuropathologies (homologs Single Nucleotide Polymorphism) were cataloged in a special file in preparation for the next steps (these are the same mutations taken into consideration in the work of Dimos et al.). As follow figures representing the topological position of each single mutation along the peptide chain have been prepared, figure 5.4 and table 5.2 refer to gain-of-function mutations, figure 5.5 and table 5.3 are refered to hSNP mutations.

5.1 Homology Modeling

As a preliminary phase to the production of the structures by homology-modeling, an excel file was prepared with all the sequences listed, in each column of the file a sequence has been reported, and each row shows the corresponding amino acid in sequence (for a total of 1977 lines, how many the total number of amino acids that make up the subunit-α of the channel). Each box in the .csv file returns the acidic amino in the simple convention: ARG=R, HIS=H, LYS=K.. (RHK..) it is easily interpreted by the SWISS-MODEL program. In summary, a file was created with 86 columns (30 relating to neuropathic mutations and 55 for isomorphisms, plus the WT sequence) showing in each cell an amino acid of the sequence. Having considered point mutations, each of the 85 columns (neuropathies + hSNP) differ from the WT column only for one amino acid in the whole sequence, and obviously between them for two points. Once this material was prepared, we moved on to the actual homology-modeling phase: Swiss-Model was the tool chosen to perform this step, it was decided to prefer this instrument to MEMOIR (tool used in the reference publication), as it offered the advantage of managing sequences with more than 1500 amino acids (which are instead the maximum limit for MEMOIR). It has therefore allowed us to generate representative models of the entire sequence, otherwise, Dimos et al, had to model each single domain using MEMOIR and then unify them before performing the refining with YASARA.

Swiss-Model is an easily usable tool via its online graphical interface, from the Workspace it has been chosen the functionality that allows to carry out the homology using templates provided by the user (figure 5.1). Each mutations were uploaded in plain text format and validate and a structure was subsequently produced by loading, one at a time, the three templates taken as reference structures.

The first template was chosen in an attempt to reproduce the results of Dimos et al; this template was obtained through the homology modeling of the sequences of the 4 domains of human Nav1.7 in relation to the deposited PDB 3RVY structure (NaVAb, [56]). As they reported, the NaVAb (

homologous protein synthesized by *Acrobacter bultzeri*), template shared 28% sequence identity for DI, 24% for DII, 28% for DIII and 28% for DIV (overall 27% sequence identity). in this model the gap regions of the sequence-template alignment are missing:

- **DI**: 269-340
- **DI-DII**: 416-726
- **DII-DIII**: 967- 1175
- **DIII-DIV**: 1458-1498

to improve the goodness of the model, they have extended the S6 helix of each domain through an *Ab Initio* approach, by mean Iterative Threading ASSEmbly Refinement (I-TASSER) server [72]. This template has been renamed as MOESM3 (M3).

The other two templates were chosen, in one case because the structure derives from the homologous protein expressed by a eukaryotic organism, therefore evolutionarily much closer to humans. In the second case, we are always considering a prokaryotic organism, but in this case with the conformation of the channel in open-state, thus offering the possibility of exploring dynamic variations undergone by the protein, through a frame of different equilibrium position.

The identification codes of both these two structures are : 6A90 ([57]), 5HVX ([58]) respectively.

- **6A90**: This template refers to the structure in the closed-state of voltage-gated sodium channel of the *American Periplaneta* (NaVPas) [62]. This structure was obtained through cryo-EM techniques and, as mentioned above, refers to an eukaryotic homologous protein, so it is closer to humans under different points. First of all, there is a difference between the template used by Dimos et al, because it has a greater identity with the human sequences of interest. Furthermore, it is also constituted by 3-dimensional rewinding, in four domains of a single peptide, unlike the NaVAb homo-tetramer.

  The alignments of humans Domains and NaVPas domains shown high level of identity (alignment snapshots in Appendix A: A.1,A.2,A.3 and A.4).

- **5HVX**: Lastly 5HVX (NavMs) despite being a protein produced by a prokaryotic organism (*Magnetococcus marinus*), was an interesting template because the PDB carries information of another important structural arrangement: the open conformation of the canal. As this homolog of bacterial origin, a substantial difference persists with the eukaryotic realm, since its structure is not due to the single rewinding of a peptidic chain; as in the case of eukaryotes (the same problem of evolutionary remoteness, also present in the NaVAb template). The alignments of humans Domains and NaVMs domain are shown in the Appendix A A.5, A.6, A.7 and A.8.

Results of alignments NaV1.7 vs NaVPas and NaVMs for each domains					
	NaV1.7	NaVPas	Identity (%)	NaVMs	Identity (%)
DI	112 - 410	141 - 413	46	13 - 229	16.23
DII	715 - 978	520 - 740	45.59	13 - 229	20.59
DIII	1169 - 1477	858 - 1108	41.75	13 - 229	20.71
DIV	1486 - 1784	1172 - 1410	47.16	13 - 229	18.12

TABLE 5.1: Align NaV1.7 vs NaVPas and NaVMs.

5.1.1 What exactly does SWISS-MODEL

As first point, SWISS-MODEL copies cartesian coordinates using the sequence alignment as reference to obtain so called raw-model. The raw-model might still contain gaps or lacks sidechains, it is at this stage that specific functions intervene to complete the model. This tool uses two algorithms to perform the minimization: Steepest Descent and limited-memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS). The first algorithm is one of the oldest methods for the minimization of a general nonlinear function, it was first proposed by Cauchy in 1847 [13]. Although this algorithm has a very slow convergence ratio, it has a sold and easily implementable structure, so it still finds space in some applications. the algorithm is defined as follows: What we try to minimize is the potential energy of the system, to do this we define the vector **r** as the vector of all $3N$ coordinates, then we go on to analyze how the function varies around its coordinates. Initially a maximum displacement h_0 (e.g. 0.01 nm) must be given. **F** and potential energy are then determined and the new positions accordingly:

$$\mathbf{r}_{n+1} = \mathbf{r}_n + \frac{\mathbf{F}_n}{max[\,|\,\mathbf{F}\,|\,]} h_n \tag{5.1}$$

where h_n is the maximum displacement and $\mathbf{F}_n$ is the force. The notation max[**F**] means the largest scalar force on any atom. The forces and energy are again calculated for the new positions [1]. The itterations end when either the user has set a maximum number of itterations or has fallen below a threshold energy value. In this case, the number of maximum itterations has been set to 10, as a limit if all the clashes between the atoms have not been removed. The second algorithm, limited-memory Broyden-Fletcher-Goldfarb-Shanno, is an optimization algorithm in the family of quasi-Newton methods. The original BFGS algorithm works by successively creating a good approximation of the Hessian matrix, but the memory requirement for this calculation is however proportional to the square of the number of particles (storing a dense n x n quantity of data). It made the use of this tool impractical in the study of large systems, such as the study of biomolecules, while L-BFGS stores only a few vectors that represent the approximation implicitly. ProMod3 runs up this algorithm up to a maximum of 20 times, always if all clashes are not removed first. (see second step in figure 3.15).

Mutation Pain-related					
IEM	⊙	PEPD	⊙	SFN	⊙
I136V	1	V1298D	19	R185H	25
S211P	2	V1298F	19	I228M	26
F216S	3	V1299F	19	I739V	27
I234T	4	G1607R	19	G856D	28
S241T	5	M1627K	19	M932L	29
N395K	6	A1632E	19	M1532I	30
V400M	7				
L823R	8				
I848T	9				
L858H	10				
L858F	11				
A863P	12				
V872G	13				
P1308L	14				
V1316A	15				
F1449V	16				
W1538R	17				
A1746G	18				

TABLE 5.2: List of pain-related mutation.

Mutation not associated with abnormalities					
hSNP	⊙		⊙		⊙
S126A	31	R1207K	50	H531Y	69
L127A	32	T1210N	51	M1532V	70
M145L	33	I1235V	52	E1534D	71
N146S	34	N1245S	53	Y1537N	72
V194I	35	L1267V	54	T1548S	73
L201V	36	T1398M	55	H1560C	74
N206D	37	I1399D	56	H1560Y	75
T370M	38	D1411N	57	V1565I	76
E759D	39	K1412E	58	I1577L	77
A766T	40	K1412I	59	D1586E	78
A766V	41	K1415I	60	T1590K	79
I767V	42	S1419N	61	T1590R	80
T773S	43	V1428I	62	T1596I	81
V795I	44	A1505V	63	V1613I	82
A815S	45	S1509A	64	D1662A	83
D890E	46	S1509T	66	G1674A	84
D890V	47	Q1530D	66	K1700A	85
T920N	48	Q1530K	67		
K1176R	49	Q1530P	68		

TABLE 5.3: List of mutation that are not associated with protein function disorders.

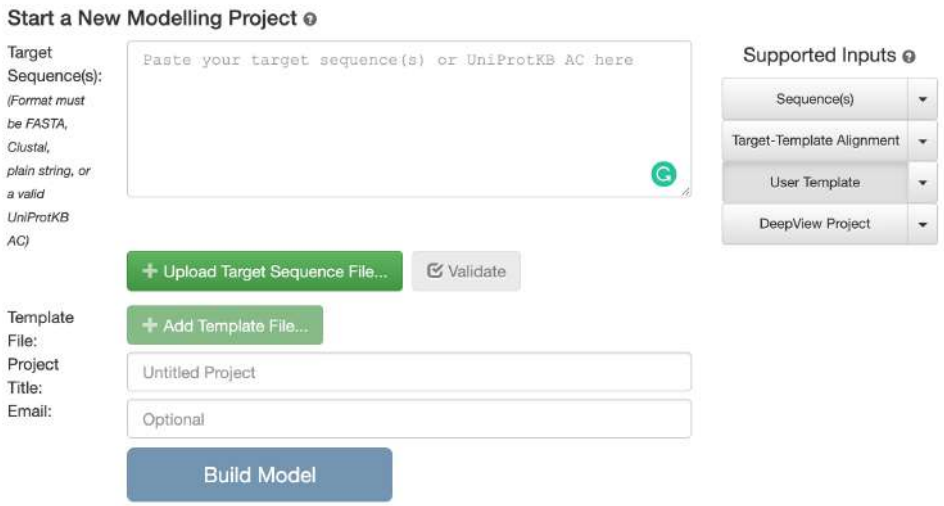

FIGURE 5.1: Snapshot of Swiss-Model Workspace [46].

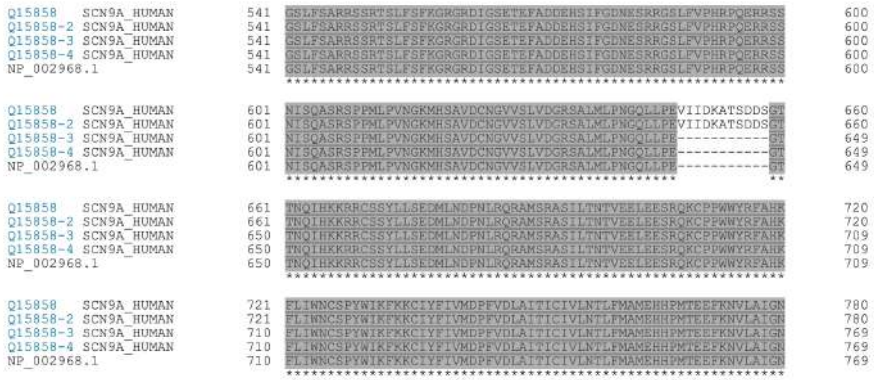

FIGURE 5.2: Align sequences

5.2 Energy Minimization

At the end of the previous step, 86 structures were obtained for each template used, the coordinates of these structures were used as starting geometries for a further refinement of the models. This second step was performed using the FG-MD online tool, that using the same algorithms mentioned above, it aims to probe the energy landscape and converge to the minimum energy (native state). Also in this case, the online tool offers a user friendly interface, and in a completely automated manner takes the uploaded PDB file, carries out remote minimization and sends the refined structure to a reference e-mail address (figure 5.6). FG-MD relies on a modified version of LAMMPS, that is a molecular dynamics (MD) code able to models ensembles of particles in a liquid, solid, or gaseous state. It can handle atomic, polymeric, biological, solid-state, granular, coarse-grained, or macroscopic systems using a variety of interatomic potentials (force fields) and boundary conditions. Its code is written in C++ with its earlier version written in F77 and F90 [53]. The details on the set of parameters used are as follows:

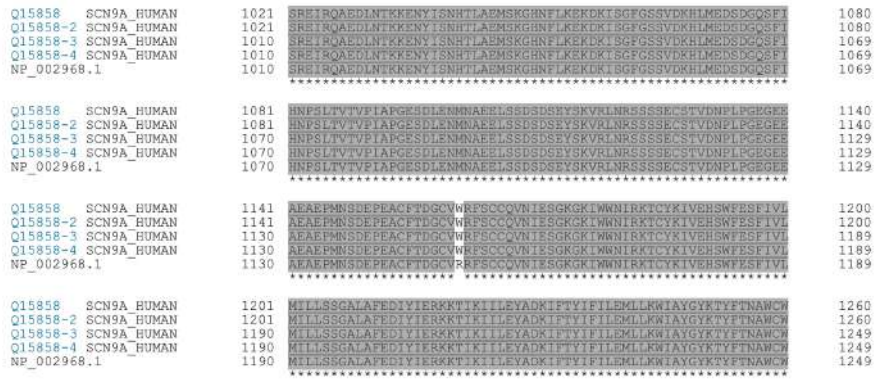

FIGURE 5.3: Align sequences.2

FG-MD setting parameters	
units	real
neigh_modify	every 10
atom_style	full
bond_style	harmonic
angle_style	harmonic
dihedral_style	hybrid harmonic
pair_style	lj/cut/coul/cut/ 10.0 10.0
pair_modify	mix arithmetic
boundary	p p p
special_bonds	amber
thermo	1
thermo_style	multi
timestep	2.0
minimize	1.0e-3 | 1.0e-6 | 100 | 1000
run	10000

TABLE 5.4: List of setting parameters improved during structures refinments.

5.3 Quality Assessment

Each structures obtained in this way, was subjected to an evaluation of the model, through a special scoring function designed for membrane proteins. QMEANBrene is still a model evaluation tool devised by the Swiss Institute of Bioinformatics; it is easily accessible from the same Swiss-Model workspace shown above, from the *Tool* panel (figure 5.7).

The evaluation tool has shown that the models produced are of high quality in the transmembrane regions (values very close to 1). While in connection areas (inter-domain loops) the reliability of the model is significantly lower. The values found were reported punctually for each amino acid of interest (tab:5.5, 5.6 are refered to models-MOESM3-template based and tab: 5.7, 5.8 are refered to models-6A90-template based), while a visual and graphic representation of the chain as a whole is shown in the figures: 5.8,5.9,5.10 and

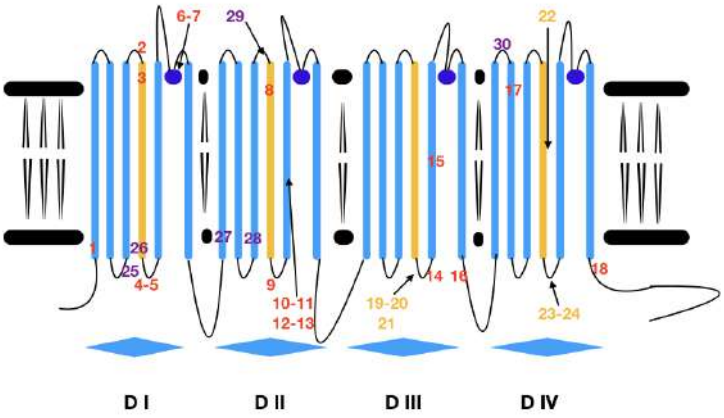

FIGURE 5.4: Schematic illustration of the poly-peptide chain and localization of mutations associated with pain conditions, indicating the pathologies with a different color.

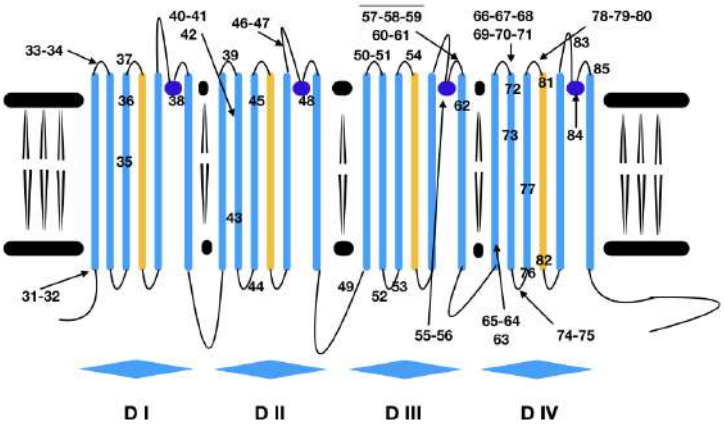

FIGURE 5.5: Schematic representation of mutation locations not associated with pain syndromes.

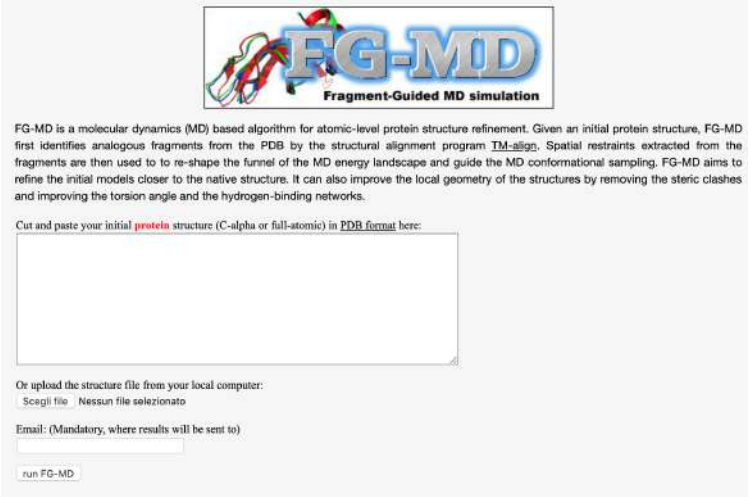

FIGURE 5.6: Snapshot of FG-MD Workspace [73].

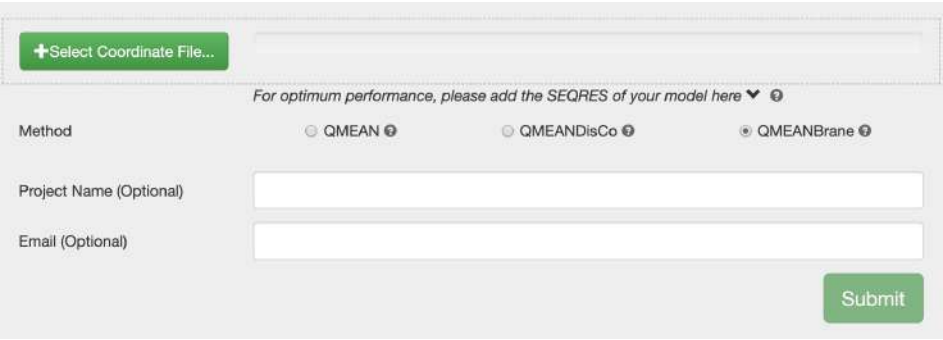

FIGURE 5.7: Snapshot of QMEANBrane interface [47]

5.11.

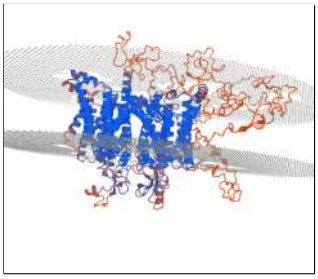

FIGURE 5.8: QMEANBrane analysis of WTM3.

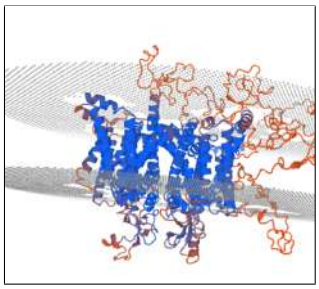

FIGURE 5.9: QMEANBrane analysis of WT6A.

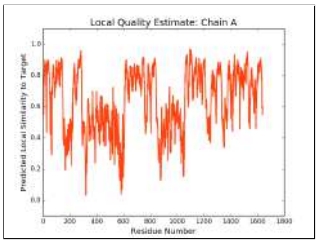

FIGURE 5.10: QMEAN-Brane: local quality estimation of WTM3.

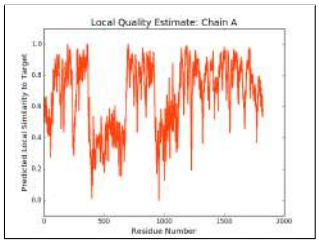

FIGURE 5.11: QMEAN-Brane: local quality estimation of WT6A.

QMEANBrane is an assessment tool that provides an estimate of the quality of each model based on its own database of reference protein structures. Knowledge-based assessment techniques are a reliable means of measurement, however, further confirmation was also sought through the RAMPAGE model evaluation. RAMPAGE evaluates the quality of the models based on the values reported in the pdb-files, of the angles ψ and ϕ along to peptide-sequence. Given the low quality in the regions outside the membrane, and because the mutations of interest are all located in the transmembrane or in close proximity, the RAMPAGE value was performed for the protein segments characterized by a secondary helix structure. As a result, in the case of the model produced starting from the MOESM3 model, the amino acids that fall within the favorable region are 95.8% (percentage that increases to 97.9% for the WT produced starting from the 6A90 model); the residues that fall within the permitted region are respectively 4% (28 residues) for MOESM3 and 2% (16residues) for 6A90. For both structures only one amino acid falls in the forbidden region: Arg 896 (MOESM3) and Ile 720(6A90), either are not taken in account for local analysis (plots are reported in figures 5.12 and 5.13). This tool also offered us a yardstick with the results shown in the publication that inspired this work: at first sight, and relying on the results of

MOESM3 QMEANBrane of WT and mutations pain-related				
	Mut	WT(Swiss)	WT(FG-MD)	Mut(FG-MD)
IEM	I136V	0.87	0.86	0.9
	S211P	0.72	0.73	0.8
	F216S	0.84	0.8	0.82
	I234T	0.83	0.81	0.84
	S241T	0.92	0.88	0.92
	N395K	0.84	0.85	0.83
	V400M	0.95	0.92	0.94
	L823R	0.81	0.82	0.78
	I848T	0.8	0.79	0.79
	L858H	0.82	0.8	0.77
	L859F	0.82	0.8	0.61
	A863P	0.92	0.9	0.87
	V872G	0.89	0.87	0.88
	P1308L	0.63	0.61	0.74
	V1316A	0.86	0.85	0.88
	F1449V	0.89	0.84	0.89
	W1538R	0.58	0.6	0.57
	A1746G	0.91	0.91	0.91
PEPD	V1298D	0.86	0.77	0.87
	V1298F	0.86	0.77	0.72
	V1299F	0.89	0.77	0.81
	G1607R	0.85	0.83	0.76
	M1627K	0.77	0.77	0.74
	A1632E	0.84	0.82	0.77
SFN	R185H	0.47	0.29	0.38
	I228M	0.84	0.77	0.77
	I739V	0.84	0.84	0.87
	G856D	0.8	0.72	0.71
	M932L	0.78	0.76	0.73
	M1532I	0.56	0.62	0.49

TABLE 5.5: Columns in the middle report QMEANBrane values of WT model after Swiss-Model modeling and after FG-MD refinement; the last column report the same value of mutated structure after FG-MD. This table refers to models generated from MOESM3 template

MOESM3 **QMEANBrane of WT and mutations not related with pain-syndrome**							
Mut	WT(Swiss)	WT(FG-MD)	Mut(FG-MD)	Mut	WT(Swiss)	WT(FG-MD)	Mut(FG-MD)
S126A	0.82	0.81	0.77	A1505V	0.82	0.73	0.72
L127A	0.82	0.81	0.84	S1509A	0.89	0.9	0.93
M145L	0.67	0.54	0.77	S1509T	0.89	0.9	0.87
N146S	0.71	0.53	0.73	Q1530D	0.58	0.69	0.56
V194I	0.89	0.85	0.84	Q1530K	0.58	0.69	0.42
L201V	0.81	0.8	0.82	Q1530P	0.58	0.69	0.62
N206D	0.73	0.73	0.6	H531Y	0.62	0.7	0.47
T370M	0.55	0.64	0.56	M1532V	0.56	0.62	0.62
E759D	0.65	0.65	0.74	E1534D	0.44	0.47	0.58
A766T	0.77	0.76	0.78	Y1537N	0.54	0.55	0.64
A766V	0.77	0.76	0.81	T1548S	0.9	0.88	0.89
I767V	0.75	0.76	0.85	H1560C	0.67	0.67	0.77
T773S	0.82	0.82	0.83	H1560Y	0.67	0.67	0.58
V795I	0.78	0.84	0.77	V1565I	0.79	0.8	0.71
A815S	0.77	0.76	0.88	I1577L	0.81	0.81	0.85
D890E	0.52	0.54	0.51	D1586E	0.75	0.75	0.6
D890V	0.52	0.54	0.51	T1590K	0.5	0.5	0.53
T920N	0.62	0.64	0.73	T1590R	0.5	0.5	0.58
K1176R	0.37	0.32	0.54	T1596I	0.82	0.84	0.75
R1207K	0.65	0.66	0.68	V1613I	0.8	0.78	0.78
T1210N	0.72	0.75	0.77	D1662A	0.54	0.52	0.58
I1235V	0.71	0.71	0.86	G1674A	0.79	0.84	0.84
N1245S	0.75	0.72	0.74	K1700A	0.64	0.63	0.67
L1267V	0.78	0.79	0.89				
T1398M	0.8	0.77	0.7				
I1399D	0.76	0.74	0.79				
D1411N	0.54	0.55	0.55				
K1412E	0.56	0.55	0.62				
K1412I	0.56	0.55	0.67				
K1415I	0.71	0.61	0.67				
S1419N	0.81	0.84	0.84				
V1428I	0.88	0.88	0.86				

TABLE 5.6: Columns in the middle report QMEANBrane values of WT model after Swiss-Model modeling and after FG-MD refinement; the last column report the same value of mutated structure after FG-MD. This table refers to models generated from MOESM3 template

6A90 QMEANBrane of WT and mutations pain-related				
	Mut	WT(Swiss)	WT(FG-MD)	Mut(FG-MD)
IEM	I136V	0.84	0.85	0.89
	S211P	0.66	0.68	0.78
	F216S	0.74	0.78	0.83
	I234T	0.88	0.86	0.86
	S241T	0.99	0.93	0.94
	N395K	0.88	0.86	0.88
	V400M	1.0	0.99	0.94
	L823R	0.74	0.74	0.74
	I848T	0.73	0.78	0.75
	L858H	0.87	0.88	0.82
	L859F	0.87	0.88	0.89
	A863P	0.98	0.98	0.97
	V872G	0.87	0.86	0.88
	P1308L	0.87	0.8	0.76
	V1316A	0.91	0.9	0.93
	F1449V	0.95	0.95	0.89
	W1538R	0.92	0.96	1.0
	A1746G	0.98	0.96	0.99
PEPD	V1298D	0.86	0.77	0.87
	V1298F	0.9	0.92	0.89
	V1299F	0.9	0.92	0.93
	G1607R	0.87	0.85	0.76
	M1627K	0.89	0.88	0.9
	A1632E	0.9	0.9	0.84
SFN	R185H	0.61	0.46	0.6
	I228M	0.74	0.72	0.72
	I739V	0.89	0.92	0.96
	G856D	0.85	0.86	0.8
	M932L	0.79	0.77	0.79
	M1532I	0.8	0.84	0.79

TABLE 5.7: Columns in the middle report QMEANBrane values of WT model after Swiss-Model modeling and after FG-MD refinement; the last column report the same value of mutated structure after FG-MD. This table refers to models generated from 6A90 template

6A90 **QMEANBrane of WT and mutations not related with pain-syndrome**							
Mut	WT(Swiss)	WT(FG-MD)	Mut(FG-MD)	Mut	WT(Swiss)	WT(FG-MD)	Mut(FG-MD)
S126A	0.74	0.72	0.77	A1505V	0.84	0.87	0.82
L127A	0.78	0.78	0.77	S1509A	0.92	0.93	0.89
M145L	0.59	0.73	0.77	S1509T	0.92	0.93	0.97
N146S	0.57	0.79	0.69	Q1530D	0.67	0.87	0.67
V194I	0.88	0.91	0.88	Q1530K	0.67	0.87	0.73
L201V	0.71	0.69	0.75	Q1530P	0.67	0.87	0.77
N206D	0.54	0.47	0.32	H531Y	0.62	0.88	1.0
T370M	0.92	0.93	0.92	M1532V	0.8	0.84	0.88
E759D	0.72	0.87	0.9	E1534D	0.83	0.88	0.88
A766T	0.85	0.88	0.92	Y1537N	0.84	0.91	0.91
A766V	0.85	0.88	0.92	T1548S	0.91	0.93	0.94
I767V	0.82	0.84	0.93	H1560C	0.75	0.64	0.65
T773S	0.89	0.9	0.93	H1560Y	0.75	0.64	0.7
V795I	0.82	0.75	0.69	V1565I	0.74	0.72	0.72
A815S	0.66	0.66	0.74	I1577L	0.84	0.86	0.9
D890E	0.63	0.53	0.59	D1586E	0.48	0.58	0.42
D890V	0.63	0.53	0.49	T1590K	0.39	0.47	0.38
T920N	0.92	0.92	0.89	T1590R	0.39	0.47	0.42
K1176R	0.74	0.74	0.64	T1596I	0.79	0.8	0.81
R1207K	0.5	0.59	0.71	V1613I	0.74	0.77	0.78
T1210N	0.85	0.82	0.69	D1662A	0.62	0.7	0.63
I1235V	0.87	0.84	0.83	G1674A	0.9	0.87	0.85
N1245S	0.76	0.74	0.76	K1700A	0.59	0.64	0.68
L1267V	0.46	0.34	0.05				
T1398M	0.92	0.9	0.9				
I1399D	0.78	0.86	0.86				
D1411N	0.71	0.67	0.7				
K1412E	0.73	0.68	0.65				
K1412I	0.73	0.68	0.63				
K1415I	0.67	0.66	0.66				
S1419N	0.7	0.76	0.75				
V1428I	0.89	0.88	0.86				

TABLE 5.8: Columns in the middle report QMEANBrane values of WT model after Swiss-Model modeling and after FG-MD refinement; the last column report the same value of mutated structure after FG-MD. This table refers to models generated from 6A90 template

RAMPAGE, our models obtained a higher score (95.8% compared to 88.5%). However, we must take into account that the models we evaluated have been deprived of all the secondary structures with the exception of the propeller rewinding, applying the same selection to the MOESM3 template, the percentage of residues in favoured region rises to 96.6%.

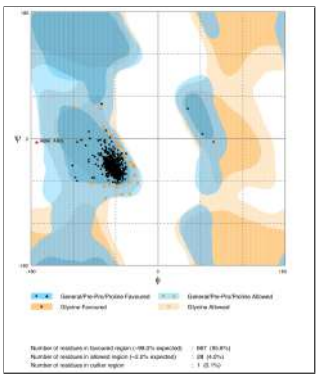

FIGURE 5.12: Ramachandran Plot Analysis of WTM3 model after FG-MD minimization.

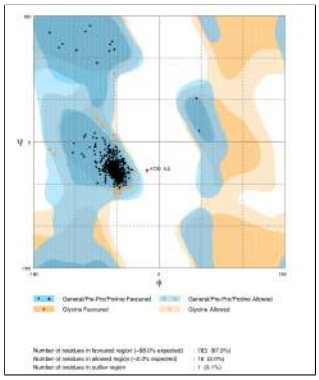

FIGURE 5.13: Ramachandran Plot Analysis of WT6A model after FG-MD minimization.

5.4 Production of Residue Interaction Network (RIN)

The set of models thus generated and evaluated, is a set of extremely complex data whose interpretation is not trivial. Transforming each structure into a graph at first reduces the complexity of the system and secondly, but more importantly, it can help in the search for a common pattern that associates neuropathic mutations.

The graphs were constructed by considering each residue as a node and the arcs identify the interactions between the residues, the interactions considered are hydrogen bonds, Van der Waals, ionic bridge, π-π stacking and π-cation interaction. Of each structure two graphs have been built, the first each arc represents an interaction between the residues if this comes to be established at no more than 5 Å, in the second graph, however, each interaction has its own cutoff distance. The conversion from pdb to graph was performed using the online tool RING2.0 [52].

5.5 Network Analysis

The RINs supplied in output from RING2.0 have been analyzed using different approaches. A first evaluation involved the use of the Cytoscape software, more precisely the use of two apps: RINalyzer and CytoNCA. With these tools, some metrics characterizing the graphs have been calculated, in

Ramachandran Plot Analysis of WT and mutations pain-related					
	Mut	MOESM3(SW)	MOESM3(FG-MD)	6A90(SW)	6A90(FG-MD)
IEM	I136	√	√	√	√
	S211	√	√	×∘	×∘
	F216	√	√	√	√
	I234	√	√	√	√
	S241	√	√	√	√
	N395	√	√	√	√
	V400	√	√	√	√
	L823	√	√	√	√
	I848	√	√	√	√
	L858	√	√	√	√
	A863	√	√	√	√
	V872	√	√	√	√
	P1308	√	√	√	√
	V1316	√	√	√	√
	F1449	√	√	√	√
	W1538	√	√	√	√
	A1746	√	√	√	√
PEPD	V1298	√	√	√	√
	V1299	√	√	√	√
	G1607	√	√	√	√
	M1627	√	√	√	√
	A1632	√	√	√	√
SFN	R185	×	×	√	√
	I228	×∘	×∘	×∘	×∘
	I739	√	√	√	√
	G856	√	√	√	√
	M932	√	√	√	√
	M1532	×∘	√	√	√

TABLE 5.9: √= residue present in the model containing only the sequence sections having a secondary helical structure, ×=residue not present and ×∘= residual not present, but its first neighbor yes.

Ramachandran Plot Analysis of WT and mutations not related with pain-syndrome									
Mut	M3(SW)	M3(FG-MD)	6A(SW)	6A(FG-MD)	Mut	M3(SW)	M3(FG-MD)	6A(SW)	6A(FG-MD)
S126	√	√	√	√	A1505	√	√	√	√
L127	√	√	√	√	S1509	√	√	√	√
M145	×○	×○	×○	×○	Q1530	√	√	×○	√
N146	×	×	×	×	H1531	√	√	√	√
V194	√	√	√	√	M1532	×○.	√	√	√
L201	√	√	√	√	E1534	×	×○	√	√
N206	√	×	×	×	Y1537	×○	√	√	√
T370	×	×	√	√	T1548	√	√	√	√
E759	√	√	√	√	H1560	√	√	×	×
A766	√	√	√	√	V1565	×○	×○	×○	×○
I767	√	√	√	√	I1577	√	√	√	√
T773	√	√	√	√	D1586	√	√	×	×
V795	×○	×○	×○	×○	T1590	×	×	×	×
A815	√	×○	×	×	T1596	√	√	√	√
D890	×	×	×	×	V1613	√	√	×	×○
T920	×	×	√	√	D1662	√	√	×	×
K1176	×	×	√	√	G1674	√	√	√	√
R1207	√	√	×	×	K1700	×○	×	×	×
T1210	√	√	√	√					
I1235	√	√	√	√					
N1245	×○	×○	×○	×○					
L1267	√	√	×	×					
T1398	√	√	√	√					
I1399	√	√	√	√					
D1411	×○	×	×	×					
K1412	√	√	×	×					
K1415	√	√	×	×					
S1419	√	√	×	×					
V1428	√	√	√	√					

TABLE 5.10: √= residue present in the model containing only the sequence sections having a secondary helical structure, ×=residue not present and ×○= residual not present, but its first neighbor yes.

Strict cutoff interaction distance	
Interaction type	distance Å
Hydrogen bonds	3.5
Van der Waals	0.5
Ionic Bridge	4.0
π-π stacking	6.5
π-cation	5.0

TABLE 5.11: differentiated cutoff for each type of bond according to the strict parameters of the RING2.0 tool [52].

particular a metric seems to have been recognized to be discriminating in discerning gain-of-function mutations from mutations that do not alter protein functions. Finally, the networks were analyzed through a kernel that takes into account the variations to the connections suffered by each node, instead of the Shortest Path approach used in the calculation of the metrics.

5.5.1 Cytoscape Results

After turning all the structures produced into .xml files, the networks were easily viewed with Cytoscape. Metrics were determined for each network using RINalyzer and CytoNCA in succession (see figures 5.14 and 5.15).

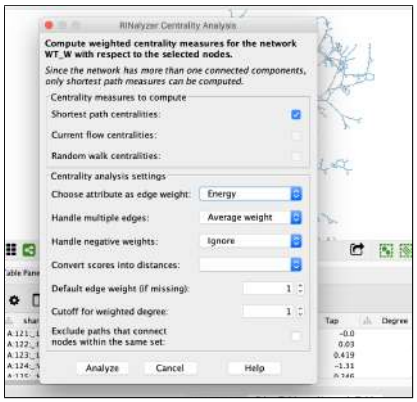

FIGURE 5.14: Frame of the control shall of RINalyzer.

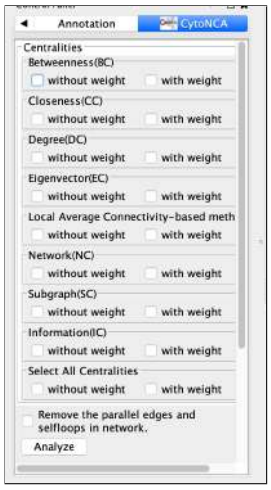

FIGURE 5.15: Frame of the control shall of CytoNCA.

The results obtained are in partial agreement with what was stated in the reference publication, even the recorded values of betweenness centrality show how the gain-of-function mutations lead to a greater departure from the WT standard. However, it is necessary to take into account the fact that the analyzed RIN contained a significantly higher number of nodes, compared to those generated by Dimos et al, due to the fact that the structures generated in this work took into account the protein connection traits between

domains. The difference between the two networks is 600 nodes (1000 nodes for the graphs of Dimos et al and 1600 for those produced in this work), this 60% increase has certainly made it difficult to directly compare the results obtained, as it is a parameter that significantly influences the metrics based on a shortest path approach. To be able to make more significant inferences further studies are needed on how the increase of nodes within the graphs influences the relationships between these metrics. Nevertheless, the same pattern found by the authors, was also found in the measurements made in this work (see figures 5.16,5.17,5.18 and 5.19).

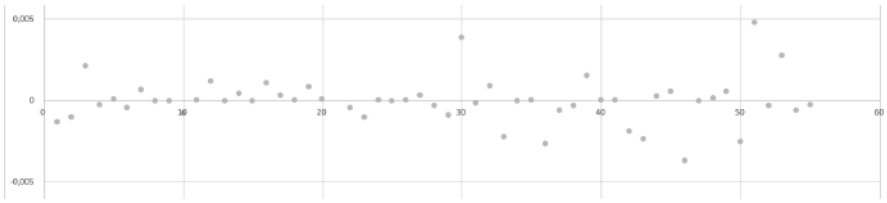

FIGURE 5.16: ΔB_{ct} measurements related to mutations not associated with neuropathies, for the structures produced starting from the template of Dimos et al.

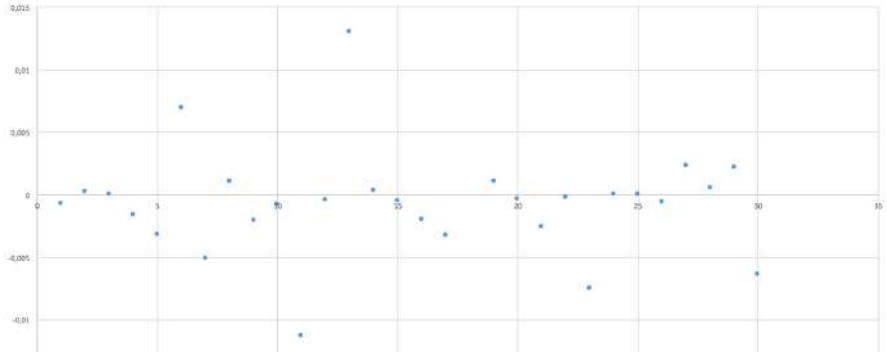

FIGURE 5.17: ΔB_{ct} measurements related to mutations associated with neuropathies, for the structures produced starting from the template of Dimos et al.

5.5.2 Kernel Methods Results

Taken in account only the data obtained through a survey on the variation of graph metrics we can see that betweenness seems to be the parameter that best discriminates between gain-of-function mutations and mutations not associated with neuropathies. This result is in agreement with what was stated in the reference publication ([32]), but it is through an analytical approach based on the use of graph kernel methods that better results have been obtained.

the results of the analyzes with the kernel approaches are reported in diagrams in which both row and column are shown in succession the graphs relating first to the gain-of-function mutations and in succession those relating to mutations not linked to pathologies.

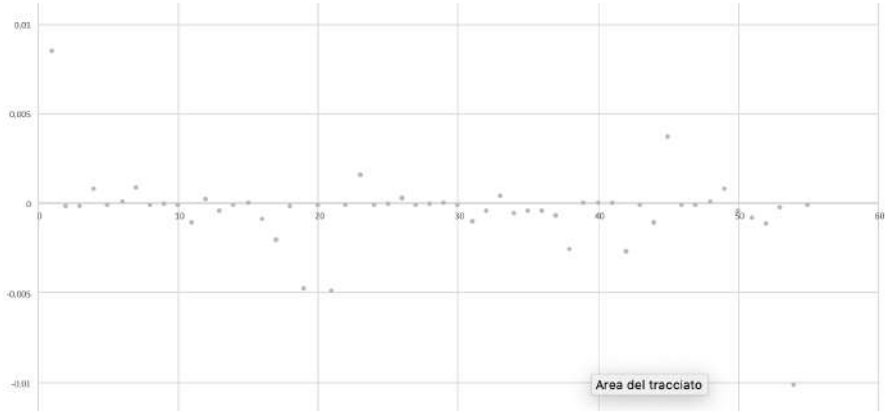

FIGURE 5.18: ΔB_{ct} measurements related to mutations not associated with neuropathies, for the structures produced starting from the template 6A90.

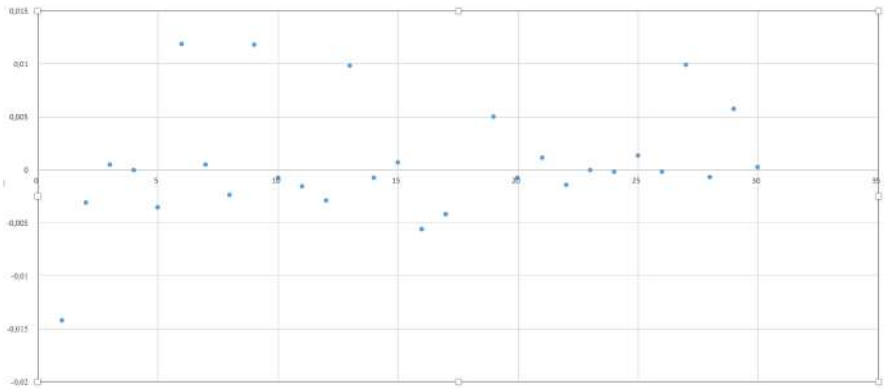

FIGURE 5.19: ΔB_{ct} measurements related to mutations associated with neuropathies, for the structures produced starting from the template 6A90.

The intersection between rows and columns identifies the level of similarity between the structures, obviously the diagonal that identifies the encounter between a graph in a row and itself in a column has the lightest color (the light tonality is normalized for each row). There are two graph kernel approaches used, the first always built on a shortest path-based algorithm:

- Shortest Path-based algorithm
  In the case of MOESM3-based graphs there is a greater similarity in the lower right quadrant (graph relating to non-pain-related structures), but it is not clear. The information on the upper quadrant, concerning pain-related graphs, is also poor (see figure 5.20). A single case deviates from all other structures (except of course with respect to itself) is the graph generated by the L1267V model, further investigations would be necessary to clarify this discrepancy.

  It is also possible to perform comparisons between graphs whose arcs take into account only one interaction at a time (thus trying to discriminate if there is a type of bond that most guides these phenomena: H-bond, ionic, π-π stacking, VdW or π-cation) and from this analysis the only ones that seem to identify common patterns between the graphs of gain-of-function and graphs of mutations not related to pain are the Ionic and the VdW diagrams (see figures 5.21 and 5.22).

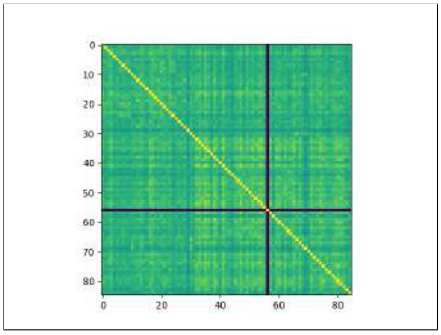

FIGURE 5.20: Results of clustering produced by improving Shortest path kernel, taking into account of all kind of interactions (for MOEM3-based graphs).

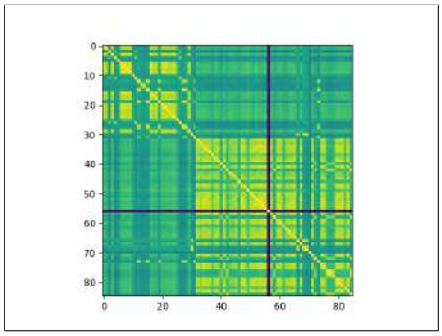

FIGURE 5.21: Results of clustering produced by improving Shortest path kernel, taking into account of ionic interactions (for MOEM3-based graphs).

Conversely, for the graphical representation of similarities between 6A90-based graphs, a common pattern among pain-related mutations is more evident (the upper right quadrant is characterized by a lighter color),

indicating that all 30 mutations seem to share a common topology (see figure 5.23).

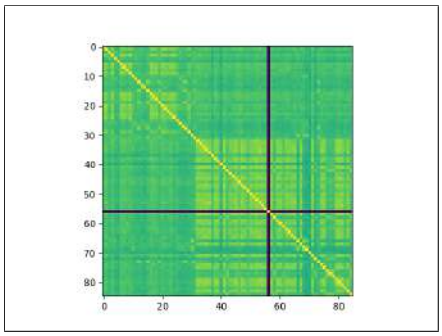

FIGURE 5.22: Results of clustering produced by improving Shortest path kernel, taking into account of VdW interactions (for MOEM3-based graphs).

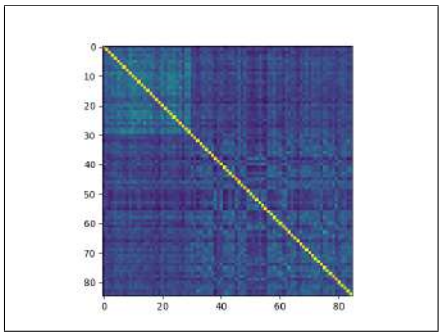

FIGURE 5.23: Results of clustering produced by improving Shortest path kernel, taking into account of ionic interactions (for 6A90-based graphs).

Using the previous selection by links, except for the comparison diagram on the π-cation interaction, all the other bonds identify common patterns among mutations of different biological significance (see figures from 5.24 to 5.27).

- Weisfeiler-Lehman-based algorithm
  An increase of the distinction is obtained by resorting to the Weisfeiler-Lehman graph kernel that since the first iterations appears to clearly show a pattern that accumulates mutations gain of function between them and a different and common pattern for mutations not involving pain syndromes, the same results are also inferred also going to analyze graphs that take only one interaction at a time (see fiugres from 5.28 to 5.34).

  To support the results obtained previously, in the case of 6A90-based graphs, all the comparison diagrams generated (with different iteration scales from 1 to 5, both by dividing the arcs by type of bond) show how the two types of mutation are classified otherwise (see figure from 5.35 to 5.41).

The last two figures (5.42 and 5.43) show how this way of clustering where able to group 100% of the gain-of-function mutations in a common set, for the structures developed starting from the 6A90 template. While, for models based on the MOESM3 template, there is a selectivity of 93.33%. Data however very good, it is possible to justify this minor accuracy as the starting

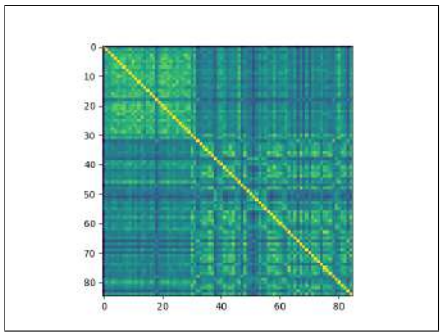

FIGURE 5.24: Results of clustering produced by improving Shortest path kernel, taking into account of h-bond interactions (for 6A90-based graphs).

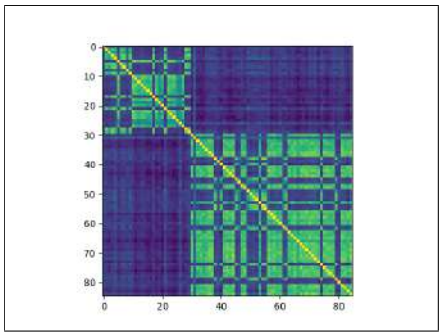

FIGURE 5.25: Results of clustering produced by improving Shortest path kernel, taking into account of ionic interactions (for 6A90-based graphs).

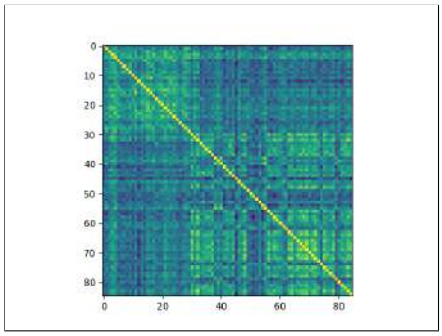

FIGURE 5.26: Results of clustering produced by improving Shortest path kernel, taking into account of π-π stacking interactions (for 6A90-based graphs).

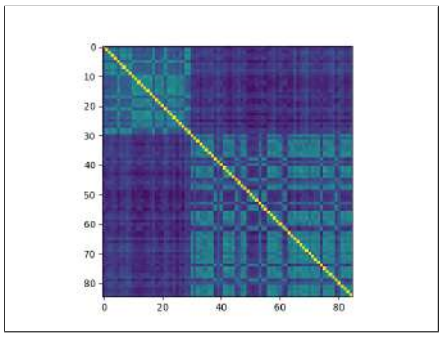

FIGURE 5.27: Results of clustering produced by improving Shortest path kernel, taking into account of VdW interactions (for 6A90-based graphs).

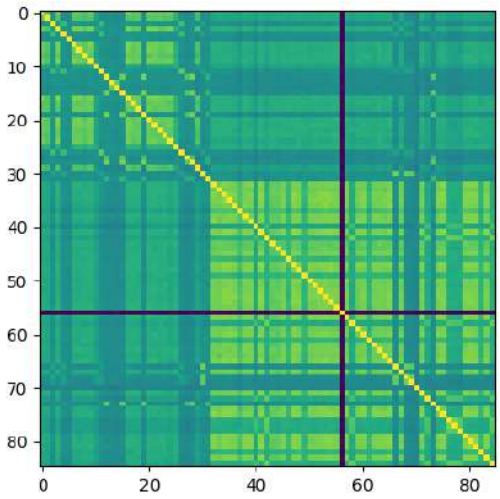

FIGURE 5.28: Results of clustering produced by improving Weisfeiler-Lehman kernel, taking into account of all interactions and after one iteration (for MOESM3-based graphs).

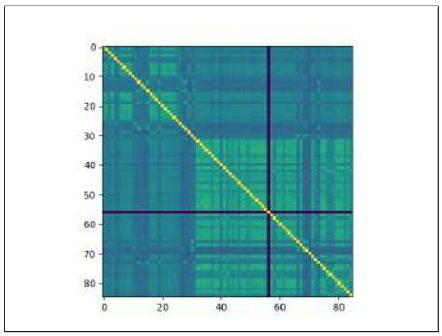

FIGURE 5.29: Results of clustering produced by improving Weisfeiler-Lehman kernel, taking into account of all interactions and after five iteration (for MOESM3-based graphs).

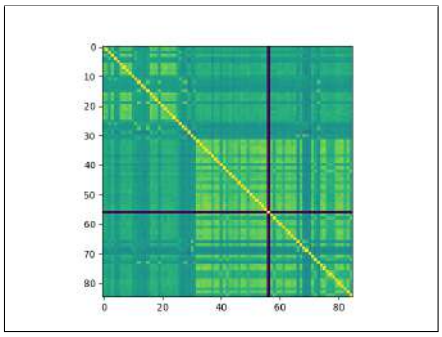

FIGURE 5.30: Results of clustering produced by improving Weisfeiler-Lehman kernel, taking into account of H-bond interactions (for MOESM3-based graphs).

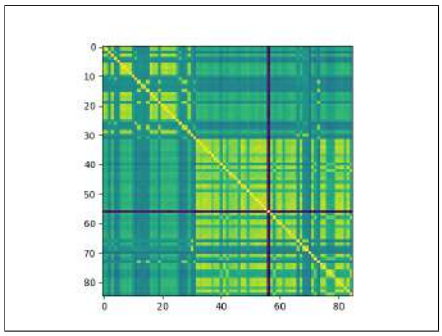

FIGURE 5.31: Results of clustering produced by improving Weisfeiler-Lehman kernel, taking into account of ionic interactions (for MOESM3-based graphs).

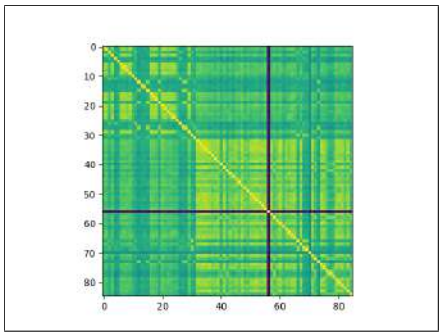

FIGURE 5.32: Results of clustering produced by improving Weisfeiler-Lehman kernel, taking into account of π-π stacking interactions (for MOESM3-based graphs).

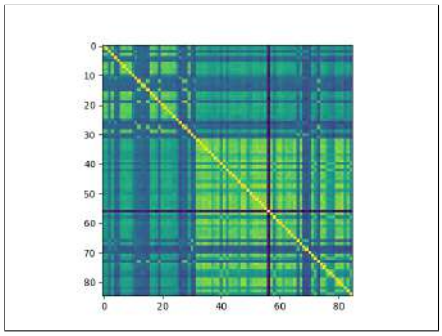

FIGURE 5.33: Results of clustering produced by improving Weisfeiler-Lehman kernel, taking into account of π cation interactions (for MOESM3-based graphs).

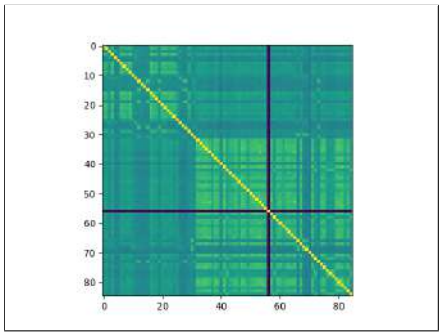

FIGURE 5.34: Results of clustering produced by improving Weisfeiler-Lehman kernel, taking into account of VdW interactions (for MOESM3-based graphs).

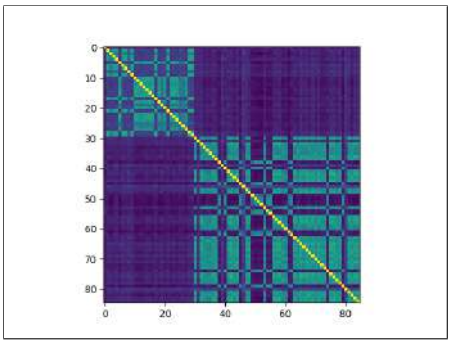

FIGURE 5.35: Results of clustering produced by improving Weisfeiler-Lehman kernel, taking into account of all interactions, after 1 iteration (for 6A90-based graphs).

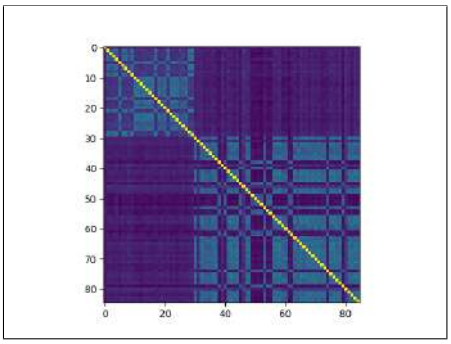

FIGURE 5.36: Results of clustering produced by improving Weisfeiler-Lehman kernel, taking into account of all interactions, after 5 iteration (for 6A90-based graphs).

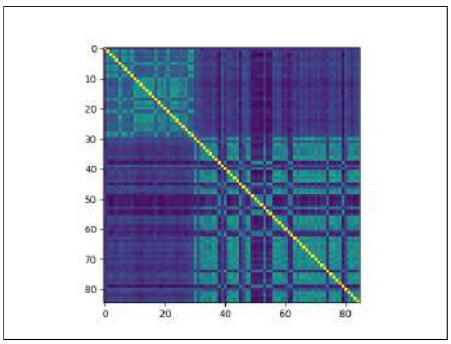

FIGURE 5.37: Results of clustering produced by improving Weisfeiler-Lehman kernel, taking into account of h-bond interactions (for 6A90-based graphs).

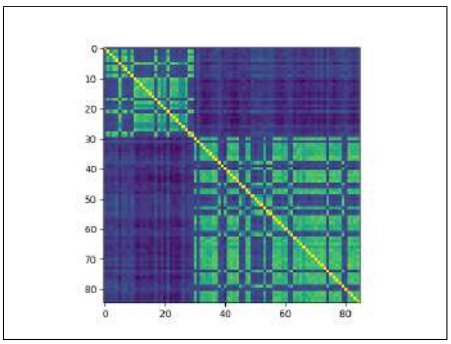

FIGURE 5.38: Results of clustering produced by improving Weisfeiler-Lehman kernel, taking into account of ionic interactions (for 6A90-based graphs).

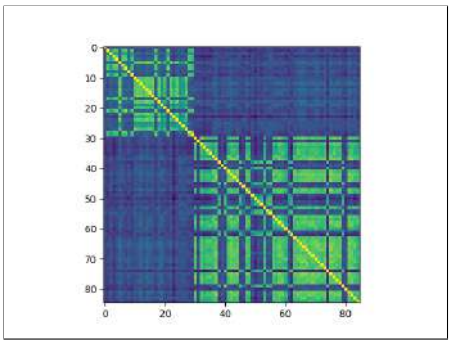

FIGURE 5.39: Results of clustering produced by improving Weisfeiler-Lehman kernel, taking into account of π-cation interactions (for 6A90-based graphs).

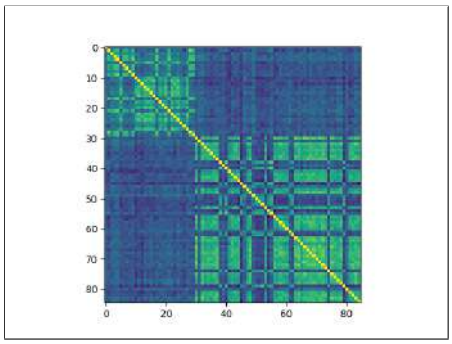

FIGURE 5.40: Results of clustering produced by improving Weisfeiler-Lehman kernel, taking into account of π-π stacking interactions (for 6A90-based graphs).

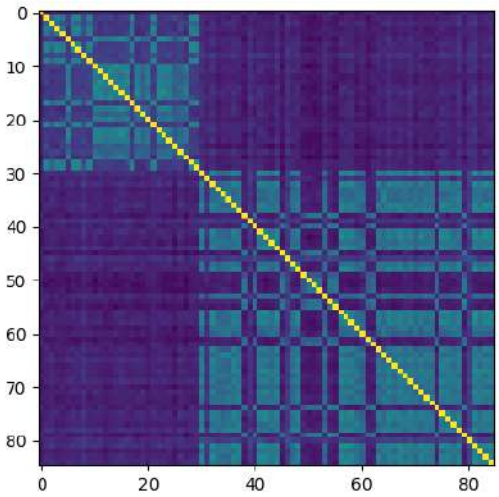

FIGURE 5.41: Results of clustering produced by improving Weisfeiler-Lehman kernel, taking into account of VdW interactions (for 6A90-based graphs).

template was derived from a phylogenetically very distant protein from us, which however led to a proper grouping in for more than 90% of the structures. With regard to structures not directly related to pain disorders, the structures that emerge from non-pathogenic clusters are 23% for 6A90-based structures and 27.3% for MOESM3-based structures. This result can also be very interesting, as these structures have been derived from sequences that seem to be unrelated to pathologies, but these assumptions are not proven with certainty. More recruitment would require more in-depth investigations.

5.6 Surface Analysis

The surface characterization analyzes of the structures were carried out using the UCSF Chimera tools, following these steps in succession:

- **Hydrophobicity**: *Select* → *Structure* → *protein; Actions* → *Surface* → *show* and from *Tools* → *Structure Analysis* → *Render by Attribute*. Here select *Attribute of = residue* and as kind of attribute *kdHydrophobicity* (see figure A.9).
- **Charged surface**: *Select* → *Structure* → *protein; Actions* → *Surface* → *show* and from *Tools* → *Surface/Binding Analysis* → *Coulomb Surface coloring* (see figure A.10).

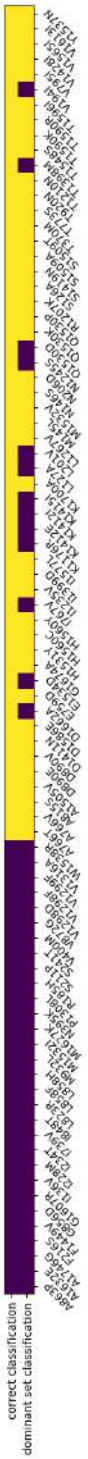

FIGURE 5.42: Results of dominant cluster for 6A90-based structures [10].

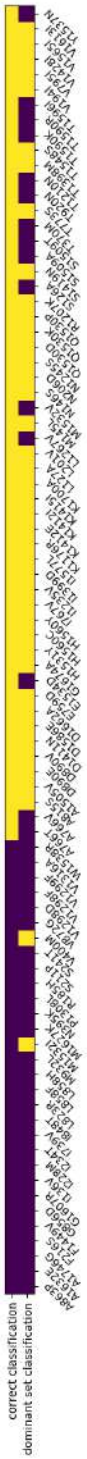

FIGURE 5.43: Results of dominant cluster for MOESM3-based structures [10].

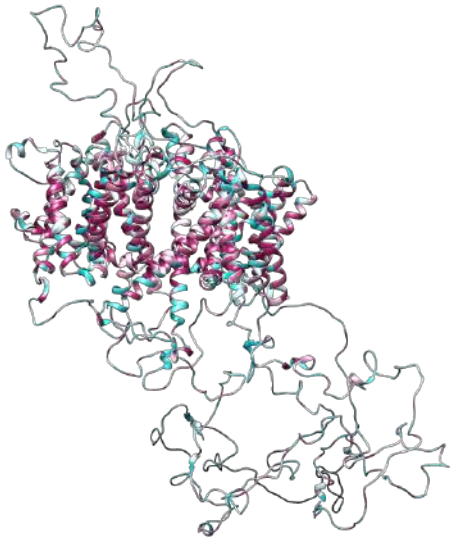

FIGURE 5.44: Wilde Type MOEMS3 colored by its hydrophobicity.

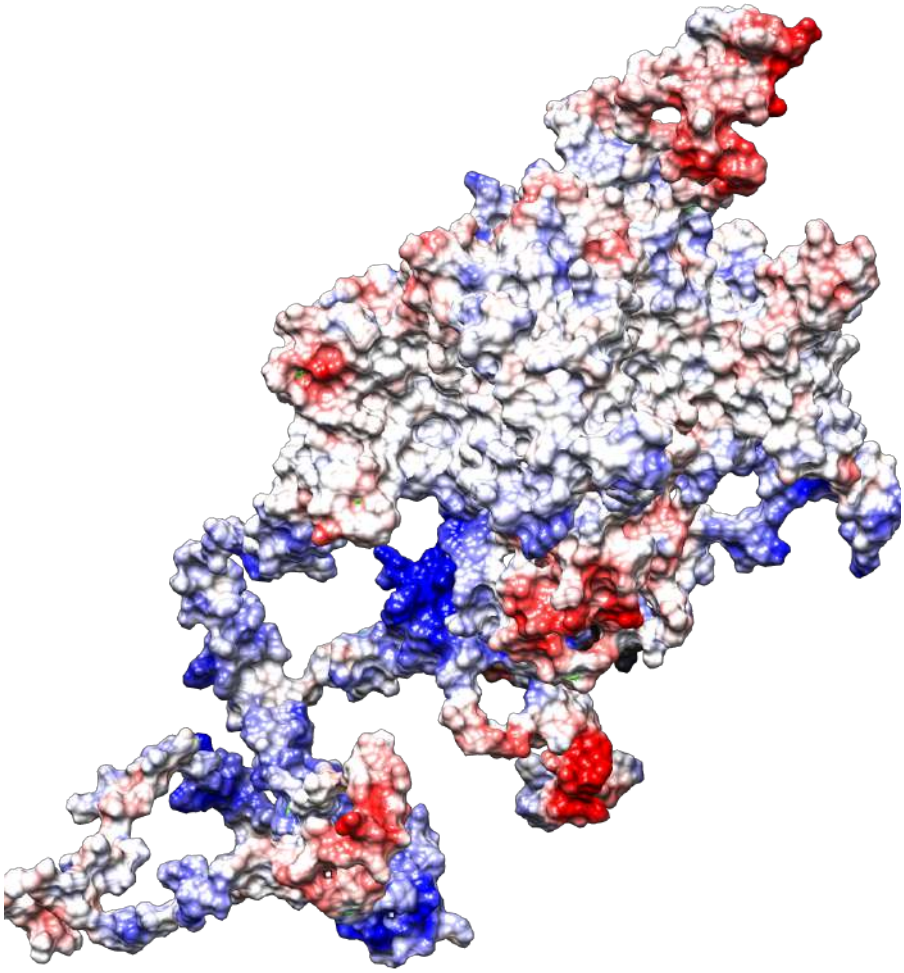

FIGURE 5.45: Wilde Type MOESM3 colored by its Coulomb surface.

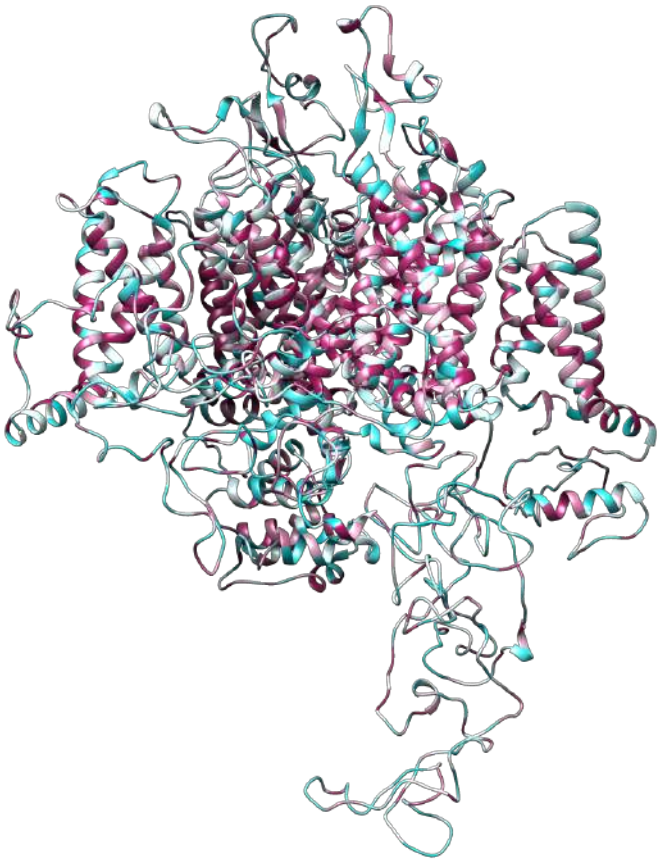

FIGURE 5.46: Wilde Type 6A colored by its hydrophobicity.

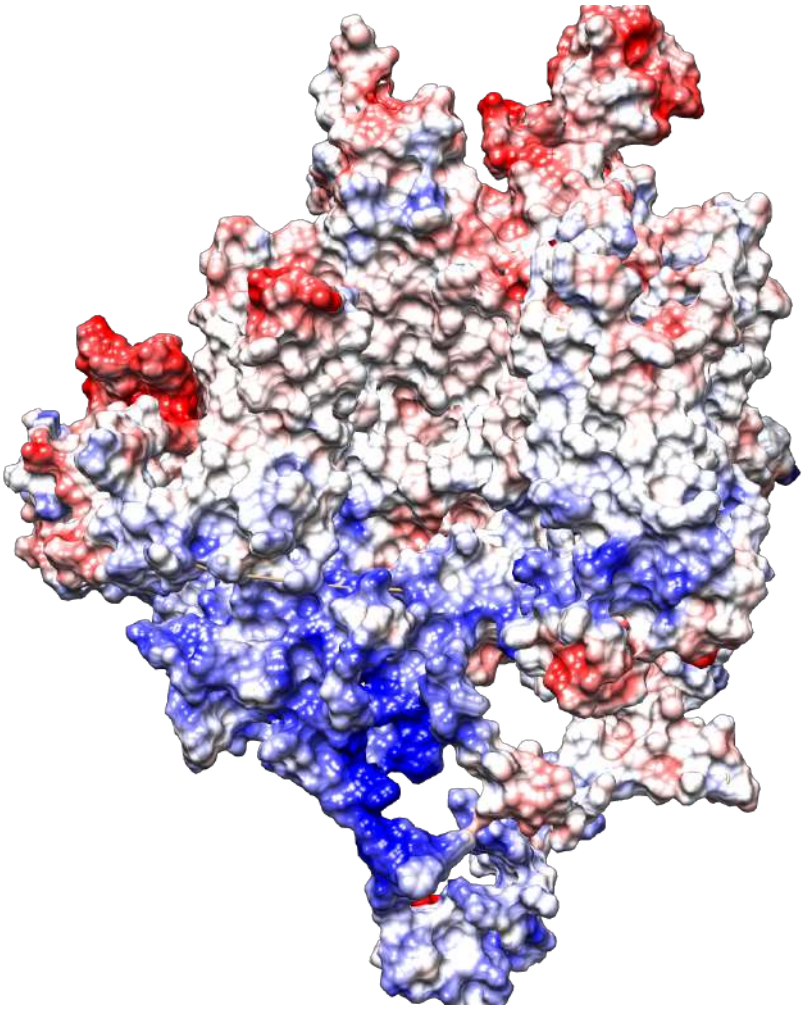

FIGURE 5.47: Wilde Type 6A colored by its Coulomb surface.

Chapter 6

Conclusion

The central problem we addressed in this thesis was to verify whether there is a relationship between point mutations in Nav 1.7 protein sequences and the occurrence of neuropathies in channel functionality. To this aim, we followed a two-fold strategy. On the one hand, we revised the computational pipeline implemented by the Carlo Besta research group [32], that was devised for the same goal. On the other hand, we aimed at improving it both from the methodological viewpoint and from the number of explored case studies. The latter include the MOESM3 already studied in [32], the 6A90, and the 5HVX templates. As regards the methodology, we improved it on the following points. Firstly, we used SWISS-MODEL to replace the MEMOIR tool. This was necessary because of the MEMOIR limitations in terms of the length of the possible analyzed sequences (below 1500 amino acids). Indeed, this is the case of the human sequence encoding NaV1.7, exceeding 1500 peptides. SWISS-Model allows for a complete analysis of the various domains including the connecting loops, rather than an individual analysis of each domain as was done before in [32]. Our approach turns out to be a significant improvement to the generation of high quality models in the areas of interest (sections inside the membrane and in the interface strips), that include all the examined mutations. Furthermore, a good quality Ramachandran Plot with more than 90% of the residues in favorable regions, were obtained for MOESM3-based and 6A90-based structures: both of them exceed this threshold, 95.6% in one case and 97.6% in other. Hence, in addition to obtaining a good set of models starting from the same MOESM3-based template [32], it was also possible to widen the analysis by using two more templates (the 6A90 and the 5HVX) derived from homologous proteins closer to humans. Within this framework, an important result is represented by the significant improvement obtained by using graph theory and machine leaning methods for the topological analysis of the protein networks. The analysis proposed by Dimos et al [32] hinged upon the use of simple centrality metrics, within which the betweenness centrality turns out to be the only one able to discriminating between pathological and non-pathological cases. We explicitly reproduced this result in the same original case study. However, a direct comparison was impossible, as the approach we followed led to the production of RINs having many more nodes (about 600 nodes more) as compared with the original ones [32]. The present study goes beyond this by accounting for a new type of comparison involving the whole RINs rather than single point mutation metrics. This approach allowed to classify in the

same cluster 100% of the gain-of-function mutations in the case of RINs related to the 6A90-based models, and of the 93.33% for the RINs derived from MOESM3-based models. Albeit based on preliminary analysis, these results are rather encouraging and reinforce the results obtained by Dimos et al [32] on the relationship between protein sequence mutations and neuropathies in ionic sodium channels membrane proteins. Further work along the line of the present study will be necessary to have a final confirmation.

We can envisage a number of possible perspectives for the present work, along two main lines. The present work has unveiled the usefulness of studying the topological properties of the network associated with a the three dimensional structure of a protein. Indeed it provides a very fast and effective way to identify common patterns within a large set of available structures. However, we were forced to use many different and unconnected tools to reach that goal. It would be extremely useful to set up a more flexible and user-friendly computational pipeline that builds upon a single thread. Another possible development of the present study, would be to perform a very detailed all-atom calculation of the full membrane protein for the specific mutations that were identified as interesting by the present study.

Following the lines suggested by ([24]), we could use the models generated not only as frames, but also to perform a full dynamical study. An attempt to mimic the real biological conditions computationally undoubtedly would allow a full understanding of the implications of each single mutation. Highlighting how this is involved in every single phase of the biological role of the protein and not only in a snapshot which can only provide an incomplete image, is the final goal for the next decade research in the field.

Appendix A

Appendix A

A.1 Results of Alignments

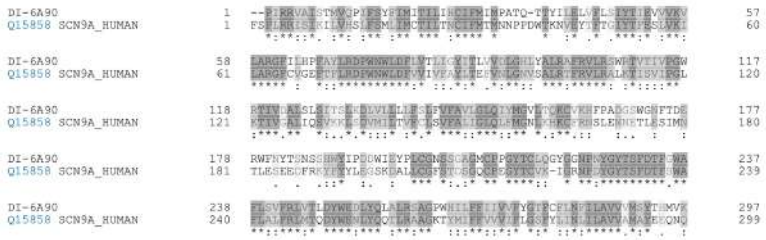

FIGURE A.1: Alignment of DIs of NaVPas and NaV1.7, with and identiy score of 46%.

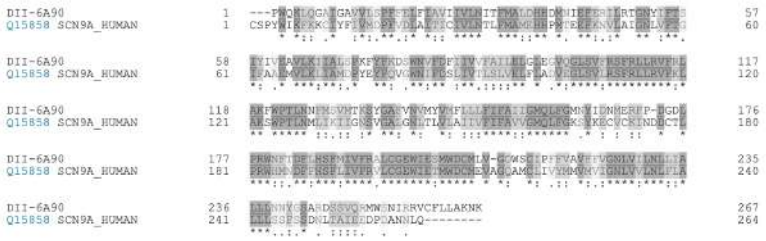

FIGURE A.2: Alignment of DIIs of NaVPas and NaV1.7, with and identiy score of 45.59%.

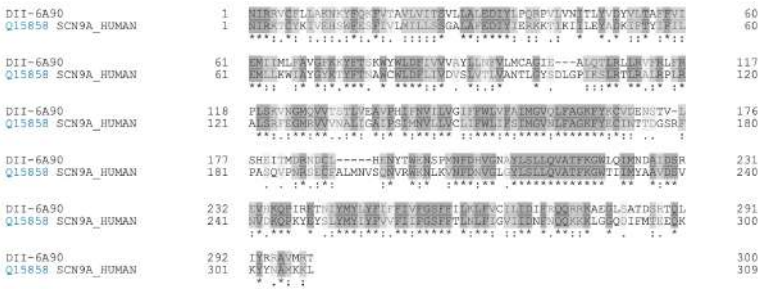

FIGURE A.3: Alignment of DIIIs of NaVPas and NaV1.7, with and identiy score of 41.75%.

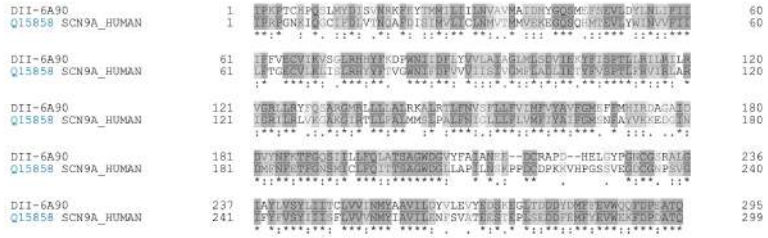

FIGURE A.4: Alignment of DIVs of NaVPas and NaV1.7, with and identiy score of 47.16%.

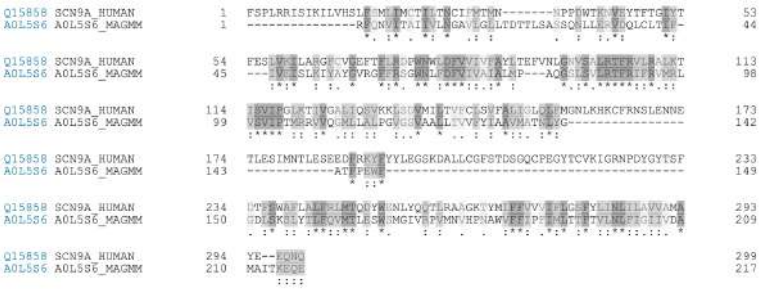

FIGURE A.5: Alignment of DIs of NaVMs and NaV1.7, with and identiy score of 16.23%.

FIGURE A.6: Alignment of DIIs of NaVMs and NaV1.7, with and identiy score of 20.59%.

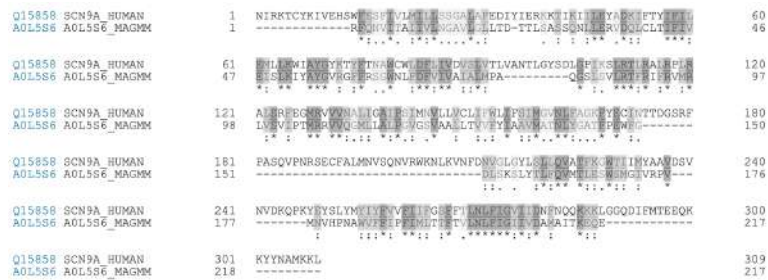

FIGURE A.7: Alignment of DIIIs of NaVMs and NaV1.7, with and identiy score of 20.71%.

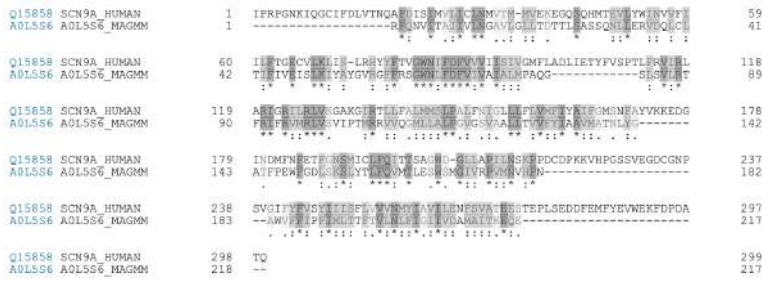

FIGURE A.8: Alignment of DIVs of NaVMs and NaV1.7, with and identiy score of 18.21%.

A.2 TM-Score

TM-score (Template Modeling) is a score-function that measure the similarity between two protein structure. The TM-score indicates the difference between two structures by a score between (0,1], where 1 indicates a perfect match between two structures (thus the higher the better). Generally scores below 0.17 corresponds to randomly chosen unrelated proteins.

$$TMscore = max[\frac{1}{L_{(target)}} \sum_{i}^{L_{(aligned)}} \frac{1}{1 + (\frac{d_i}{d_0(L_{(target)})})}] \tag{A.1}$$

A.3 UCSF Chimera

Chimera is a high performance extensible software for the visualization and analysis of molecular structures, including density maps, supramolecular assemblies, sequence alignments, docking results, trajectories, and conformational ensembles. Chimera's primary programming language is Python, the choice fell on phyton as it is an easy to understand programming language enabling others to develop extensions without undue effort. This software was used to check the results obtained from homology and energy minimization steps, and it has been involved in the production of structures for evaluation with Ramachandran plots (production of the .pdb files that took into account only the features characterized by a secondary structure with α-helix). Moreover, it is with chimera that the images of the colored structures have been realized based on the chemical-physical properties of the residues (hydrophobicity in colored scale cyan-maroon and charged surface in scale from blue to red) (see figures A.9 and A.10). [51].

A.4 DSSP method

DSSP (*Define Secondary Structure of Proteins*)is a standard method for assigning secondary structure to the amino acids of a protein, given its atomic coordinates. This method identifies the intra-backbone hydrogen bonds of the

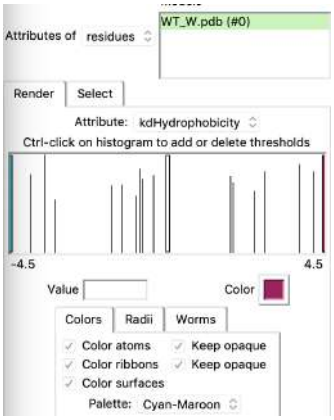

FIGURE A.9: Command shall of Chimera to highlight hydrophobicity feature of structures.

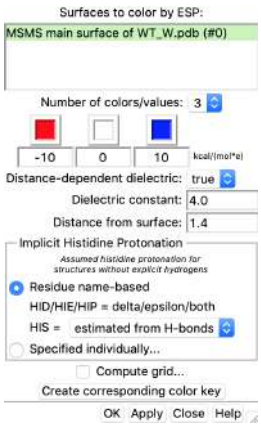

FIGURE A.10: Command shall of Chimera to highlight charged feature of structures.

protein using a purely electrostatic definition, assuming partial charges of -0.42 e and +0.20 e to the carbonyl oxygen and amide hydrogen respectively, their opposites assigned to the carbonyl carbon and amide nitrogen. A hydrogen bond is identified if E in the following equation is less than -0.5 kcal/mol [70].

A.5 Ramachandran Plots

Ramachandran plots is a tool for evaluating the quality of structures, taken its name from Gopalasamudram Narayana Ramachandran and his collaborators, who in 1963 came up with the idea. The reasoning that led to the development of this tool starts from some intelligent assumptions. First, there are 4 covalent bonds that make up the backbone of a protein, one of them is the carbonylic double bond C=O. Which is of little relevance, as rotations around its axis are extremely unfavorable and in any case would not affect the shape of the backbone. Although to a lesser extent, the bond between the carbonyl C of a residue and the amidic N of the subsequent residue also has the character of a double bond, implying that there are only two possible angles that stabilize the structure: 0°(*cis*) and 180°(*trans*). Thus the analysis of the possible configurations is reduced to the study of the other two dihedral angles:

- The N-C_α rotation, identified by the dihedral angle C_{i-1}-N-C_α-C, which is named φ.
- And the C_α-C rotation, identified by the dihedral angle N-C_α-C-N_{i+1}, which is named ψ.

Around these angles the rotations are easier, but not all are allowed, strong obstructions are due to the clashes between the atoms during the rotations. Using a hard-sphere atomic model based on quantum-mechanics principles, atomic co-penetrations are impossible, i.e. *'forbidden'*. G. N. Ramachandran and coworkers have put in place a protein model to test the energetic landscape according to the angles. This model was based on the compound N-acetyl-l-alanine-methylamide (see figure A.11).

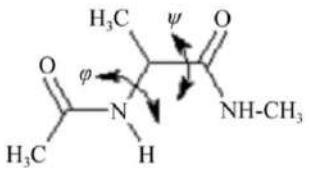

FIGURE A.11: The model compound N-acetyl-l-alanine-methylamide used by Ramachandran and coworkers to explore the conformational space defined by the two dihedral angles [12].

All possible combination of φ and ψ were computed and for each it was verified if interaction clashes occurred [12]. The results of the possible favorable combinations have been reported in a graph (see figure 5.12).

A.6 Weisfeiler-Lehman Kernel script

Script written by Giacomo Chiarot.

0.5

```
 ===================================
||  Classification on the PROTEINS  ||
||          Kernel generator        ||
||  -------------------------------  ||
||  Giacomo Chiarot                 ||
||  giacomochiarot@gmail.com        ||
 ===================================
"""
from __future__ import print_function

print(__doc__)

from sklearn.model_selection import cross_val_score
from grakel import GraphKernel
from grakel import Graph
from sklearn import svm
import numpy as np
import matplotlib.pyplot as plt
from scipy.cluster.hierarchy import dendrogram, linkage


"""
Reads the list of files and returns the name of the files and their value which represent the class of each protein
"""
def readProteins():
    print("-- reading file list")
    f = open("trainingSet/fileList.txt", "r")
    proteinNames = []
    labels = []
    for line in f:
        lineDivided = line.split(' ')
        proteinNames.append(lineDivided[0])
        labels.append(int(lineDivided[1]))
    f.close()
    return proteinNames, labels

"""
Reads the list of arcs for each protein and stores them as graphs
"""
def readGraphs(proteinNames):
    print("-- reading graphs")
    graphs = []
    for name in proteinNames:
        proteinFile = open("trainingSet/" + name + "_adj_ALL.csv", "r")
        labelFile = open("trainingSet/" + name + "_nl.csv", "r")
        graph = {}
        labels = {}
        first = True
        for line in proteinFile:
            if first:
                first = False
            else:
                values = line.split(';')
                edges = (values[0], values[1])
                graph[edges] = float(values[2])
        first = True
        for line in labelFile:
            if first:
                first = False
            else:
                values = line.split(';')
                labels[values[0]] = int(values[1])
        proteinFile.close()
        labelFile.close()
        graphs.append(Graph(graph, labels))
    return graphs

"""
Computes the weisfeiler_lehman kernel
"""
def computeKernel(graphs):
    print("-- computing kernel")
    wl_kernel = GraphKernel(kernel=[{"name": "weisfeiler_lehman", "niter": 5}, {"name": "subtree_wl"}], normalize=True)
    return wl_kernel.fit_transform(graphs)

"""
```

```
Computes 10-times cross validation with svm and returns the mean of results
"""
def runSVM(K, labels):
    print("-- computing scores with SVM")
    mod = svm.SVC(kernel='precomputed')
    scores = cross_val_score(mod, K, labels, cv=10)
    return np.mean(scores)

def main():
    proteinNames, labels = readProteins()
    graphs = readGraphs(proteinNames)
    K = computeKernel(graphs)
    np.savetxt("6A90manual-kernelWL-ALL5.txt", np.array(K), fmt="%s")
    np.savetxt("labels.txt", np.array(labels), fmt="%s")
    result = runSVM(K, labels)
    print("Accuracy is: " + str(result))
    print("Plot of the similarity matrix saved in file")
    fig1 = plt.figure()
    plt.imshow(K)
    #plt.show()
    fig1.savefig("6A90manual-plotMatrixWL_ALL5.png", dpi = fig1.dpi)
    plt.close(fig1)
"""
    Kdist = 1.0 - K
    print(Kdist)
    c = linkage(Kdist,"complete")
    fig2 = plt.figure()
    d = dendrogram(c)
    #plt.tight_layout()
    plt.savefig("clustering_complete.png", dpi=fig2.dpi)
    plt.close(fig2)
"""
if __name__ == "__main__":
    main()
```

A.7 Dominant-set clustering script

Script written by Giacomo Chiarot.

```
    0.5
=====================================
~~~~~~~~~~PROTEIN CLUSTERING~~~~~~~~~~
                whit
            Dominant set
=====================================

Giacomo Chiarot giacomochiarot@unive.it

"""
print(__doc__)

from sklearn.metrics import pairwise_distances
from scipy.cluster.hierarchy import linkage
from sklearn.datasets import load_digits
from sklearn.manifold import Isomap
from past.builtins import execfile
import matplotlib.pyplot as plt
from numpy.linalg import norm
import numpy as np
import random

execfile('library.py')

filesList = ['6A90manual-kernelWL-ALL5', '6A90strict-kernelWL-ALL5', 'MOESM3manual-kernelWL-ALL5', 'MOESM3strict-kernelWL-ALL5'
result = [[], [0 for i in range(85)]]
result[0] += [0 for i in range(30)]
result[0] += [1 for i in range(30, 85)]
labelx = ['A863P', 'A1632E', 'A1746G', 'F216S', 'F1449V', 'G856D', 'G1607R', 'I136V', 'I228M', 'I234T', 'I739V', 'I848T', 'L823
labely = ['correct classification', 'dominant set classification']

for file in filesList:
    result[1] = [0 for i in range(85)]
    print(file)

    # -- READING DATA --
    print("-- Loading similarity matrix --")
    S = np.loadtxt('data/' + file + ".txt")

    # -- COMPUTING DOMINANT SET --
    print("-- Compoting dominant-set --")
    x = dominant_set(S, epsilon=2e-6)

    # -- EXTRACTING ONE CLUSTER --
    print("-- Extracting one cluster --")
    cutoff = np.median(x)
    cluster = np.where(x > cutoff)
```

```
    for i in cluster[0]:
        result[1][i] = 1
    fig1 = plt.figure()
    fig1, ax = plt.subplots(figsize=(18,5))
    plt.imshow(result)
    ax.set_xticks(np.arange(len(labelx)))
    ax.set_yticks(np.arange(len(labely)))
    ax.set_xticklabels(labelx)
    ax.set_yticklabels(labely)

    # Rotate the tick labels and set their alignment.
    plt.setp(ax.get_xticklabels(), rotation=45, ha="right", rotation_mode="anchor")

    fig1.tight_layout()
    fig1.savefig(file + ".png", dpi = fig1.dpi)
    plt.plot()
    plt.close(fig1)
```

A.8 RIN Parser

0.5

```
"""
    ====================================
    ||            RIN parser            ||
    ||  ________________________________  ||
    ||  Fabio Rosada                     ||
    ||  Davide Crosariol                 ||
    ====================================
    """


from bs4 import BeautifulSoup as Soup
import numpy as np
import time
import random
import re
import csv

# PUT THE RIGHT PATH FOR YOUR DIRECTORIES
OUTPUT_DIRECTORY = "output/"
CACHE_DIRECTORY = "cache/"


# CONSTANTS
WEIGHT_ENERGY = "e_Energy"
WEIGHT_DISTANCE = "e_Distance"
NODE_POSITION = "v_Position"
REALLY_HIGH_NUMBER = 100000.0


class GraphMatrix:

    # If you don't pass any parameter it'll load the default file included in this repo
    def __init__(self, file):
        if not file:
            print("You need to give at least one xml file in input")
            return

        start_time = time.time()

        print("Loadind matrix from file:", file)

        xml = open(file, "r").read()
        xml = Soup(xml, 'lxml')

        self.interactions = dict()

        edges = xml.find_all("edge")
        floor = 0
        for edge in edges:
            tmp = interaction_to_key(edge.find(key="e_Interaction").get_text())
            if tmp not in self.interactions:
                self.interactions[tmp] = floor
                floor += 1

        print(self.interactions)

        self.file_name = file
        self.n_nodes = len(xml.find_all("node"))
        self.nodes = []
        self.nodesPosition = []
        # metto zeri in tutta la matrice
        self.matrix = np.zeros((self.n_nodes, self.n_nodes, len(self.interactions)))
        self.edges = []

        # initialize matrix with a really big int for floyd-warshall
```

```
        # la diagonale rimane tutta a zero
        self.initialize_matrix()

        nodes = xml.find_all("node")
        for node in nodes:
            self.nodes.append(node.find(key="v_NodeId").get_text())
            self.nodesPosition.append(node.find(key=NODE_POSITION).get_text())

        self.edges.append(len(edges))
        for edge in edges:
            # print(edge['source'], edge['target'])
            src = int(edge['source'][1:])
            trg = int(edge['target'][1:])
            tmp_inter = edge.find(key="e_Interaction").get_text()
            interaction = self.interactions[interaction_to_key(tmp_inter)]

            # Choose between DISTANCE or ENERGY
            weight = float(edge.find(key=WEIGHT_DISTANCE).get_text())
            self.matrix[src, trg, interaction] = weight
            self.matrix[trg, src, interaction] = weight

        self.print_info()
        print("************************************\nTotal loading time: ", time.time() - start_time, "\n")


    def get_dimen(self):
        return self.n_nodes

    def get_node(self, n):
        return self.nodes[n][-3:]

    def get_node_info(self, n):
        return self.nodes[n].split(':')


    def name(self):
        r = re.compile('/([^/]+)_network\.xml')
        return r.search(self.file_name)[1]


    def print_info(self):
        n_edges = 0
        for edge in self.edges:
            n_edges += edge
        print("# Nodes:\t", self.n_nodes, "\n# Edges:\t", n_edges)

    def get_interaction_number(self):
        return len(self.interactions)

    def get_interaction_id(self, interaction: str):
        if "ALL" in interaction:
            return -1
        try:
            return self.interactions[interaction]
        except KeyError:
            print("This interaction is not present in this protein")
            return -2

    # inizializza la matrice con numeri altissimi
    def initialize_matrix(self):
        for row in range(self.n_nodes):
            for col in range(self.n_nodes):
                if row != col:
                    for f in range(self.get_interaction_number()):
                        self.matrix[row, col, f] = REALLY_HIGH_NUMBER


    # stampa il file con la lista delle adiacenze: considera tutte le interazioni
    # e scrive poi la distanza minima
    def print_adj(self):
        file = open(OUTPUT_DIRECTORY + str(self.name()) + "_adj_ALL" + ".csv", "w")
        writer = csv.writer(file, delimiter=";", lineterminator='\n')
        writer.writerow(('source', 'destination', 'distance'))


        for c in range(self.get_dimen()):
            for r in range(0, c):
                min = 100000
                for el in self.matrix[r, c]:
                    if el < min:
                        min = el
                if min < 100000 and min != 0:
                    writer.writerow((self.nodes[r], self.nodes[c], min))

    # stampa un file con la lista delle adiacenze, per ogni specifica interazione
    def print_adj_interactions(self):

        for el in self.interactions:
            file = open(OUTPUT_DIRECTORY + str(self.name()) + "_adj_" + el + ".csv", "w")
            writer = csv.writer(file, delimiter=";", lineterminator='\n')
            writer.writerow(('source', 'destination', 'distance'))
```

```
            k = self.interactions[el]

            for c in range(self.get_dimen()):
                for r in range(0, c):
                    value = self.matrix[r,c,k]

                    if value < 100000 and value != 0:
                       writer.writerow((self.nodes[r], self.nodes[c], value))


    # stampa il file con la lista dei nodi e le loro etichette (posizioni nella sequenza primaria)
    def print_nl(self):
        file = open(OUTPUT_DIRECTORY + str(self.name()) + "_nl" + ".csv", "w")
        writer = csv.writer(file, delimiter=";", lineterminator='\n')
        writer.writerow(('node', 'label'))
        i=0
        while i< self.n_nodes:
            writer.writerow((self.nodes[i],self.nodesPosition[i]))
            i = i+1


def interaction_to_key(name: str):
    r = re.compile('(.*):.*')
    x = r.search(name.upper())
    if x is None:
        return name.upper()
    else:
        return x[1]
```

0.5

```
    ====================================
    ||            RIN parser            ||
    ||  ------------------------------  ||
    ||  Fabio Rosada                    ||
    ||  Davide Crosariol                ||
    ====================================
    """

import sys

from src.graphs_parser import GraphMatrix
#from src.graph_measures import GraphMeasures as measures
#from src.graph_measures import *

    ############################
    # VARIABLE INITIALIZATION #
    ############################

# FILL THESE FIELDS BEFORE EXECUTION (Or pass them as args from terminal)
FILE_NAME = "assets/3rvy_van0_network.xml" # default test file


# TERMINAL ARGUMENTS (only file name)
if len(sys.argv) == 2:
    FILE_NAME = sys.argv[1]



    ########
    # MAIN #
    ########

x = GraphMatrix(FILE_NAME)  # load the graph

x.print_adj()   # Print adjacency matrix for ALL interactions on output folder
x.print_adj_interactions(); # Print adjacency matrix for each single interaction on output folder
x.print_nl()    # Print list of nodes with their labels
```

A.9 FG-MD setting parameters details

Explanation of the meaning of command lines:

- Units **real**, for this style unites are:
  - mass = grams/mole
  - distance = Angstroms
  - time = femtoseconds

  - energy = Kcal/mole
  - velocity = Angstroms/femtosecond
  - force = Kcal/mole-Angstrom
  - torque = Kcal/mole
  - temperature = Kelvin
  - pressure = atmospheres
  - dynamic viscosity = Poise
  - charge = multiple of electron charge (1.0 is a proton)
  - dipole = charge x Angstroms
  - electric field = volts/Angstrom
  - density = $gram/cm^{dim}$

- neigh_modify **every** x, this command sets parameters that adjust the building and use of pairwise neighbor lists. The every setting means build lists every M steps (after the delay has passed, that means never build new lists until at least N steps after the previous build).

- atom_style **full** define which style of atoms improve during simulation. *full* use the attributes *molecular + charge*, which is particularly suitable for the study of bio-molecules.

- bond_style **harmonic**, sets which formula(s) LAMMPS has to use to compute bond interactions between pairs of atoms. *harmonic* treats the interactions between atoms describing them with harmonic functions.

- angle_style **harmonic**, as in the previous case, with reference to the calculation of angles.

- dihedral_style **hybrid harmonic**, define multiple styles to describing dihedral angles,including harmonic.

- pair_style **lj/cut/coul/cut**, this command sets the formula(s) used to compute pairwaise interactions. Pair potentials are defined between pairs of atoms that are within a cutoff distance and the set of active interactions typically changes over time. In this case it has been set 10 Åas cutoff for Lennard-Jones and Coulomb interactions.

- pair_modify **mix arthimetic**, goes to modify the parameters defined with the previous command:
  epsilon_ij = sqrt(epsilon_i * epsilon_j)
  sigma_ij = (sigma_i + sigma_j)/2

- boundary **p p p**, the style *p* means the box is periodic, so that particles interact across the boundary, and they can exit one end of the box and re-enter the other end. *p* is replicated three times for the directions of space, it is valid for both the lower and upper face of the box.

- special_bond **amber**, set weighting coefficients for pairwise energy and force contributions between pairs of atoms that are also permanently bonded to each other, either directly or via one or two intermediate bonds. For this type of pairs of atoms, the calculation of the LJ and Coulomb interactions either does not make sense to be calculated or their weight should be reduced. *amber* sets the 3 coefficent of Lennard-Jones potential to 0.0, 0.0 and 0.5, while fro Coulomb potential to 0.0, 0.0 and 0.8333.

- thermo **N**, compute and print thermodynamic info (e.g. temperature, energy, pressure) on timesteps that are a multiple of N and at the beginning and end of a simulation.

- thermo **multi**, style *multi* prints a multiple-line listing of thermodynamic info that is the equivalent of *"thermo_style custom etotal ke temp pe ebond eangle edihed eimp evdwl ecoul elong press"*.

- timestep **2.0**, set the timestep size for subsequent molecular dynamics simulations, based on the unit of measurement chosen in the *units* command. In this case are femtoseconds.

- minimize **etol ftol maxiter maxeval**, sets energy minimization paratemters. MD simulation continue to iterate until one of the stopping criteria is satisfied.

  - *etol*: stopping tolerance for energy (unitless), it was set to 10e-3 Kcal/mole.
  - *ftol*: stopping tolerance for force (force units), it was set to 10e-6 Kcal/mole*Angstrom.
  - *maxiter*: max iterations of minimizer, it was set to 100.
  - *maxeval*: max number of force/energy evaluations, it was set to 1000.

- run **10000**, simply sets the total number of iterations to do.

Bibliography

[1] Energy minimization - GROMACS 2019.3 documentation. Available at http://manual.gromacs.org/current/reference-manual/algorithms/energy-minimization.html.

[2] What is an action potential? Available at https://www.moleculardevices.com/applications/patch-clamp-electrophysiology/what-action-potential#gref.

[3] What is fasta format? Available at https://zhanglab.ccmb.med.umich.edu/FASTA/.

[4] Spinal cord anatomy - parts and spinal cord functions. Available at https://healthjade.com/spinal-cord/, Dec 2017.

[5] Regina Bailey. Learn about the peripheral nervous system and why it's important. Available at https://www.thoughtco.com/nervous-system-373574, Jun 2019.

[6] Dennis A Benson, Ilene Karsch-Mizrachi, David J Lipman, and James Ostell. and david l. wheeler. *Nucleic Acids Research*, 31(1):23–27, 2003.

[7] Bryan J Black, Rahul Atmaramani, Sarah Plagens, Zachary T Campbell, Gregory Dussor, Theodore J Price, and Joseph J Pancrazio. Emerging neurotechnology for antinoceptive mechanisms and therapeutics discovery. *Biosensors and Bioelectronics*, 2018.

[8] Iulia Blesneac, Andreas C Themistocleous, Carl Fratter, Linus J Conrad, Juan D Ramirez, James J Cox, Solomon Tesfaye, Pallai R Shillo, Andrew SC Rice, Stephen J Tucker, et al. Rare nav1. 7 variants associated with painful diabetic peripheral neuropathy. *Pain*, 159(3):469, 2018.

[9] K.m. Borgwardt and H. Kriegel. Shortest-path kernels on graphs. *Fifth IEEE International Conference on Data Mining (ICDM05)*.

[10] Samuel Rota Bulò and Marcello Pelillo. Dominant-set clustering: A review. *European Journal of Operational Research*, 262(1):1–13, 2017.

[11] Carlo Camilloni, Daniela Bonetti, Angela Morrone, Rajanish Giri, Christopher M Dobson, Maurizio Brunori, Stefano Gianni, and Michele Vendruscolo. Towards a structural biology of the hydrophobic effect in protein folding. *Scientific reports*, 6:28285, 2016.

[12] Oliviero Carugo and Kristina Djinovic-Carugo. Half a century of ramachandran plots. *Acta Crystallographica Section D Biological Crystallography*, 69(8):1333–1341, 2013.

[13] Augustin Cauchy. Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris*, 25(1847):536–538, 1847.

[14] Wonseok Chang, Temugin Berta, Yong Ho Kim, Sanghoon Lee, Seok-Yong Lee, and Ru-Rong Ji. Expression and role of voltage-gated sodium channels in human dorsal root ganglion neurons with special focus on nav1. 7, species differences, and regulation by paclitaxel. *Neuroscience bulletin*, 34(1):4–12, 2018.

[15] The UniProt Consortium. *UniProt: a worldwide hub of protein knowledge*, volume 47. 2019.

[16] Biology Dictionary. Peripheral nervous system definition. Available at https://biologydictionary.net/peripheral-nervous-system, 2018. Last accessed December 11, 2018.

[17] Ken A Dill and Justin L MacCallum. The protein-folding problem, 50 years on. *science*, 338(6110):1042–1046, 2012.

[18] Christopher M. Dobson. *Protein folding and misfolding*, volume 426. 2003.

[19] Nadezhda T Doncheva, Karsten Klein, Francisco S Domingues, and Mario Albrecht. Analyzing and visualizing residue networks of protein structures. *Trends in biochemical sciences*, 36(4):179–182, 2011.

[20] Jean-Paul Ebejer, Jamie R. Hill, Sebastian Kelm, Jiye Shi, and Charlotte M. Deane. Memoir: template-based structure prediction for membrane proteins. *Nucleic Acids Research*, 2013.

[21] David Eisenberg. *The discovery of the α-helix and β-sheet, the principal structural features of proteins*, volume 100. 2003.

[22] Edward C Emery, Ana Paula Luiz, and John N Wood. Nav1. 7 and other voltage-gated sodium channels as drug targets for pain relief. *Expert opinion on therapeutic targets*, 20(8):975–983, 2016.

[23] Mark Estacion, T Patrick Harty, Jin-Sung Choi, Lynda Tyrrell, Sulayman D Dib-Hajj, and Stephen G Waxman. A sodium channel gene scn9a polymorphism that increases nociceptor excitability. *Annals of Neurology: Official Journal of the American Neurological Association and the Child Neurology Society*, 66(6):862–866, 2009.

[24] Tianhua Feng, Subha Kalyaanamoorthy, Aravindhan Ganesan, and Khaled Barakat. Atomistic modeling and molecular dynamics analysis of human cav1.2 channel using external electric field and ion pulling simulations. *Biochimica et Biophysica Acta (BBA) - General Subjects*, 1863(6):1116 – 1126, 2019.

[25] Richard J Lewis Fernanda C Cardoso. *Sodium channels and pain: from toxins to therapies*, volume 175. BJP, 2018.

[26] Marco Biasini Gabriel Studer and Torsten Schwede. *Assessing the local structural quality of transmembrane protein models using statistical potentials (QMEANBrane)*, volume 30. 2014.

[27] Lewis Y Geer, Aron Marchler-Bauer, Renata C Geer, Lianyi Han, Jane He, Siqian He, Chunlei Liu, Wenyao Shi, and Stephen H Bryant. The ncbi biosystems database. *Nucleic acids research*, 38(suppl_1):D492–D496, 2009.

[28] Abdella M Habib, John N Wood, and James J Cox. Sodium channels and pain. In *Pain Control*, pages 39–56. Springer, 2015.

[29] Yungok Ihm. *A threading approach to protein structure prediction: studies on TNF-like molecules, Rev proteins, and protein kinases*. 2004.

[30] Yu Liang Jian Zhang and Yang Zhang. *Atomic-Level Protein Structure Refinement Using Fragment-Guided Molecular Dynamics Conformation Sampling*, volume 19. 2011.

[31] Sunhwan Jo, Taehoon Kim, Vidyashankara G. Iyer, and Wonpil Im. Charmm-gui: A web-based graphical user interface for charmm. *Journal of Computational Chemistry*, 29(11):1859–1865, 2008.

[32] Dimos Kapetis, Jenny Sassone, Yang Yang, Barbara Galbardi, Markos N Xenakis, Ronald L Westra, Radek Szklarczyk, Patrick Lindsey, Catharina G Faber, Monique Gerrits, et al. Network topology of nav1. 7 mutations in sodium channel-related painful disorders. *BMC systems biology*, 11(1):28, 2017.

[33] Dukka B Kc. Recent advances in sequence-based protein structure prediction. *Briefings in bioinformatics*, 18(6):1021–1032, 2016.

[34] Oaklander Klein. *Ion channels and neuropathic pain*. 2018.

[35] Nils M. Kriege, Fredrik D. Johansson, and Christopher Morris. A survey on graph kernels. *CoRR*, abs/1903.11835, 2019.

[36] Elmar Krieger and Gert Vriend. New ways to boost molecular dynamics simulations. *Journal of Computational Chemistry*, 36(13):996–1007, 2015.

[37] Anna Krylov, Theresa L Windus, Taylor Barnes, Eliseo Marin-Rimoldi, Jessica A Nash, Benjamin Pritchard, Daniel GA Smith, Doaa Altarawy, Paul Saxe, Cecilia Clementi, et al. Perspective: Computational chemistry software and its advancement as illustrated through three grand challenge cases for molecular science. *The Journal of chemical physics*, 149(18):180901, 2018.

[38] Richard H Lee. Protein model building using structural homology. *Nature*, 356:543–544, 1992.

[39] Simon C. Lovell, Ian W. Davis, W. Bryan Arendall III, Paul I. W. de Bakker, J. Michael Word, Michael G. Prisant, Jane S. Richardson, and

David C. Richardson. Structure validation by cα geometry: ϕ, ψ and cβ deviation. *Proteins: Structure, Function, and Bioinformatics*, 50(3):437–450.

[40] G.P.S.Raghava Manoj Bhasin. *Computational Methods in Genome Research*, volume 6. 2006.

[41] Ron Milo and Ron Philips. About us. Available at http://book.bionumbers.org/about-us/.

[42] John Moult, Krzysztof Fidelis, Andriy Kryshtafovych, Torsten Schwede, and Anna Tramontano. Critical assessment of methods of protein structure prediction (casp) - round x. *Proteins: Structure, Function, and Bioinformatics*, 82:1–6, 2013.

[43] Maria Musgaard, Teresa Paramo, Laura Domicevica, Ole Juul Andersen, and Philip C Biggin. Insights into channel dysfunction from modelling and molecular dynamics simulations. *Neuropharmacology*, 132:20–30, 2018.

[44] NCBI. Blast. Available at https://blast.ncbi.nlm.nih.gov/Blast.cgi.

[45] NPTEL. Available at https://nptel.ac.in/courses/104102016/15.

[46] Swiss Institute of Bioinformatics. Available at https://swissmodel.expasy.org/interactive#structure.

[47] Swiss Institute of Bioinformatics. Available at https://swissmodel.expasy.org/qmean/.

[48] National Institute of Health.

[49] Marco Biasini Pascal Benkert and Torsten Schwede. *Toward the estimation of the absolute quality of individual protein structure models*, volume 27. 2011.

[50] Silvio C. E. Tosatto Pascal Benkert and Dietmar Schomburg. *QMEAN: A comprehensive scoring function for model quality assessment*, volume 71. 2008.

[51] Eric F. Pettersen, Thomas D. Goddard, Conrad C. Huang, Gregory S. Couch, Daniel M. Greenblatt, Elaine C. Meng, and Thomas E. Ferrin. Ucsf chimera?a visualization system for exploratory research and analysis. *Journal of Computational Chemistry*, 25(13):1605–1612, 2004.

[52] Damiano Piovesan. Available at http://protein.bio.unipd.it/ring/.

[53] Steve Plimpton. Fast parallel algorithms for short-range molecular dynamics. *Journal of Computational Physics*, 117(1):1–19, 1995.

[54] QUORA. Unmyelinated axons.

[55] Rachel. Why are there 20 amino acids? Available at https://www.chemistryworld.com/features/why-are, Sep 2018.

[56] RCSB_PDB. Available at https://www.rcsb.org/structure/3RVY.

[57] RCSB_PDB. Available at https://www.rcsb.org/structure/6A90.

[58] RCSB_PDB. Available at https://www.rcsb.org/structure/5HVX.

[59] Peter J. Russell. *IGenetics: a molecular approach*. Benjamin Cummings, 2010.

[60] Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S Baliga, Jonathan T Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research*, 13(11):2498–2504, 2003.

[61] John Shawe-Taylor and Nello Cristianini. *Kernel methods for pattern analysis*. Cambridge University Press, 2012.

[62] Huaizong Shen, Zhangqiang Li, Yan Jiang, Xiaojing Pan, Jianping Wu, Ben Cristofori-Armstrong, Jennifer J Smith, Yanni KY Chin, Jianlin Lei, Qiang Zhou, et al. Structural basis for the modulation of voltage-gated sodium channels by animal toxins. *Science*, 362(6412):eaau2596, 2018.

[63] Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(Sep):2539–2561, 2011.

[64] Toshihiko Sugiki, Naohiro Kobayashi, and Toshimichi Fujiwara. Modern technologies of solution nuclear magnetic resonance spectroscopy for three-dimensional structure determination of proteins open avenues for life scientists. *Computational and structural biotechnology journal*, 15:328–339, 2017.

[65] Yu Tang, Min Li, Jianxin Wang, Yi Pan, and Fang-Xiang Wu. Cytonca: a cytoscape plugin for centrality analysis and evaluation of protein interaction networks. *Biosystems*, 127:67–72, 2015.

[66] Shimon Ullman. Using neuroscience to develop artificial intelligence. *Science*, 363(6428):692–693, 2019.

[67] S Vichy N Vishwanathan, Nicol N Schraudolph, Risi Kondor, and Karsten M Borgwardt. Graph kernels. *Journal of Machine Learning Research*, 11(Apr):1201–1242, 2010.

[68] Junmei Wang, Piotr Cieplak, and Peter A Kollman. How well does a restrained electrostatic potential (resp) model perform in calculating conformational energies of organic and biological molecules? *Journal of computational chemistry*, 21(12):1049–1074, 2000.

[69] Andrew Waterhouse, Martino Bertoni, Stefan Bienert, Gabriel Studer, Gerardo Tauriello, Rafal Gumienny, Florian T Heer, Tjaart A P de Beer, Christine Rempfer, Lorenza Bordoli, et al. Swiss-model: homology

modelling of protein structures and complexes. *Nucleic acids research*, 46(W1):W296–W303, 2018.

[70] Wikipedia. Available at https://en.wikipedia.org/wiki/DSSP_(hydrogen_bond_estimation_algorithm).

[71] Frank H Yu and William A Catterall. *Overview of the voltage-gated sodium channel family*, volume 4. Genome Biology, 2003.

[72] Yang Zhang. *I-TASSER server for protein 3D structure prediction*, volume 9. 2008.

[73] Zhang-Lab_University_of_Michigan. Available at https://zhanglab.ccmb.med.umich.edu/FG-MD/.