UNIVERSITÀ DEGLI STUDI DI SALERNO

UNIVERSITÀ DEGLI STUDI DI SALERNO
Dipartimento di Farmacia

PhD Program
in **Drug Discovery and Development**
XXX Cycle — Academic Year 2017/2018

PhD Thesis in

Computational design of new antimicrobial peptides

Candidate

Rosaura Parisi

Supervisor

Prof. *Stefano Piotto*

PhD Program Coordinator: Prof. Dr. *Gianluca Sbardella*

To my brother...

Table of contents

Abstract

Antimicrobial peptides (AMP) are evolutionarily conserved components of the innate immune system. They have a broad spectrum of action against bacteria, fungi and viruses. Therefore, AMP are studied as probable substitutes of the traditional antibiotics, for which most pathogens have developed resistance.

The main objective of this work was the design of novel linear peptides capable to interact with the cellular membrane of the common pathogens.

In this work, sequences of active AMP were carefully obtained from the scientific literature and collected in Yadamp (http://yadamp.unisa.it/), a database of AMP created recently in the laboratory where this project was carried out. In Yadamp, there are information about peptides name, amino acid sequence, length, presence of disulfide bridges, date of discovery, activity and taxonomy. The most relevant chemical-physical properties are also listed. This database is mainly focused on the peptides activities. Experimental MIC values (the lowest concentration of an antimicrobial that inhibits the visible growth of a microorganism) are constantly obtained from careful reading the original papers. In this work, a great contribution was made in the enrichment of the database. In fact, 1009 sequences were added to Yadamp. It currently contains 3142 AMP sequences. For these AMP, 573 molecular descriptors were calculated. In addition, this project also involved the search for new molecular descriptors. Yadamp is a resource for QSAR investigations on AMP. It allows to create subsets of AMP, homogeneous in one, two or more parameters. The working hypothesis was that AMP with similar chemical physical features can share the same mechanism of action. Therefore, during this work, genetic algorithms (GA), artificial neural networks (ANN) and classification analyses were performed on homogeneous subsets of AMP. AMP with activity against five different microorganisms were studied: *Staphylococcus aureus* and *Bacillus subtilis* (Gram + bacteria), *Escherichia coli* and *Pseudomonas aeruginosa* (Gram - bacteria), and *Candida albicans*

(saprophytic fungus). Numerous prediction models of activity were obtained, each of them validated through effective statistical techniques. These obtained models gave a preliminary idea of the probable mechanism of action that the studied AMP have. For example, the results suggest that the charge and the hydrophobicity of the amino acid residues are important factors for the binding of the AMP to the target membranes.

However, the descriptors 1D and 2D currently available fail to capture all of the peptides properties. The peptides are extremely flexible molecules and when they interact with the target membranes, they undergo conformational changes. Consequently, one of the goal of this project was also to find new molecular descriptors of AMP. For example, a new molecular docking software (www.yada.unisa.it) was developed in our laboratory. The idea was to use YADA to calculate the binding energy of the interaction between the AMP and other peptides, protein receptors and target membranes.

All the models obtained by computational studies were implemented in the "Yadamp predict" tool (http://yadamp.unisa.it/predict.aspx). It allows researchers to submit sequences of unknown molecules and to see if and to which organisms these molecules are potentially active.

In this work, 10000 amino acidic sequences were generated through a combinatorial calculation. The "Yadamp predict" tool allowed the prediction of the interaction between these peptides and the lipid membranes of specific pathogens. The results of the "Yadamp predict" tool suggested a specificity of three sequences toward Gram positive bacterial membranes. These peptides, called p458 (WMLKKFRWMF), p459 (KILGKLWKWVK) and p460 (KILKKIKKLLW), were synthesized for further analysis. Since the 3 peptides contained tryptophan, an aromatic amino acid with a maximum absorption and emission of 280 nm and ~ 360 nm, the peptides binding was monitored via spectrophotometric assays. This interaction was tested in vitro on unilamellar vesicles of 400 nm having different lipid composition. According to in silico studies, the fluorescence and absorbance results suggest that the three peptides predominantly bind Gram + bacteria. They probably bind the target membranes through a mechanism of action that does not depend only on electrostatic

interaction, but also on structural changes that occur in the lipid membrane after the binding process. Highlights on the mechanism of interaction were provided by all atoms molecular dynamics simulations (data not shown) carried out in the lab of Prof. Piotto.

All together, these findings support the proposed mechanism of action of the 3 peptides and pave the way for novel and more focused design of antimicrobial peptides.

Chapter I

1.1 Introduction: Why AMP?

"*I did not invent penicillin. Nature did that. I only discovered it by accident*." **Alexander Fleming**

In 1945, Fleming, Florey and Chain received the Nobel for Medicine for the discovery and the development of penicillin, a group of antibiotics that act against many bacterial infections. This event changed the course of history and started the "antibiotic age". Subsequently, other antibiotics were identified. Unfortunately, their massive use generated a phenomenon known as "drug resistance" that is a limit to the choice of an efficient antibiotic therapy. The annual death-toll is >700.000 people world-wide, rising to ~10 million by 2050 [1]. Consequently, we could be moving towards a "post-antibiotic era": there is a strong need for new antibiotics to limit the risk that many bacterial infections become incurable [2].

Antimicrobial peptides are currently the most promising strategy against various pathogenic microbes. Compared to the traditional antibiotics, AMP are more stable and have lower propensity for developing resistance [3]. They can interfere with cell membranes without specific receptors (membranolytic activity) and kill or inhibit the proliferation of important multidrug resistant microorganisms. Most peptides are not cytotoxic against mammalian cells [4].

1.1.1 What are AMP?

AMP are small molecules (less than 100 amino acids) produced by the immune system of bacteria, insects, plants and vertebrates. The presence of amino acids as Lys and Arg in their sequence, determines a net positive charge. In addition, there is a large proportion of hydrophobic residues. The balance between positively charged and hydrophobic amino acids determines the amphipathic conformation of AMP (figure 1) [5].

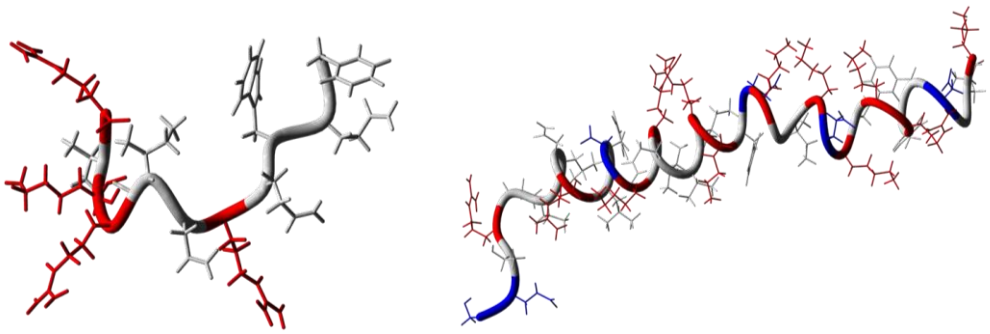


Figure 1 Two examples of AMP: on the left **2LX2**, Human Defensin 5; on the right **2K6O**, Human LL-37. Positive amino acid residues are colored in red, negatively charged residues are colored in blue, neutral residues are gray.

This amino acid composition suggests that they perform their lethal action by targeting lipid membranes [4]. AMP rapidly kill bacteria, yeasts, fungi and viruses with micromolar or submicromolar minimal inhibitory concentrations (MIC) [6]. MIC is the lowest concentration of an antimicrobial peptide which prevents visible growth of a bacterium. Several AMP have been identified and characterized. They have a considerable diversity in sequence, structure and biological activity. Based on their secondary structure, AMP are grouped in α-helical, β-sheet, extended and loop peptides [3]. The first two classes are the most common in nature. The first antimicrobial peptides identified and studied were α-helix peptides [4].

In 1987, magainins, a family of α-helix antimicrobial peptides, was isolated from the ventral skin of the African frog Xenopus laevis (figure 2) [7].

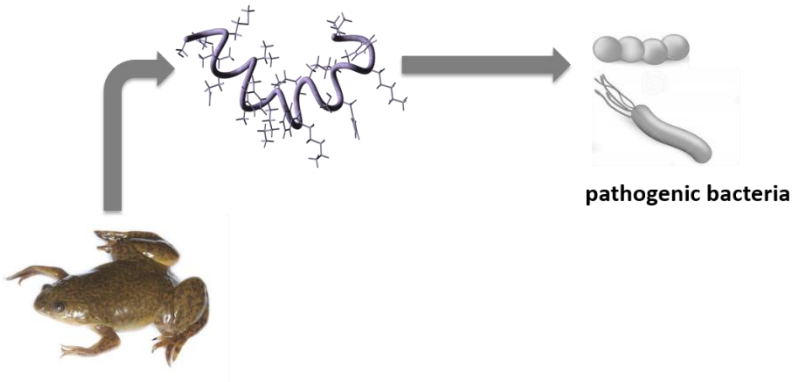


Figure 2 Magainin (PDB: **2LSA**) was the first AMP discovered. It was extracted from the ventral skin of the African frog, Xenopus leavis. This AMP has been extensively studied for its killing action on various pathogenic bacteria.

Other known α-helix antimicrobial peptides are:

- cecropins, andropins, melittins and ceratotoxins from insects;
- bombinins, dermaseptins, esculentins and buforins II from amphibians;
- cathelicidin LL-37 from human (figure 3);

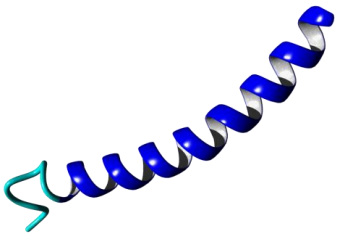


Figure 3 Human LL-37 Structure (PDB: **2K6O**)

Other antimicrobial peptides, with 16-18 amino acid residues and one or two disulfide bridges, form a single β-hairpin. Protegrin family is a model among β-sheet AMP. It includes small peptides isolated from porcine leukocytes. They have a high content of positively charged arginine (Arg) and cysteine (Cys) residues. For example, PG-1 is a one-turn β-hairpin peptide in which two

antiparallel strands linked by a β-turn are stabilized by two disulphide bonds [8] (figure 4).

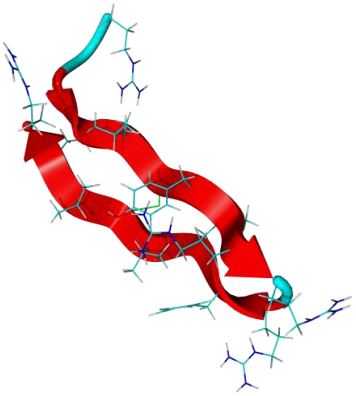


Figure 4 Protegrin 1 (PDB: **1PG1**)

Linear extended peptides are very flexible in solution and they do not fold in a regular secondary structure. They are rich in proline, arginine and aromatic amino acids. Two examples are tritrpticin (figure 5) and indolicidin [9].

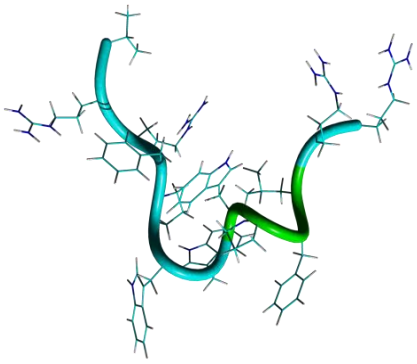


Figure 5 Tritrpticin (PDB: **1D6X**)

There is a small group of AMP that, in the C-terminal of the structure, adopts a loop formation with one intramolecular disulfide bridge. For example, thanatin, an antimicrobial peptide with 21 amino acids, includes two cysteine residues that form a disulfide bridge. Thanatin adopts a well-defined anti-parallel β-sheet structure from residue 8 to the C-terminus [10] (figure 6).

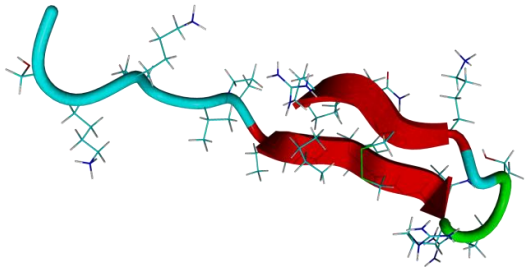


Figure 6 Thanatin (PDB: **8TFV**)

1.1.2 Mechanisms of action: current ideas

The behavior of antimicrobial peptides in vivo is not yet clear and each peptide can have various mechanisms of action. A common denominator for the mechanism of action of all these molecules, is the interaction with the lipid membranes of the target organisms. The positive charge of AMP allows an electrostatic interaction with the negatively charged surfaces, such as the bacterial membranes. This is an important aspect for the selectivity of AMP (figure 7). In fact, the eukaryotic cell membranes are predominantly constituted by zwitterionic lipids and the interaction with antimicrobial peptides is very weak [5]. The AMP amphipatic conformation guarantees their insertion into the lipid layer and, for example, an action mechanism consisting in the formation of pores.

Recent studies demonstrate that, in addition to antimicrobial activity, AMP can have anticancer activity. An increased exposure of phosphatidylserine (PS) and the presence of O-glycosylated mucines, are determining factors for the interaction of antimicrobial peptides to the surface of cancer cells [11].

Some AMP can have a cytotoxic activity. For example, the cationic amphiphilic antimicrobial peptide gramicidin S (GS) has an applicability restricted to topical infections due to its hemolytic activity [12].

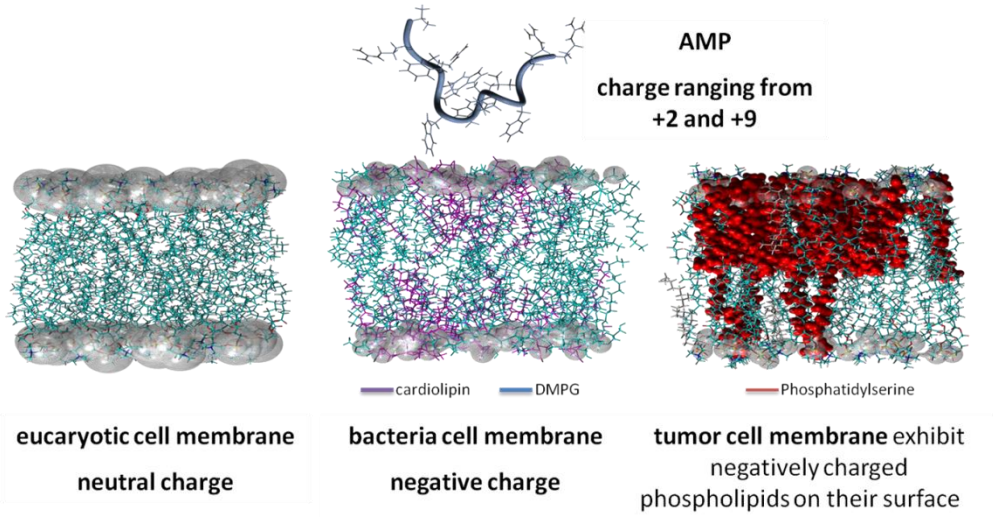


Figure 7 The positive charge of the antimicrobial peptides allows them to preferentially bind the bacterial membranes negatively charged. The tumor cell membranes exhibit negatively charged phospholipids on their surface, such as the phosphatidylserine (PS). Thus, cancer cells can also be a target for AMP.

Recent studies with fluorescent probes show that AMP can be associated with cell division, cell wall remodeling and secretion. They can interfere with these processes and/or cause cell lysis [5]. The mechanism of action of antimicrobial peptides depends mainly on their secondary structure and on the lipid composition of the target membranes. At the same time, the perturbation that occurs on the target membranes, depends on the peptide concentration, pH and temperature. For this reason, many studies have been performed to clarify the mechanism of action of AMP and to design selective peptides toxic only for pathogenic organisms. It emerges from the literature that AMP have four different mechanisms of action: barrel-stave, carpet, toroidal-pore and detergent model [13]. In the barrel-stave model (figure 8), AMP insert into the target membrane and form a transmembrane pore, where hydrophobic amino acids interact with the lipid core of the bilayer. This mode of action causes cell lysis [14]. In 1998, Matsuzaki and others studied the mechanism of action of some AMP and they proposed that magainin permeabilizes the phosphatidylglycerol bilayers by forming a pore (figure 8) [15].

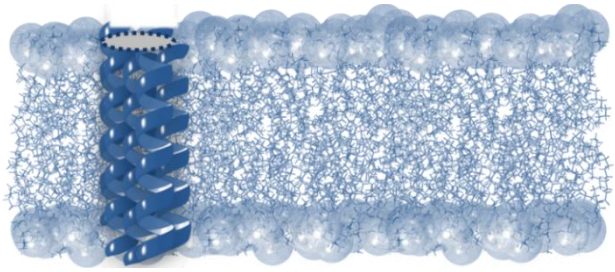


Figure 8 AMP insert into the target membrane and form a transmembrane pore.

In the carpet model (figure 9), high concentration of AMP cover the outer surface of the bacterial membrane like a carpet. The peptides are in contact with the hydrophilic heads of the lipids and this causes a new orientation of the hydrophilic residues and the creation of a hydrophobic core. The deformation of the membrane curvature causes cell breakage and death.

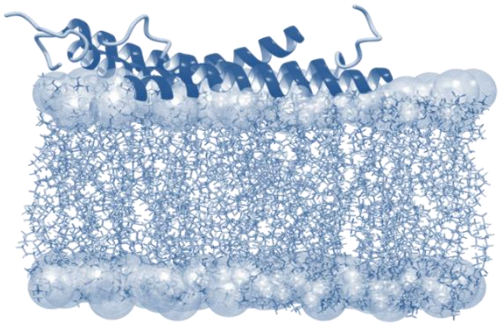


Figure 9 AMP on the lipidic surface like a carpet

The toroidal-pore model is a variant of the barrel-stave model. In this process, the membrane is bent inward and the intercalation of AMP with phospholipids forms the pore (figure 10).

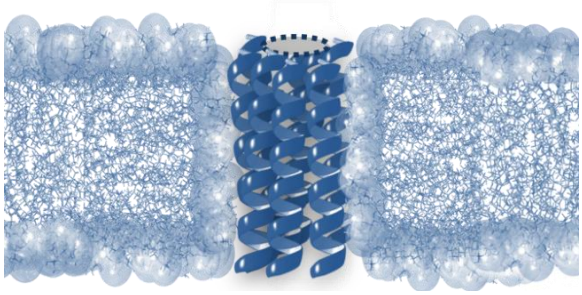


Figure 10 AMP induce local defects in the bilayer and form a toroidal pore. The head-groups of the lipids line the pore together with the peptides.

AMP at a high concentration can act as "detergents" to form peptide/lipid micelles, resulting in a collapse of the membrane [16] (figure 11).

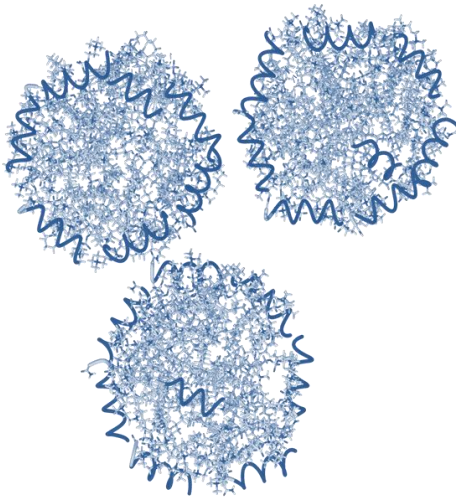


Figure 11 AMP act as "detergents" and form peptide/lipids micelles.

The models just described are very useful to interpret molecular phenomena. Some studies have shown that alpha-helix peptides and beta-peptides act on the membranes predominantly forming pores. Among them, for example, there are the alpha-peptides Magainin II and Cecropin or the bovine beta-peptides Lactoferricin and Protegrin I. Extended peptides, such as Indolicidin or Pyrrhocoricin, that contain high proportions of certain amino acids (Arg, Pro or Thr residues), can cause cell membrane depolarization, lysis and inhibition of the DNA synthesis [3]. Some studies demonstrated that the chemical-physical characteristics of the

antimicrobial peptides, such as conformation, charge, hydrophobicity and amphipathicity, impact their selective toxicity [17]. Changes in composition, sequence and intramolecular bonds affect the structure-activity relationships of AMP.

1.1.3 Bacterial resistance strategy

There are few naturally AMP-resistant organisms, such as *Burkholderia, Proteus* and *Serratia sp*. [18], but the molecular bases for this peptide resistance are not clear. For example, some studies suggest that constitutive alterations in cytoplasmatic membrane structure or function may be the key to determine antimicrobial peptide resistance in *S.aureus*. At the same time, modifications in the outer membrane of some Gram-negative bacteria preserve the membrane integrity when it interacts with antimicrobial peptides [17]. Many studies have also identified genes in the bacterial DNA that result in AMP resistance. For example, the gene *mcr-1* in *Escherichia coli* and *Klebsiella pneumonia,* codifies for a transferase that modifies the lipid A in the membrane, in order to reduce the anionic charge [19]. For example, the inactivation of the genes *lpxA, lpxD, or lpxC* in the DNA of *Acinetobacter baumannii,* that are involved in lipid A biosynthesis, determine the loss of LPS production and a reduction of the AMP binding [20]. Other mechanisms that bacteria use to overcome the action of AMP could involve the production of extracellular proteases and biofilms. For example, some bacteria, such as S.*epidermidis,* form biofilms to prevent AMP insertion and pore formation [21]. Overexpression of efflux pumps on the cell membrane, with the function of ejecting AMP from the cell, is also one of the resistance mechanisms of bacteria [22].

In conclusion, there are resistance mechanisms that bacteria perform against AMP. However, despite the fact that bacteria are exposed to antimicrobial peptides for millions of years, the development of these forms of resistance occurred to a much less degree. In fact, to develop resistance against AMP microorganism should redesign their membrane and this is very hard in terms of

energy. Furthermore, there is a large number of antimicrobial peptides in the host and it is very difficult for bacteria to develop resistance against all peptides at the same time.

1.1.4 Limits of antimicrobial peptides as therapeutic agents

AMP possess several disadvantages that limit their development as therapeutic agents. These disadvantages include hemolytic activity, broad spectrum of activity, protease susceptibility, a rapid turnover in the human body, salt sensitivity and high cost of production [23]:

- **Hemolytic activity**

The bacterial cell membrane is negatively charged compared with mammalian cell membrane. This property increases the affinity between the cationic antimicrobial peptides and the bacterial cell membranes. The binding of AMP to the mammalian cells membranes depends on lipid charge, lipid composition and/or transmembrane potential. However, all AMP present a certain level of cytotoxicity towards mammalian cells.
For example, indolicidin and bactenecin are strongly toxic to rat and human T lymphocytes [24]. It was observed that some natural AMP with amidated C-terminal show higher hemolytic effect [25]. The ratio of antimicrobial activity and hemolytic activity is defined "therapeutic index". A high therapeutic index is necessary to avoid hemolysis of host cells [23].

- **Broad activity spectrum**

AMP are fascinating alternatives to antibiotics because of their broad-spectrum activity against various microorganisms, including Gram-positive and Gram-negative bacteria, fungi, and viruses [26]. This positive aspect is also a

disadvantage. In fact, during therapy, AMP could also act on the bacterial microflora of the organism, causing other infections.

- **Rapid turnover in the human body: protease susceptibility**

Antimicrobial peptides are susceptible to proteases: they are rapidly degraded in the human body. Proteolytic stability is essential for therapeutic use and there are a lot of strategies to overcome this problem. For example, the most frequent modification is the replacing of natural amino acids with D-amino acids in the sequence: the proteases cannot act on unnatural residues [27, 28]. Another way to increase the stability of antimicrobial peptides is to create peptide mimetics. For example, De Grado *et al*. designed a series of amphiphilic arylamide polymers with the same physical properties of AMP. They have a mechanism of action comparable to the conventional antibiotics and exhibit high stability against proteases [29, 30]. Finally, cyclization seems to be a good approach to improve the pharmacodynamics of AMP [23].

- **Salt sensitivity**

When antimicrobial peptides interact with the target membranes, they form secondary structures. This step is particularly sensitive to the high concentrations of salts present in body fluids. For this reason, it is necessary to design peptides that are not sensitive to salt. For example, the introduction of helix-capping motifs is a way to elude this problem and to stabilize the structure of AMP [31].

- **High cost of production**

The cost of production of AMP is very high compared to the production of the conventional antibiotics. In general, to isolate peptides from natural sources is a very intensive and time-consuming process. At the same time, the process of

chemical synthesis of peptides is complex and costly. An efficient strategy is to obtain peptides through a recombinant production in various heterologous hosts, such as *E.coli* [32].

1.1.5 AMP in clinical trial

AMP are considered molecules that may be able to replace traditional antibiotics in the difficult treatment of multidrug-resistant bacteria [33]. Actually, there are two strategies to exploit the action of antimicrobial peptides. The first one consists in a synergistic action of antimicrobial peptides with drugs that act directly on bacteria. The second strategy involves the development of substances that increase the production of antimicrobial peptides in the patient [33].

Currently, eleven antimicrobial peptides are in phase I, twenty-four in phase II, three AMP are in phase III and only three studies are in phase IV (https://clinicaltrials.gov). For example, actually, the MD Anderson Cancer Center is studying the induction of antitumor response in melanoma patients using the Antimicrobial Peptide LL37 (phase I). The goal of this clinical research study is to find the appropriate dose of LL37 for patients with melanoma. A research that is in phase II of the study concerns the evaluation of the safety and the bacterial impact of the drug STAMP C16G2 gived in multiple oral gel doses to adolescent and adult subjects. Another study, in the phase III, regards the comparison of the MSI-78 (magainin peptide) topical therapy and a conventional oral antibiotic therapy in the reduction in symptoms of diabetic foot. Particular is the case of Pexiganan (Locilex®), a chemically synthesized 22-amino-acid peptide, isolated from the skin of the African clawed frog. It is a foot ulcer candidate drug, which, unfortunately, has failed two phase III. Its action, in treating the disease, is considered not better than traditional antibiotics. Dipexium Pharmaceuticals, in disagreement with these results, has chosen to continue the study of this drug.

Chapter II

2.1 Molecular descriptors of antimicrobial peptides

"The molecular descriptor is the final result of a logic and mathematical procedure which transforms chemical information encoded within a symbolic representation of a molecule into a useful number or the result of some standardized experiment." [34]

Molecular descriptors are widely used in computational chemistry in order to find a relationship between the structure and the activity of a molecule. These chemical-physical properties of the molecules are the elements on which a QSAR analysis is based. They reflect various levels of chemical structure: the molecular formula (so-called 1D), the two-dimensional (2D) and three-dimensional (3D) structural formula, the orientation and time-dependent dynamics of molecules (4D and higher) [35]. In this work, one-dimensional (1D) and two-dimensional (2D) molecular descriptors were used. The two-dimensional representation of molecules defines the connectivity of atoms in the molecule in terms of the presence and nature of chemical bonds [35]. The amino acid sequence, the length, the presence of disulfide bridges, the charge, the hydrophobic moment, the helicity, the flexibility, the isoelectric point, the Boman and instability index, the penetration capabilities and the ΔG, represent only one part of the molecular descriptors of AMP (figure 12). Molecular descriptors, such as helicity, isoelectric point, Boman index, flexibility or hydrophobicity, were calculated using online tools or MATLAB script. Others molecular descriptors were calculated from the AAindex database (http://www.genome.jp/aaindex/), using a MATLAB script. AAindex is a database of numerical indices representing various physicochemical and biochemical properties of amino acids [36]. In this database, the molecular descriptors are indicated with a code of four letters and six numbers (for example NAKH900104 or PALJ810106). All data are obtained from the literature.

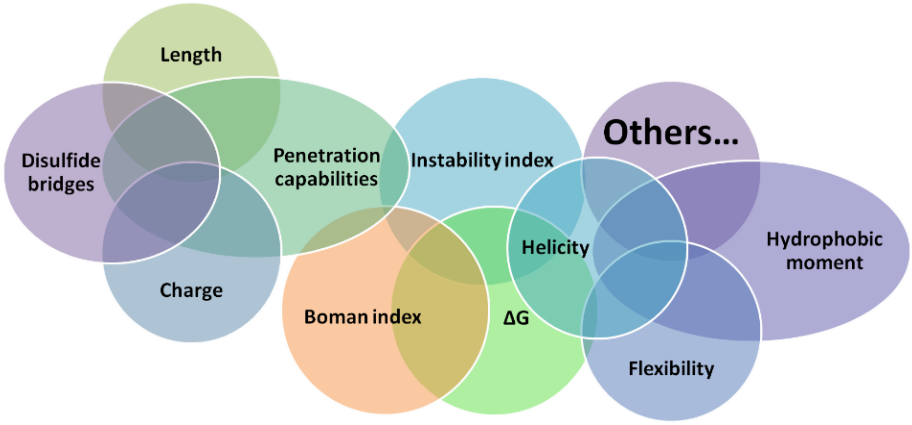


Figure 12 Some molecular descriptors used in the QSAR analysis performed on antimicrobial peptides.

2.1.1 Yadamp: yet another database of antimicrobial peptides

In recent years, many novels antimicrobial peptides have been discovered and characterized. Data concerning these antimicrobial peptides have been included in web databases, most of which, dedicated to specific classes of AMP (table 1). Examples of these are PhytAMP [37], BACTIBASE [38] and DAMPD [39]. APD3 [40] is available since 2003 and contains 2884 active sequences on bacteria. In this database, information about the MIC (the lowest concentration of an antimicrobial peptide which prevents visible growth of a bacterium), is not directly accessible. Another database, CAMP [41], contains 1386 AMP sequences, but only a fraction of them are completed with MIC values.

Current web databases	
CAMP: Collection of Antimicrobial Peptides	• 1386 AMP • Information about sequence, protein definition, accession numbers, activity, source organism, target organisms, protein family descriptions and links to other antimicrobial peptide databases
APD3: The Antimicrobial Peptide Database	• 2884 AMP • Information about peptide name, amino acid sequence, peptide motifs, chemical modifications, length, charge, hydrophobic content, PDB ID, 3D structure, methods for structural determination, peptide source organism, peptide family name, life domain/kingdom biological activity, synergistic effects, target microbes, molecular targets, mechanism of action, contributing authors, and year of publication.
DAMPD: Dragon Antimicrobial Peptide Database	• 232 AMP • Information about taxonomy, species, AMP family, citation, keywords and a combination of search terms and fields
PhytAMP: A Database Dedicated to Antimicrobial Plant Peptides	• Plant antimicrobial peptides • Information about taxonomic, microbiological and physicochemical data.
BACTIBASE: Database Dedicated to Bacteriocin	• 230 bacteriocins • Microbiological and physicochemical data

Table 1 Some of the most widely used AMP databases

Due to their considerable diversity in chemical-physical properties and in their mechanism of action, a classification of AMP is very difficult [42]. In 2012, in the

laboratory where this project was performed, a database of antimicrobial peptides was developed: YADAMP (http://yadamp.unisa.it/) [43]. It contains more quantitative data than any other database. In YADAMP, there are relevant molecular descriptors for AMP: *charge*, *hydrophobic moment*, *helicity*, *flexibility*, *isoelectric point*, *Boman index*, *instability index*, and many others. YADAMP, unlike other databases, is especially focused on peptides MIC. The idea was to create a resource for researchers to retrieve all information on antimicrobial peptides in a short time, to select and to cluster the peptides according to certain parameters. "YADAMP predict" is a tool, implemented in the YADAMP database, that allows researchers to submit sequences of unknown molecules and to see if and to which organisms these molecules are potentially active (figure 13). Users can also know the degree of reliability of their result through appropriate statistical validation systems.

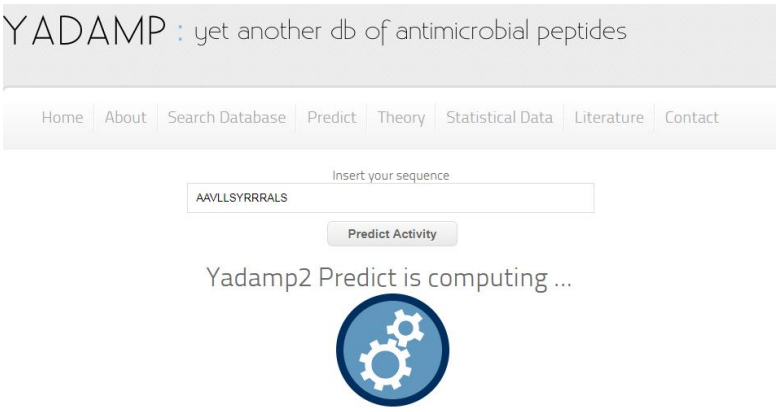


Figure 13 YADAMP database interface

2.1.2 Data collection

Yadamp collects data about AMP from scientific papers or web databases, providing structural data and information on antimicrobial activities.
In Yadamp, a user can obtain information about peptide name, amino acid sequence, length, presence of disulfide bridges, date of discovery, activity and taxonomy. In addition, the most relevant chemical-physical properties were calculated such as charge, hydrophobic moment, helicity, flexibility, isoelectric point, Boman index, instability index and penetration capabilities.

Sequences

Sequences of active AMP were mainly extracted from the scientific literature and were compared with data in public databases (UniProtKB/Swiss-Prot [44], APD2 [45], CAMP [41]). Each of the collected sequences was validated with literature available data.

MIC values

Yadamp is mainly focused on peptides activities. In microbiology, the MIC is the lowest concentration of an antimicrobial that inhibits visible growth of a microorganism. Yadamp allows the selection of AMP with the lowest MIC value. Experimental MIC values (expressed in μM) were manually extracted from careful reading the original papers. MIC values expressed in μg/mL were converted to μM to allow a quick comparison, using the formula:

$$\textit{eq.1}\quad \boldsymbol{Concentration}\ (\mu\mathbf{M}) = \frac{Concentration\frac{\mu g}{mL}}{Molecular\ weight\ peptide} \times 1000$$

The most intensively studied organisms are *Escherichia coli* (Gram -), *Pseudomonas aeruginosa* (Gram -), *Salmonella enterica* serotype *Typhimurium* (Gram -), *Staphylococcus aureus* (Gram +), *Micrococcus luteus* (Gram +), *Bacillus subtilis* (Gram +) and the fungus *Candida albicans*. These organisms are also the primary source of infections in humans. Data against all other bacteria were inserted in fields: Other Gram −, Other Gram + and Other for fungi and yeast. In addition, for each peptide, in YADAMP the link to the references from which information were extracted were added in order to check the data and the antibacterial assay conditions.

Biological classification

Biological classification is a method used to group and categorize organisms into groups having attributes or traits in common. Taxonomy materials are important to understand and identify sequence patterns conserved across species. It is a hierarchical classification, in which each level is named 'rank'. The data about taxonomic information were extracted from NCBI Taxonomy database [46]. YADAMP permits a selection for five main ranks: *phylum, class, order, family,* and *genus*.

Calculated parameters

Yadamp was created to be a resource for QSAR investigations on AMP. For an accurate QSAR analysis, it is essential to group peptides sharing some features, such as similar secondary structure, flexibility or charge. For this reason, Yadamp enriches the experimental data with some theoretical information.
In the *Appendix A* it is possible to read the MATLAB scripts associated with each parameter computed: *Charge, Boman index, Hydrophobicity, Helicity, Instability index, CPP*.

- **Charge**

AMP can act in very different pH conditions, depending on the tissue in which the bacteria are grown. The charge of each peptide was calculated at three different pH values (pH 5, 7 and 9) by the formula:

$$\boldsymbol{eq.2}\quad \boldsymbol{Charge} = \sum_{i} N_i \frac{10^{pKai}}{10^{pH} + 10^{pKai}} - \sum_{j} N_j \frac{10^{pH}}{10^{pH} + 10^{pKaj}}$$

where N_i is the number of the *N*-terminus and of the side chains of arginine, lysine and histidine. The *j*-index refers to the *C*-terminus and the aspartic acid, glutamic acid, cysteine and tyrosine amino acids. pK_{a_i} and pKaj values, taken from Lehninger Principles of Biochemistry [47], refer to amino acids labeled with the index *i* and j.

This algorithm has some limitations, such as:

- the residues are assumed to be independent of each other;
- N- and C-termini have fixed pKa values;
- only the 20 natural amino acids are considered;
- the resulting net charge depends on what pKa values were used;

A quick inspection at the database reveals that, mainly because of the wide variation in lysine abundance, the charge of certain peptides can largely vary at different pH. Peptides acting as antimicrobial compounds do not always experiment the neutral pH, so this parameter can be decisive for peptide simulations in specific tissue.

- **Isoelectric point**

The isoelectric point (pI) is the pH at which a protein has no net electrical charge. Below the pI proteins carry a net positive charge and above it they have a net negative charge. Theoretical pI values were calculated using a free online tool [48]. According to Bjellqvist et al. [49] it was assumed that the same pK value could be used for an amino acid residue in all polypeptides and in all positions in the peptide except for *N*- or *C*-terminally placed amino acids. For the pK values of the *N*-terminal amino groups the effect of the different substituents on the α-carbon were taken into account.

- **Boman index**

Most authors have agreed that a potential AMP should possess a positive net charge to facilitate binding to bacterial phospholipids as well as a certain degree of amphipathicity to allow molecule adaptation to a bacterial membrane. These criteria are not enough to predict the ability of a peptide to interact with cell membrane. Boman [50] introduced a parameter which shows a certain degree of discrimination between membrane-interacting and protein-interacting peptide. This value established the tendencies of amino acids to leave water and move in a nonpolar condensed phase calculating the distribution coefficients for each side chain of the natural amino acids at pH 7. The Boman index for all sequences was calculated as the sum of the free energies (kcal/mol) of the respective amino acid side chains for transfer from cyclohexane to water divided by the total number of residues (eq.3).

$$\boldsymbol{eq.3 \quad Boman\ index} = \sum_{i} \frac{free\ energies\ \left(\frac{kcal}{mol}\right)}{residues}$$

The free energies values were taken from Radzeka and Wolfenden [51].

- **Hydrophobicity**

Hydrophobicity is another critical characteristic of amino acid residues that determines protein folding, protein subunit interaction, binding to receptors, and interactions of proteins and peptides with biological membranes. The calculation of hydrophobicity assigns a numerical hydrophobicity value to each type of amino acid, and then relates these values in a particular protein or fragment to some aspect of the structure or the function. The hydrophobicity of an amino acid residue is not a property that can be easily defined or simply measured. Nevertheless, several groups have attempted to derive numerical hydrophobicity scales using a variety of experimental and computational methods. The distribution of the hydrophobic residues in amphipathic peptides is revealed by the hydrophobic moment, which depends on the spatial conformation of the peptide. To calculate the hydrophobicity of the AMP sequences it was used the method of Eisenberg, David, et al. (1982) [52]. It was assumed that the polypeptide backbone follows some periodic arrangements such as an α-helix or a strand from a β-sheet. The hydrophobicity of each residue *i* by a vector of length H_i, having a direction perpendicular to the axis of the helix or strand of beta structure. The value of the estimated hydrophobic dipole moment (μH) is:

$$\boldsymbol{eq.4} \quad \mu_{\boldsymbol{H}} = \left(\left(\sum_i H_i \cos(i\delta)\right)^2 + \left(\sum_i H_i \sin(i\delta)\right)^2\right)^{1/2}$$

where **δ** is the angle separating side chains along the backbone (e.g. **δ** = 100° for an α-helix). Finally, the mean hydrophobicity was calculated as the total hydrophobicity (sum of all residue hydrophobicity indices) divided by the number of residues (eq.5).

$$\boldsymbol{eq.5} \quad \boldsymbol{mean\ H} = \sum_i \frac{\mu H_i}{residues\ number}$$

- **Helicity**

The secondary structure of a peptide is crucial for the investigation. If it is not experimentally available, peptide structure prediction is essential.

In Yadamp, the prediction is based upon the DSC (Discrimination of protein Secondary structure Class) algorithm from King and Sternberg [53]. The method extracts the maximum information from the primary sequence and allows the prediction of the secondary structure from multiply aligned homologous sequences and linear statistics. The DSC Method is accessible as 'Secondary Structure Prediction' (SSP) option in Discovery Studio from Accelrys.

- **Flexibility**

The molecular flexibility of proteins is a crucial factor in determining their biological activity, including binding affinity, and for the theoretical understanding of peptide dynamics. The identification of regions in proteins with the highest conformational flexibility and rigidity is essential for predicting the mechanism of protein folding. Consequently, there is a considerable interest in predicting the flexibility or, conversely, the rigidity of peptides from their amino acid sequence. Obviously, prediction of the secondary structure of an AMP is a hard task due to the different conformations that a peptide shows in different chemical environments. Moving from the water bulk into the membrane, the structure of peptides varies considerably. The flexibility of α-AMP was calculated according to a conformational flexibility scale for amino acids in peptides [54], which provides an absolute measure for the time scale of conformational changes in short unstructured peptides as a function of the

amino acid type. This experimental scale derived from kinetic measurements of the collision frequency between the two ends of short random-coil polypeptides. These peptides were labeled with a fluorescent probe at the *C*-terminus and Trp as a fluorescence quencher at the *N*-terminus. The fluorescence lifetimes of fluorescent probe/Trp peptides provide the quenching rate constants (k_q), which measure the *end-to-end* collision frequency. The authors have shown different collision frequencies when the probe and the quencher were separated by different amino acids. This arrangement allowed them to correlate the collision frequency with the type of amino acid and build up a flexibility scale.

In YADAMP the flexibility was calculated by the formula:

$$\boldsymbol{eq.6}\quad Flexibility = \sum_i \frac{k_q}{residues\ number}$$

For amino acids not found in the Huang work, the missing values were estimated by comparison with reported k_q constants.

- **Instability index**

To estimate the instability values, the Guruprasad work was considered [55]. They made a statistical analysis of 12 unstable and 32 stable proteins to reveal patterns in the occurrence of certain dipeptides. Some dipeptides appeared particularly frequent in stable proteins, whereas other dipeptides were common in unstable proteins. The contribution of each of the dipeptides towards instability was obtained by summing the instability weight values corresponding to the conditions satisfied by the dipeptide and termed as the dipeptide instability weight value (DIWV). The instability index (II) was

calculated using the DIWV values for all 400 combinations reported in the Guruprasad paper (eq.7).

$$\boldsymbol{eq.7}\quad \boldsymbol{II} = \left(\frac{10}{L}\right)\sum_{i=1}^{L-1} DIWV(x_i y_i + 1)$$

where $\mathbf{x_i y_{i+1}}$ is a dipeptide, **L** is the length of the sequence and **10** is a scaling factor.

- **CPP**

This parameter is the acronym of Cell Penetrating Peptides and is an estimate of the tendency for a peptide to penetrate a cell membrane. The parameter can take values between 0 and 1, where 1 corresponds with the highest probability of a peptide to penetrate a membrane, and 0 indicates the impossibility to enter a membrane. To predict this ability, it was used a free online tool [56] in which the peptide sequences were inserted.

2.1.3 Extending Yadamp

Yadamp, at the time of its creation (2012), contained detailed information for 2133 peptides active against bacteria. During this work, 1009 new sequences of AMP were added to the database. All these data are the result of an extensive and careful bibliographic research from existing AMP databases and the most recent literature. For these sequences, 573 chemical-physical parameters were calculated.

2.1.4 Using Yadamp

The web interface of Yadamp offers a simple use of the database. It is possible to query the database by name, sequence, length and by other molecular descriptors.

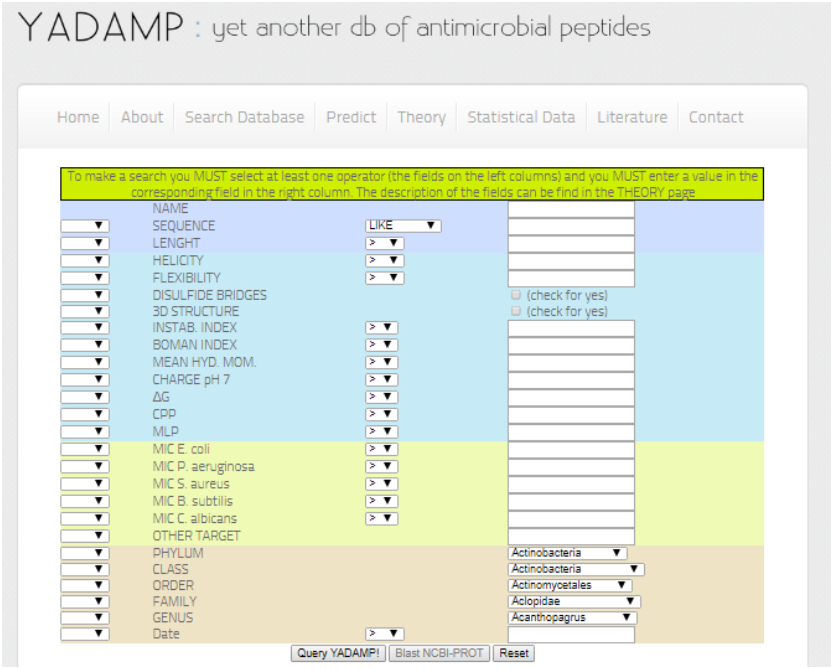


Figure 14 Yadamp interface

In Yadamp, it is possible to look for antimicrobial peptide sequences based on the antimicrobial activity that they have against five organisms (*E.coli, P.aeruginosa, S.aureus, B.subtilis* and *C.albicans*), common target of AMP. Soon, It will also be

possible to query Yadamp on the hemolytic activity of antimicrobial peptides. Information about the activity of the peptides is manually extracted by the corresponding papers and from the DBAASP database, which contains information on the antimicrobial and hemolytic activities of more than 10000 antimicrobial peptides [57]. These data are essential to optimize the analyses of structure-activity relationships, to investigate the selectivity of antimicrobial peptides and, therefore, to obtain new activity models. In the Yadamp site, the "Theory section" provides a synopsis of the theoretical terms. Finally, due to the extraordinary interest in AMP, Yadamp provides a page dedicated to literature monitoring [43].

Chapter III

3.1 Computational studies to clarify the AMP mechanism of action

More than 19.000 antimicrobial peptides, obtained from nature sources or synthesized, have been discovered and characterized, but their mechanism of action is still poorly understood [58]. It is extremely beneficial to have approaches that can guide the design of new drugs in a rational way. Therefore, some researchers developed algorithms to predict the antibacterial activity of AMP with a high accuracy. For example, Lata *et al*. (2007) [59], using Artificial Neural Network (ANN) and Support Vector Machine (SVM), suggested that N- and C-terminals of the AMP sequence, play two different important roles in the activity: C-terminal and N-terminal are involved in the interaction with the membrane and in the pore formation, while the N-terminal intervenes in specific bacterial interaction processes [59]. Other researchers developed an approach to identify conserved motifs in the AMP. They used a computational method based on hidden Markov models (HMMs) [60], a tool that represents the distributions, in terms of probability, of the sequences of observations [61]. Unfortunately, the amino acid sequences analyzed have shown little homology, precluding the possibility to create easily a model of activity. Antimicrobial peptides are very heterogeneous molecules, with a large variability in sequences and 3D structures. Probably, this is the reason for the failure of traditional computational analyses. To overcome this limitation, an element of novelty in this work is the creation of homogeneous sets of antimicrobial peptides [43], on which QSAR Analysis (Quantitative structure–activity relationship) by genetic algorithms (GA) and artificial neural networks (ANN) were performed.

3.1.1 GA and ANN: what are and how they work?

"As many more individuals of each species are born than can possibly survive; and as, consequently, there is a frequently recurring struggle for existence, it follows that any being, if it vary however slightly in any manner profitable to itself, under the complex and sometimes varying conditions of life, will have a better chance of surviving, and thus be naturally selected. From the strong principle of inheritance, any selected variety will tend to propagate its new and modified form."

Charles Darwin, The Origin of Species

Genetic algorithms are heuristic search methods based on the Darwinian theory of natural selection [62]. According to Darwin, the evolution of the species is governed primarily by the "struggle for life", for which individuals with the best genetic material have a greater chance of survival. Through sexual reproduction, the best genetic material is transmitted from parent to child. So, the individuals that best fit the environment are those who survive and transmit their characteristics to their successors [63]. To perform an analysis with genetic algorithms, molecular descriptors must be chosen.

In this work, the molecular descriptors are the chemical-physical characteristics of AMP potentially correlated with the response. In this case, the response is represented by the activity that the peptides have towards target species (MIC). The aim is to find the properties that best correlate with this response. The molecular descriptors represent the population of data on which the genetic algorithm works and they are called "individuals". Each individual is converted to a binary string, known as "chromosome", and evaluated based on a fitness function. The fitness function represents a function that makes a solution to the input problem in the form of a mathematical equation (output). The values of each binary string are called "genes". The analysis is divided into three phases: selection, crossover and mutation. In the selection phase, the selective reproduction operator plays a similar role to the law of survival in nature: at each iteration the algorithm measures the fitness value of each individual and determines the best set of solutions to solve the given problem. The best strings are subjected to genetic recombination. In the crossover phase, each pair of strings is crossed with a certain probability and the crossing point is random. Finally, in the mutation phase, each element of a string (gene) changes its value based on a

probability. The operation is repeated until the genetic algorithm chooses a suitable set of descriptors. These values are utilized to build a nonlinear QSAR regression equation (the output of the analysis).

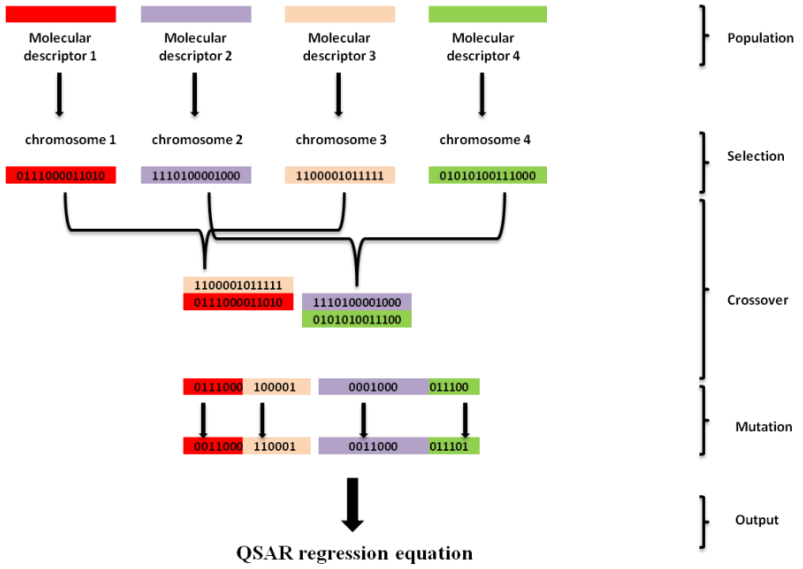


Figure 15 Representative scheme of a genetic algorithm.

The artificial neural network (ANN) analysis was developed and designed to mimic the information processing and learning in the brain of living organisms (figure 16) [64].

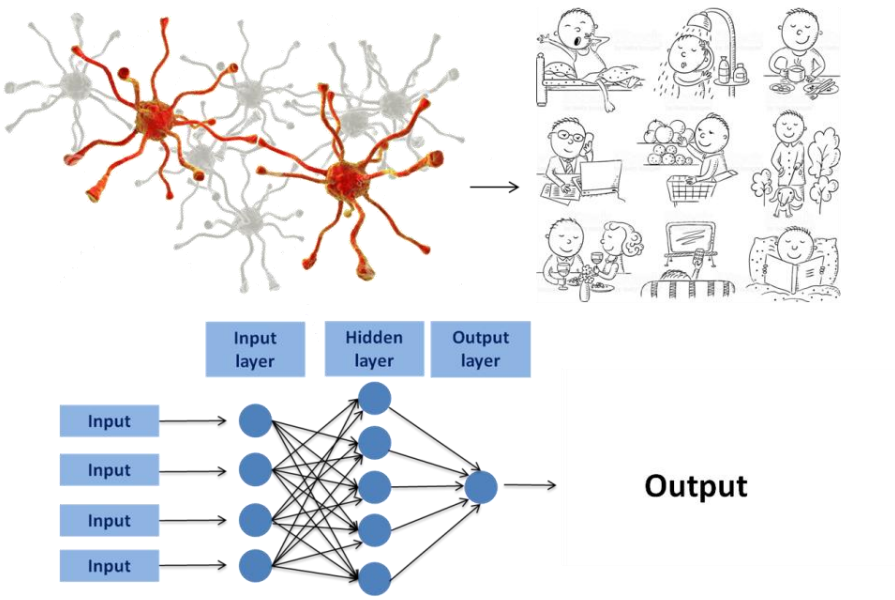


Figure 16 Artificial neural networks mimics the connection of neurons in the nervous system.

In the nervous system, a neuron receives and communicates signals to other neurons, muscles or glands. These neuronal functions are reflected in the anatomy of the neuron. The neuron receives connections from other neurons through the *dendrites*, small fibers that are branched from the neuron. The *soma* is the cell body and contains the organelles necessary for cellular function. Information from one part of the neuron to its terminal regions is transmitted through the *axon*. The terminal region of the axon is called *synapse* and here one neuron forms a connection with another through the process of synaptic transmission. The synapses can learn from the activities in which they participate and are responsible of human memory [65]. This principle is the key point of the artificial neural network architecture that acquires knowledge through learning.

A lot of researchers use artificial neural networks to solve a variety of problems (pattern recognition, prediction, clustering, etc). The first approach to the ANN research is in 1940 [66]. Papert's results [67] in the 70s created a lot of enthusiasm for researchers and ANN received considerable interest. In computational biology, an artificial neural network (ANN) consists of an input

layer of neurons, a certain number of hidden layers and a final output layer (figure 17) [68].

Typically, an artificial neuron has many inputs and one output. Inputs are the data that the operator provides to the system. The inputs are converted into vectors and indicated by the mathematical notation x (n), where n is the number of inputs. Each input is associated with a weight that indicates the conductivity of the input channel: neuron activation is a function of the weighed sum of inputs. The sum is a numerical value ranging from 0 to infinity. If the sum is 0, *bias* is added. *Bias* is always equal to '1'. To check the sum value produced by a neuron and to decide if this neuron is active or not, an activation function (or transfer function) is set. The activation function can be mainly of two types: linear or nonlinear.

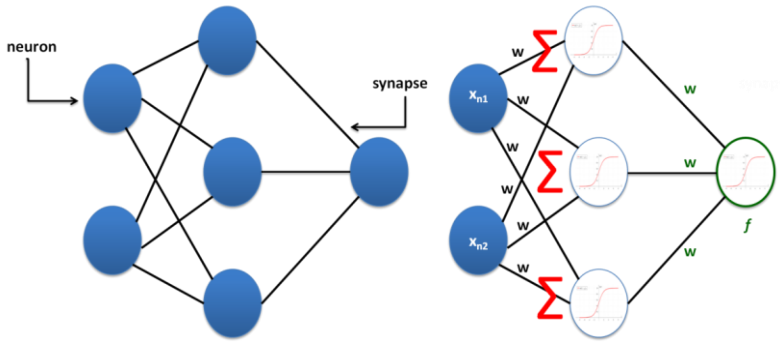


Figure 17 Representative scheme of an Artificial Neural Network.

The most commonly used method for training a neural network is to present a set of examples (training sets) to the network. The response that the neural network provides is compared to the desired response to evaluate how much they are different (error value). At each cycle, the neural network adjusts the weights associated with inputs. The neural network repeats this process until the error obtained falls below a predetermined threshold. Finally, the learning process must be validated on the data that has not already been used in the training set. The purpose of the "validation set" is to assess if the neural network has acquired the information to make predictions about new data.

3.1.2 Elements of statistical analysis

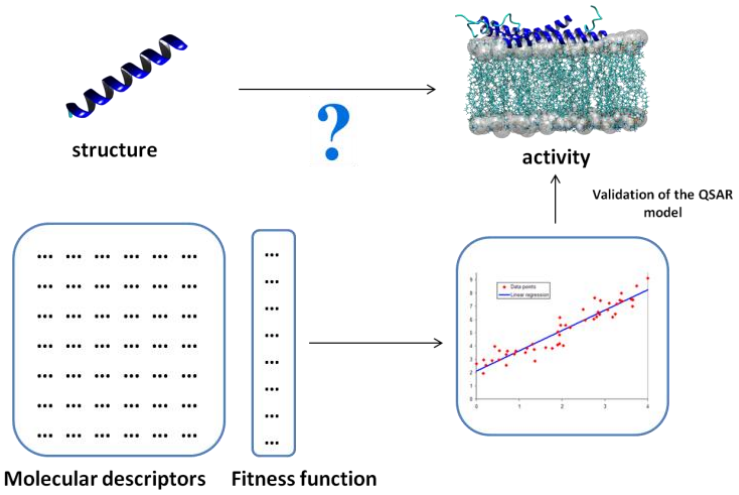


Figure 18 The QSAR analysis allows to correlate the physical-chemical properties of the molecules (molecular descriptors) with their activity. The results are validated by a careful statistical analysis.

In order to evaluate the reliability of the mathematical models obtained from the QSAR analysis, the use of a statistical validation system is indispensable. The validation phase allows to indicate the quality of the build model, including how well it fits the data and the model predictive power. Various statistical measures can be adapted to measure the fitness of mathematical models obtained from the GA analysis during the evolution process (figure 18).

To conduct analyses with genetic algorithms, the Material Studio 7.0 software was used. Firstly, a study table with the AMP homogeneous in their chemical-physical properties (molecular descriptors) was prepared. The fitness function was the MIC of AMP. On the created study tables was applied a function called "Genetic Function Approximation". At the end of each analysis, the system provides a certain number of equations (number chosen by the operator) and a validation table. This table lists a number of parameters, such as Friedman LOF, R-squared, adjusted R-squared, cross validated R-squared, and others. Among these, the Friedman lack-of-fit (LOF) measure and the correlation coefficient (R^2) were considered to estimate the fitness of each model. R^2 is a number between 0 and 1

that gives some information about the goodness of fit of the model. The formula for the calculation of the correlation coefficient is:

$$\boldsymbol{eq.8}\quad \boldsymbol{R^2} = 1 - \frac{SSres}{SStot} = 1 - \frac{\Sigma(y_i - f_i)^2}{\Sigma(y_i - \bar{y})^2}$$

Where:

- **SSres**: the residual sum of squares, a measure of the discrepancy between the data (y_i) and the estimation model (f_i);
- **SStot**: the sum of the squared differences of each observation (y_i) from the mean ($\bar{y}$);

The R^2 value increases as the terms of the equation increase and, therefore, it is not enough to understand if the overfitting phenomenon occurs. For this reason, it is also necessary to consider the value assumed by the LOF parameter. It provides an error measure, estimates the most appropriate number of features and resists to overfitting [69]. In Materials Studio [70], the LOF value is calculated with this formula:

$$\boldsymbol{eq.9}\quad \boldsymbol{LOF} = \frac{SSE}{\left(M\left[\left(\frac{1 - \lambda(c + dp)}{M}\right)\right]^2\right)}$$

Where

- **SSE**: the sum of squares of errors
- **c**: the number of terms in the model
- **d**: a scaled smoothing parameter
- **p**: the total number of descriptors
- **M**: the number of samples in the training set
- **λ**: a safety factor, with the value of 0.99, to ensure that the denominator of the expression can never become zero

Unlike the commonly used least squares measure, the LOF measure cannot always be reduced by adding more terms to the regression model. While the new term may reduce the SSE, it also increases the values of c and p, which tends to

increase the LOF score. In conclusion, a lower LOF value corresponds to a more reliable model.

ANN analysis was performed with the software Matlab 2013 [71] and the performance function for the network is the mean square error (mse). It measures the network's performance according to the mean of squared error that indicates the average quadratic discrepancy between observed data values and estimated data values:

$$\boldsymbol{eq.10} \quad \boldsymbol{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \overline{y_i})^2$$

Where:

- $\mathbf{y_i}$ = observations
- $\boldsymbol{\overline{y_i}}$ = mean

In this work, experimental MIC data were used to perform QSAR analyses. In the statistical validation phase of the results, the experimental MIC values were compared with the predicted MIC values to evaluate if the obtained results were TP (true positive), FP (false positive), TN (true negative) or FN (false negative). We talk about true positive when values are predicted correctly, reflecting the experimental data. The false positives are value incorrectly predicted as positive. With the same criterion, we can define true negative and false negative. Then, the precision (PPV), the accuracy (ACC), the sensitivity (TPR) and the specificity (SPC) were calculated, as defined in Eqs 11-14. Precision is defined as the ratio between the TP and the sum of TP and FP (eq.11). The accuracy is given by the ratio of the sum of TP and TN and the total population (eq.12). Sensitivity is the ratio between the TP and the sum of TP and FN (eq.13). Finally, the specificity is defined as the ratio between the TN and the sum between FP and TN (eq.14).

$$eq.11 \quad PPV = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

$$eq.12 \quad ACC = \frac{\mathrm{TP} + \mathrm{TN}}{\text{total population}}$$

$$eq.13 \quad TPR = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

$$eq.14 \quad SPC = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}$$

The experimental data were extracted from the literature and entered into the Yadamp database. For this reason, an intrinsic experimental error of microbiological tests, due to serial dilutions, had to be considered. The correlation coefficient is significantly affected by the difference between a predicted value and an experimental value. Then, it is more correct to talk about activity classes and the goodness of a QSAR model must be judged in terms of its ability to discriminate among very active, active and non-active peptides. However, the calculation of *precision*, *accuracy*, *specificity* and *sensitivity* is based on a binary predictive response (active peptide/inactive peptide). For this reason, to have a further measure of the reliability of the models obtained, it was calculated an index, called *score*, which evaluates prediction based on the 5 classes of activity of AMP (eq.15).

$$\textbf{eq. 15} \quad Score = \sum_{i=1}^{n} Matrix\left[Class_{observed} - Class_{predicted}\right]$$

	A	B	C	D	E
A	2	1	0	-1	-2
B	1	2	1	0	-1
C	0	1	1	0	-1
D	-1	0	0	1	0
E	-2	-1	-1	0	1

Table 2 Matrix for the computation of the overall model quality.

The scoring matrix in table 2 attributes a reward each time the model correctly predicts the MIC. If the class is not predicted correctly, there is a penalty (negative values).

Appendix B shows the Python script that was created to validate the calculation of statistical parameters, such as *sensitivity*, *specificity*, *accuracy* and *precision*, and for the calculation of the index score (eq.15).

3.1.3 GA and ANN analyses applied on homogeneous subsets of AMP: experimental procedures

The working hypothesis was that antimicrobial peptides with similar features can share the same mechanism of action [64]. The Yadamp database [43] allowed to create subsets of AMP, homogeneous in one, two or more parameters. Parameters, such as peptide length, charge, helicity, flexibility, Boman index and ΔG, were considered. On these homogeneous subsets of AMP, GA and ANN analyses were

performed. AMP with activity against five different microorganisms were studied: *Staphylococcus aureus* and *Bacillus subtilis* (Gram positive bacteria), *Escherichia coli* and *Pseudomonas aeruginosa* (Gram negative bacteria), and *Candida albicans* (saprophytic fungus).

The GA method has been implemented in the Material Studio 7.0 [72] package. In all the analyses, the fitness function was the MIC, the antimicrobial activity of the peptides against the microorganism considered, while the molecular descriptors used to generate the models, were one-dimensional (1D) and two-dimensional (2D) chemical-physical parameters related to the AMP. Nonlinear correlations in the data are explicitly dealt by use the descriptors in spline, quadratic, offset quadratic, and quadratic spline functions. The smoothness parameter was kept at the default value of 1.0 and the length of an equation was let vary between 2 and 5 descriptors. A total of 500 individuals were let evolve over 5000 new generations.

ANN analyses were performed with the software Matlab 2013 [67]. The multilayers network used had two layers: the output and the hidden layer. The hidden layer consisted of ten artificial neurons, the output layer was a single neuron. The training function of the network was the algorithm based on the Levenberg-Marquardt minimization method (trainlm). This function is very fast and performs better on function fitting (nonlinear regression) problems. The adaption learning function was learngdm. It corresponds to the momentum variant of back propagation. The two different transfer functions used for the neurons are: tan–sigmoid transfer function (tansig) for the hidden layer, that returns values between −1 and 1, and linear transfer function (pureline) for the output layer [64].

Chapter IV

4.1 Two mathematical models obtained on AMP active on *S.aureus*

In this section, two of the mathematical models obtained on AMP active on *S.aureus* are shown and discussed. When these analyses were performed, the AMP collected in the Yadamp database with an experimental MIC against *S.aureus,* were 1163. As mentioned previously, the basic hypothesis is that similar peptides have the same mechanism of action. Therefore, homogeneous subsets of peptides were created. Actually, the antimicrobial peptides present in Yadamp with information about the activity against *S.aureus* are 1346.

4.1.1 S.*aureus*: an overview

Staphylococcus aureus is a Gram-positive bacterium and it is a member of the normal flora of the body (nose, respiratory tract, skin) [73]. *S.aureus* causes a lot of clinical infections: minor skin infections (cellulitis, folliculitis, abscesses, etc) or more serious diseases such as bacteremia, infective endocarditis and pleuropulmonary infections [74]. The treatment for *S.aureus* infection is the antibiotic penicillin. When penicillin was introduced for the first time in 1943, it effectively acted in the treatment of *S.aureus* infections. In 1950, the 40% of *S.aureus* cultures were resistant to penicillin; in 1960, this value rose to the 80% [75]. So, it is very important to find drugs that can reduce or even eliminate the Staphylococcal resistance (figure 19).

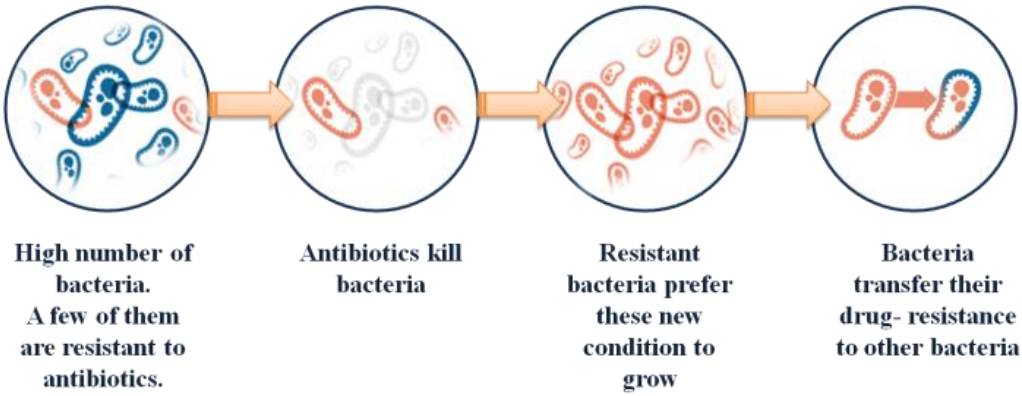


Figure 19 Representation of the way in which a bacterium can become resistant

4.1.2 QSAR Analysis – Dataset A

Genetic Algorithms: results

This model was obtained from a dataset of 55 antimicrobial peptides having a length between 7 and 11 amino acids (dataset A). The molecular descriptors used in this analysis are 45 molecular descriptors 1D and 2D. Among these there are the length, the charge at pH 5, 7 and 9, the helicity, the Boman index, etc. The equation obtained from this analysis shows that, for this homogeneous dataset, the most important parameters for AMP killing action on *S.aureus* are the peptide charge at pH 5 and pH 7 and the number of polar amino acids in the sequence (eq.16).

$$\boldsymbol{eq.16}\quad \boldsymbol{MIC} = 8.16 * \boldsymbol{POLAR\_AA} - 2571(-0.72 - \boldsymbol{Ch5})^2 + 9963(-0.90 - \boldsymbol{Ch7})^2 + 11$$

Where

- **Ch5**: peptide charge at pH5
- **Ch7**: peptide charge at pH7
- **POLAR_AA**: number of polar residues

This result is in line with several studies that have shown that the electrostatic interaction between AMP and lipid membranes is fundamental for their mechanism of action [76]. For example, a recent study has developed a model for linear antimicrobial peptides activity (magainin 2 amide and melittin), based on the effects of both lipid-peptide charge and topographical interactions. These researchers have shown that antimicrobial activity is governed by topological and electrostatic interactions between the membrane-bound peptide and the surrounding lipids [77]. As described in the previous section, the reliability of the results obtained by the QSAR analyses is evaluated by considering the values of R^2 and LOF. In this case, the analysis generated R^2 and LOF values of 0.92 and 738, respectively.

Genetic Algorithms: statistical validation

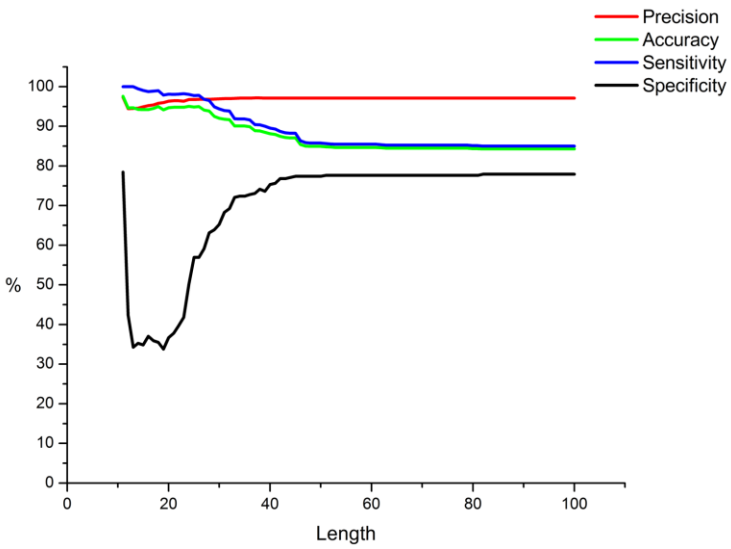


Figure 20 Result of the statistical validation of the GA analysis performed on the dataset of 55 antimicrobial peptides active on *S.aureus* with a length ranging from 7 to 11 amino acids.

The calculation of precision (PPV), accuracy (ACC), sensitivity (TPR) and specificity (SPC) indexes requires an arbitrary definition of what is considered active and inactive (see section 3.1.2). In the figure 20, the *precision*, *accuracy*, *sensitivity* and *specificity* parameters calculated for the model obtained by GA

analysis (eq.16), are shown. For this model, the behavior is acceptable only for three indexes: *specificity* (black lines in the figure) is the exception, with values that drop to 35 % for peptide with a length between 11 and 20 amino acids. Low *specificity* indicates that models displays many false positives. However, a good correlation coefficient and high values of *precision*, *accuracy* and *sensitivity*, cannot capture the quality of an activity model because the intrinsic experimental error in microbiological tests, due to serial dilutions, is not considered. It is more correct to talk about activity classes. A common view in the pharma industry was adopted to consider inactive those peptides with a MIC higher than 30 μM (table 2). The overall quality of the model (*score*) is calculated comparing MIC predictions with the experimental data according to eq.15.

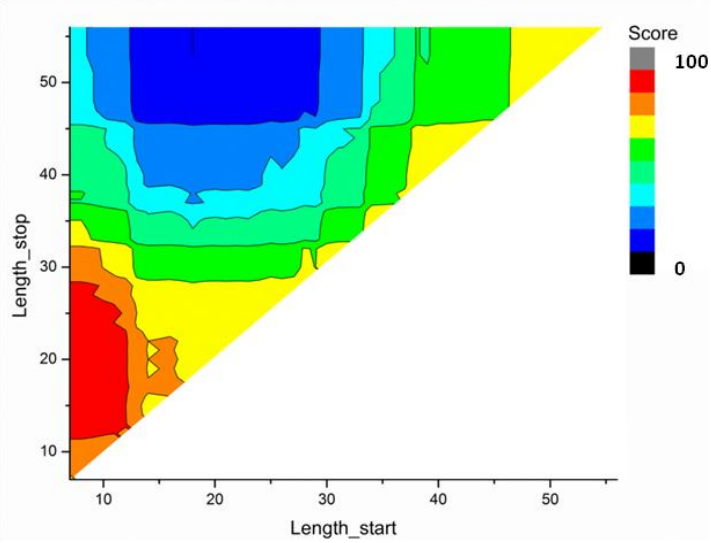


Figure 21 Application of the *score* function on the results obtained through GA analysis on the dataset of 55 antimicrobial peptides active on *S.aureus* with a length ranging from 7 to 11 amino acids. The red areas represent the areas where the model is reliable. The blue zones are the areas where the model is unreliable.

This diagram (figure 21) permits to easily evaluate the domain of applicability of the model. Each point in the figure corresponds to a set of peptides of length between Length_start and Length_stop. The overall quality, calculated with eq.15, is rescaled between 0 (blue, unreliable) and 100 (red, reliable), and color mapped.

For example, the point 20, 50 of the figure 21 indicates that the sum of the scores on all peptides with length between 20 and 50 is lower the 10 % (blue region). At the same time, the figure shows that peptides with a length less then 30 amino acids have a high score sum. This indicates that the reliable region (red) is larger than the subset where the model was calculated (AMP with a length between 7 and 11 amino acids). For longer peptides, the prediction capability of the model quickly degrade.

Artificial Neural Networks: results

On the same data set (dataset A), ANN were applied. The neural network used consisted of 2 layers with 10 neurons in the hidden layer.

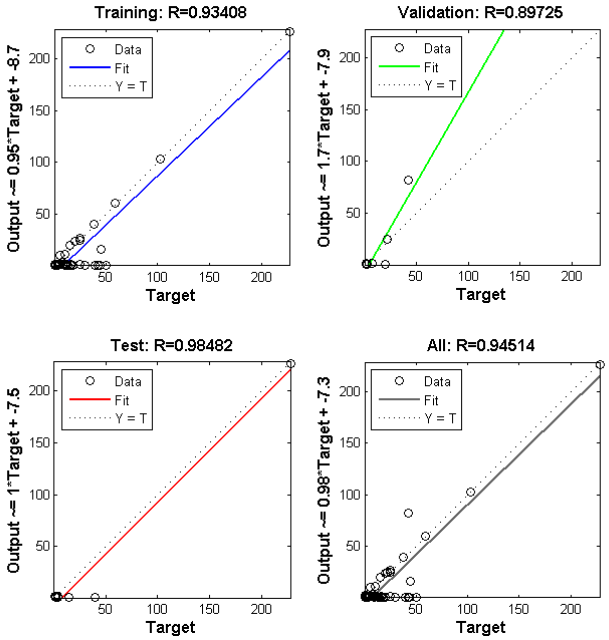


Figure 22 Results of the application of ANN for peptides with a length between 7 and 11 amino acids.

The result of the neural network is shown in the figure 22. The final correlation coefficient is 0.94. The ANN, as well as the GA analysis, have been able to learn the existing correlations between the molecular descriptors and the antimicrobial activity (training phase). In this case, however, we do not get a mathematical

equation from the analyses and, therefore, it is not possible to determine what are the important parameters for AMP mechanism of action. Therefore, the analyses with GA and ANN are strongly complementary.

Artificial Neural Networks: statistical validation

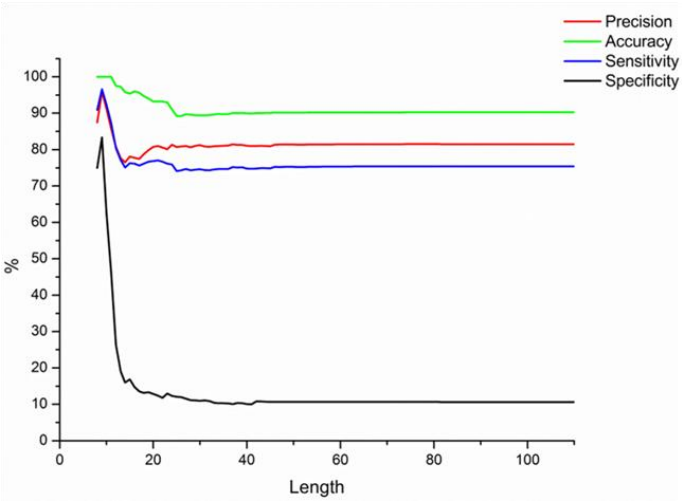


Figure 23 Result of the statistical validation of the ANN analysis performed on the dataset of 55 antimicrobial peptides active on *S.aureus* with a length ranging from 7 to 11 amino acids.

The evaluation of the applicability of the neural network models were performed in the same way of GA models. In the figure 23, the trend of *sensitivity, specificity, accuracy and precision* for active and inactive peptides, were reported. Even in this case, the behavior is acceptable only for the *sensitivity*, *accuracy* and *precision* indexes. *Specificity* is low for peptides with a length greater than 15 amino acids.

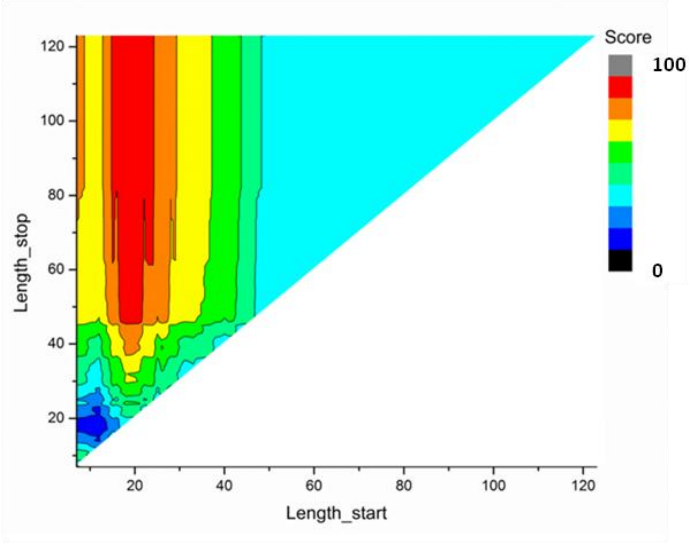


Figure 24 Application of the *score* function on the results obtained from the ANN analysis on the dataset of 55 antimicrobial peptides active on *S.aureus* with a length ranging from 7 to 11 amino acids. The red areas represent the areas where the model is reliable. The blue zones are the areas where the model is unreliable.

A more accurate evaluation is shown in the figure 24. The graph was constructed using the *score* calculated through the eq.15. In this chart we can clearly see that the range of peptides on which the model of activity obtained with neural networks is applicable is different than the range that was obtained with genetic algorithms. In fact, the ANN model is applicable in a range of peptides longer than 40 amino acids.

This means that the two activity models (one with GA analyses and the other with ANN analyses) performed on the same dataset, are applicable on different sets of peptides.

4.1.3 QSAR Analysis – Dataset B

Genetic Algorithms: results

This model was obtained from a dataset of AMP shorter than 30 amino acids, with a Boman index[1] between 1 and 2 kcal/mol (dataset B). Also in this case, 45 1D and 2D molecular descriptors were used.

$$\boldsymbol{eq.17}\quad \boldsymbol{MIC} = -\frac{(\boldsymbol{MW} - 881)^2}{250000} + 122 * (\boldsymbol{D} - 1.7)^2 + 3134 * (1.07 - \boldsymbol{Ch5})^2 - 3340 * (0.79 - \boldsymbol{Ch7})^2 + 22$$

Where

- **D**: number of residues of Aspartic acid
- **Ch5**: peptide charge at pH5
- **Ch7**: peptide charge at pH7
- **MW**: molecular weight

The R^2 obtained from this analysis is 0.81 while the LOF value is 876.01. This equation (eq.17), such as the first model obtained (eq.16), considers the *charge* parameter as an important molecular descriptor for the mechanism of action of AMP. In addition, the presence of amino acid *aspartate* in the sequence and the *molecular weight*, appear other factors that affect AMP activity.

Genetic Algorithms: statistical validation

In order to proceed with the validation, the equation obtained were applied to all the peptides with an experimental MIC value against *S.aureus*, collected in Yadamp. The values of *precision*, *accuracy*, *sensitivity* and *specificity* were calculated in agreement with the formulas in the paragraph 3.1.2.

[1] The Boman index is the sum of the free energies of the respective side chains for transfer from cyclohexane to water.

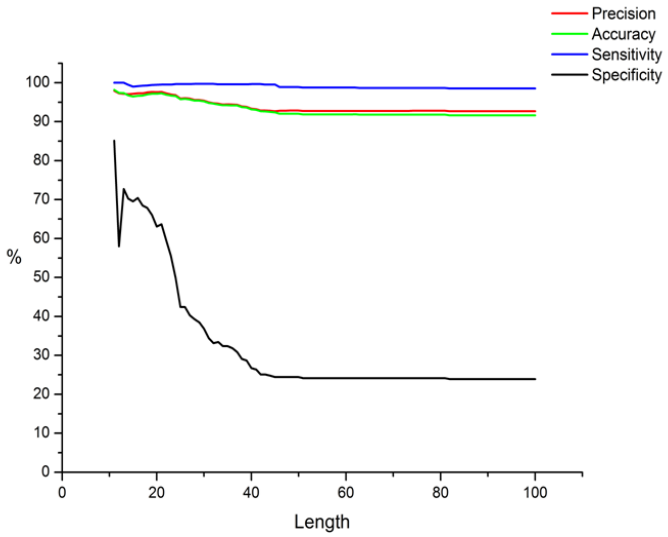


Figure 25 Result of the statistical validation of the GA analysis performed on the dataset of 92 antimicrobial peptides active on *S.aureus* with a length shorter than 30 amino acids and a Boman index between 1 and 2 kcal/mol.

Also in this case, the parameter *specificity* is an exception (black line in figure). The value of the *sensitivity* is very high (about 99-100%), as well as *precision* and *accuracy* (98-95%). However, in the range in which the model was built (length ≤ 30 amino acids), the *specificity* decreases to 30-40%. This chart, however, only reports length information and not the Boman index (the other parameter on which the model was built). For this reason, probably the MIC of peptides with a length less then 30 amino acids but with a Boman index not in the range in which the pattern was generated, is not correctly predicted. This aspect is reflected in a reduction of the *specificity* value.

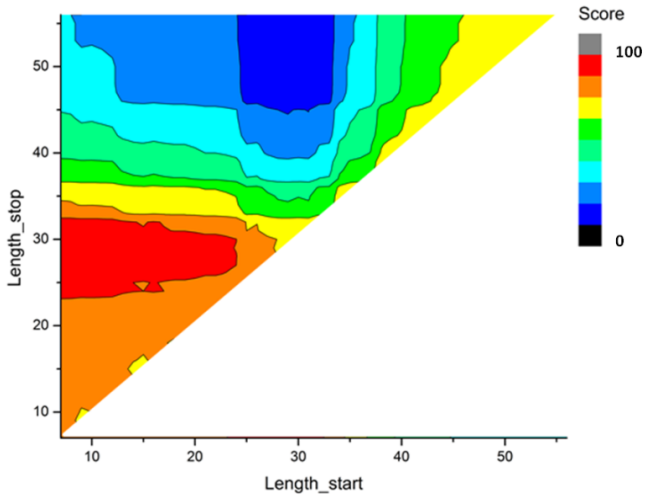


Figure 26 Application of the *score* function on the results obtained from the GA analysis on the dataset of 92 antimicrobial peptides active on *S.aureus* with a length shorter than 30 amino acids and a Boman index between 1 and 2 kcal/mol. The red areas represent the areas where the model is reliable. The blue zones are the areas where the model is unreliable.

The procedure is the same of the first model obtained. A chart based on the score calculation was prepared (figure 26), using the eq.15. The score is very high for peptides with a length longer than 20 amino acids and less than 35 amino acids (red zone). Therefore, this model is applicable to the peptides on which it was built.

Artificial Neural Networks: results

The ANN analyses performed on this dataset of AMP with a length less than 30 amino acids and a Boman index between 1 and 2 kcal/mol (dataset B), did not give satisfactory results. In fact, the total correlation coefficient is 0.43 (figure 27).

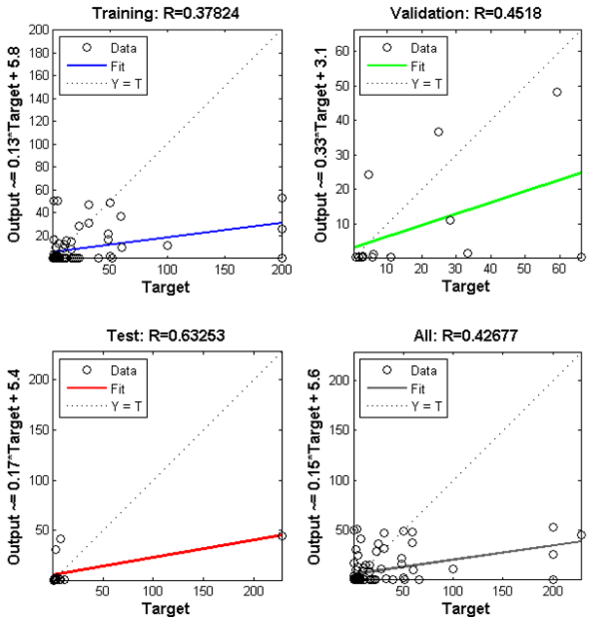


Figure 27 Results of the application of ANN on the dataset of 92 antimicrobial peptides active on *S.aureus* with a length shorter than 30 amino acids and a Boman index between 1 and 2 kcal/mol.

Therefore, in this case, genetic algorithms were more efficient than the ANN.

Artificial Neural Networks: statistical validation

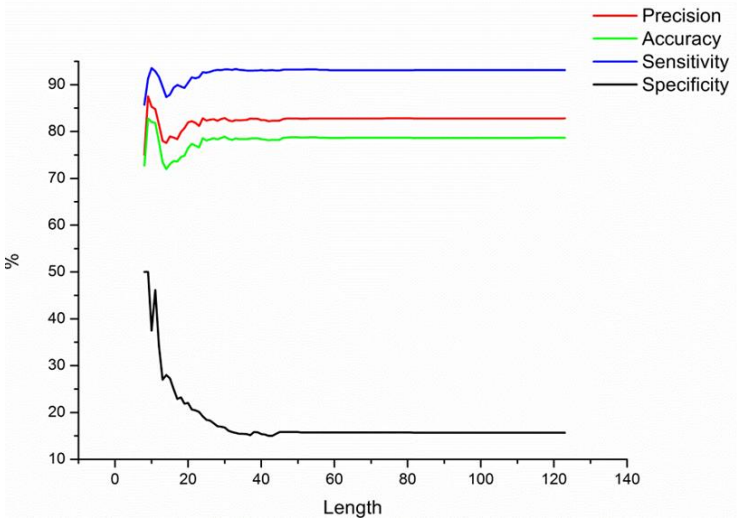


Figure 28 Result of the statistical validation of the ANN analysis performed on the dataset of 92 antimicrobial peptides active on *S.aureus* with a length shorter than 30 amino acids and a Boman index between 1 and 2 kcal/mol.

Also in this case, it was performed a statistical validation of the results obtained through ANN analyses. The *accuracy*, the *precision* and the *sensitivity* values are very high, while the *specificity* is rather low. The graph in figure 28 shows that the model can only be applied on AMP with features similar to those on which it was built.

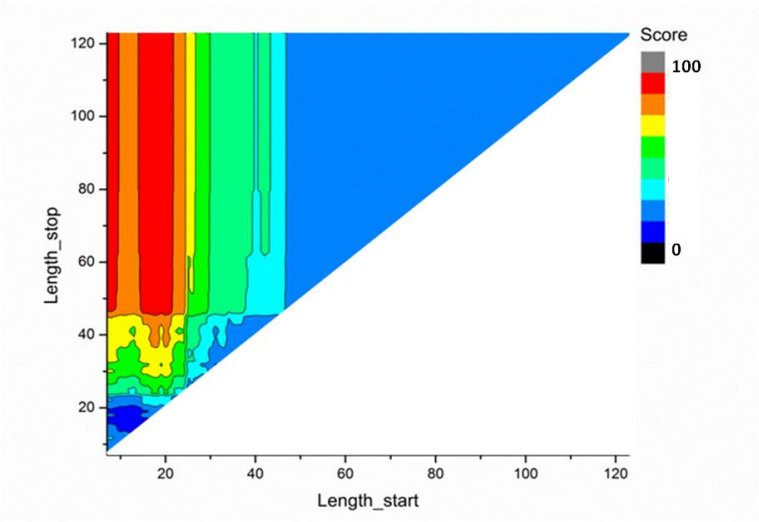


Figure 29 Application of the *score* function on the results obtained from the ANN analysis on the dataset of 92 antimicrobial peptides active on *S.aureus* with a length shorter than 30 amino acids and a Boman index between 1 and 2 kcal/mol. The red areas represent the areas where the model is reliable. The blue zones are the areas where the model is unreliable.

The score evaluation (figure 29), based on the 5 classes of activity in which antimicrobial peptides were divided, shows that the ANN models are applicable in a range of peptides narrower than ranges obtained for GA models.

4.2 Limits of GA and ANN analysis: GMDH

To reveal a structure-activity relationship of a compound it is need an effort from computational chemists, biologists and bioinformatics [78]. The father of the concept of quantitative structure-activity relationship (QSAR) was Corwin Hansch, who implemented this technique more than 50 years ago. Actually, QSAR is one of the most commonly used approaches in academy, industry, and government institutions, to modeling thousands of different molecular structures using a wide variety of statistical and machine learning techniques. This is also evidenced by the numerous articles published in the past decade [79-83]. QSAR analyses have a strong impact on human health and ecological systems [35]. The publications in this field, as shown in the literature, is directly proportional with the continuous discovery of chemical data and the development of new databases. However, it is not easy to get the optimal results through the QSAR analysis, due to the preparation of the data and to poor application of statistical methods. First, an optimal QSAR analysis needs an adequately sized data set. At the same time, the choice of molecular descriptors is a key point. In fact, if two descriptors are highly similar and they give the same information, it is not possible a correct statistical association and, probably, we have a false improvement of the predictive power of the QSAR analysis. The same thing can happen if descriptors are not clearly defined or contain errors. In addition, the assessment of model applicability domain is very important. It is defined as "the response and chemical structure space in which the model makes predictions with a given reliability" [84]. Many data for QSAR analysis originate from the literature and this aspect can lead to the increase in outliers. Outliers are values that do not fit the model [80]. The use of a large number of molecular descriptors also can make the creation of the model more difficult. Furthermore, descriptor values have different numerical ranges. To determine the contribution of each descriptor to the QSAR analysis it is need an auto scaling. In fact, large numerical values can dominate the model, compromising its statistical validity. Other problems, such as heterogeneity, inappropriate units, incorrect chemical names or structures, can

affect the performance of a QSAR analysis. The right choice of all these parameters and the algorithm to use for prediction analysis depends primarily on the problem to be resolved. QSAR requires a priori assumptions about the laws governing the data and their properties. Group method of data handling (GMDH) is an inductive approach, based on the principle of the self-organization [85]. It is an heuristic self-organization method. GMDH is an appropriate modelling procedure when it is not easy to define an input-output relationship in a complex system [86].

This method consists of several points [87]:

1. Selection of a series of descriptors that are important to the problem;

2. Division of the observations into two groups: the first is used by the system to learn (training set) and the second to estimate the values (validation set);

3. Through an iterative procedure creates a number of elementary functions with increasing complexity, producing different models;

4. Choice of the optimum model;

One of the advantages of GMDH is the possibility to automatically have a statistical validation. For example, a classification analysis performed with GMDH, provides a confusion matrix and a ROC curve. So, the GMDH approach was used to create new and more effective activity models which have been implemented in the “Yadamp predict” tool.

4.2.1 GMDH: learning algorithms

GMDH algorithms differ for the type of elementary function applied, the complexity of the model and the external criteria applied to the data set. The choice of the algorithm depends on the type of the analysis and on the data available [85]. There are two closely related learning algorithms available in GMDH:

- **Combinatorial GMDH (COMBI)**

- **GMDH-type neural networks**

For both algorithms, model coefficients are fitted using the least squares method[2]. These algorithms generate models from simple to complex until the accuracy of the test increases. Validation strategies consist of data partition in training and testing sets. The training dataset is used to fit the model coefficients, while the testing dataset is used to calculate a validation measure. At the end of each analysis, GMDH displays the performance results to evaluate the success of the modeling simulation.

- **Combinatorial GMDH (COMBI)**

This model is the basic GMDH algorithm. It produces a linear polynomial function, generated from a given set of variables. The combinatorial GMDH algorithm use a matrix of input data sample, containing N points of observations over a set of M variables. In GMDH we can choose different validation criteria. Normally, a data sample is divided into two parts: two-thirds of observations form

[2] The method of least squares is an optimization technique (or regression) that allows to find a function, represented by an optimal curve (or regression curve), which is as close as possible to a data set (typically points of the plan). In particular, the function found must be that which minimizes the sum of the squares of the distances between the observed data and those of the curve that represents the function.

the training set and the remaining part of observations forms the test set. The training set is used to estimate coefficients of the polynomial function using Least squares method and to choose the optimal model. An alternative is, for example, the use of the cross-validation criterion that take into account all information in data sample. Each point successively is taken as test set and then averaged value of criteria is used. Combinatorial GMDH chooses the best model (set of models) indicated by minimal value of the criterion.

- **GMDH-type neural networks**

GMDH-type neural networks employs combinatorial algorithm to optimize the neuron connection. The algorithm iteratively creates layers of neurons with two or more inputs. Every new layer is created using two or more neurons taken from any of previous layers. Every neuron in the network applies a transfer function (quadratic or linear) to choose a final transfer function that predicts testing data most accurately (see section 3.1.1).

GMDH can produce predictions that allow to solve *classification* and *regression* problems. *Classification* is the prediction of a category of unknown instance. The data must be in text format. *Linear regression analysis* is used to estimate the parameters.

4.3 GMDH Analyses

Classification and regression analyses were performed using the GMDH software. Some activity models were obtained for peptides active on *S.aureus* (see paragraph 4.1.1), *E.coli* and erythrocytes. The obtained results were listed in the following paragraphs.

E.coli: an overview

Escherichia coli is a Gram-negative bacterium commonly found in the lower intestine of warm-blooded organisms (endothermic) [88]. Most *E.coli* strains are harmless and are part of the normal flora of the intestine, but some serotypes can cause severe food poisoning in their hosts and occasionally are responsible for product recalls due to contamination of foods [89].

4.3.1 *E.coli*: Dataset C

Classification Analysis: results

A dataset of 63 antimicrobial peptides active on *E.coli* (dataset C) was submitted to perform the classification analyses. The classification analysis requires that the MIC values, which represent the fitness function of these analyses, are expressed in text format. The peptides with MIC values lower than 30 µM were called "active" and the peptides with the MIC values greater than 30 µM were called "inactive". GMDH allows to set up an experimenter's layout, in which we can set the parameters to use (figure 30).

In this analysis, the following parameters were set:

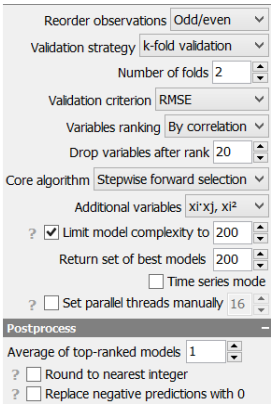


Figure 30 Screenshot of the experimenter's layout used for the classification analysis performed on 63 AMP active on *E.coli.*

The function "reorder observations" is used to give uniform statistical characteristics of training and testing tests and to make them equally informative. In this case, odd and even raws were used in the creation of training and validation sets. The "validation strategy" allows the choice of the model validation. Here, the "k-fold validation" option was chosen. It splits dataset into k parts (2 in this case). It trains a model k times using k-1 parts, each time measuring the model performance using a new remaining part. Finally, the residuals obtained from all testing parts were used for model comparison. The "validation criterion" defines the model selection criterion for the core

algorithm and the variables ranking. In this case, the variables were ranked according to the correlation values. The “core algorithm” chosen in this analysis provides a forward selection approach. This approach tests the addition of each variable using a chosen model fit criterion and repeats this process until none improves the model to a statistically significant extent. The core algorithm and the variables ranking are validated by the root-mean-square error (RMSE) criterion (eq.18).

$$\boldsymbol{eq.18}\quad \boldsymbol{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)}{n}}$$

Where:

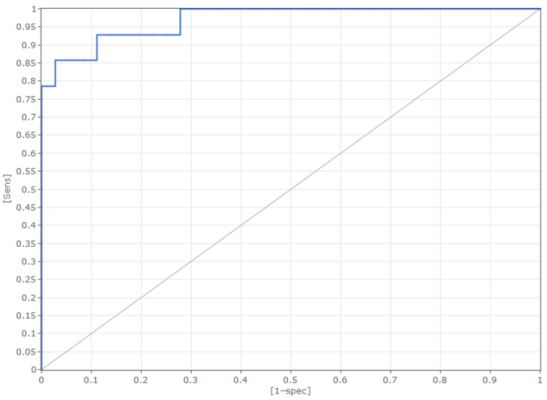
$\hat{\boldsymbol{y}}_{\boldsymbol{i}}$ = predicted values for observations *i*

$\boldsymbol{y_i}$ = regression's dependent variable

$\boldsymbol{n}$ = different predictions

The voice "additional variables" expands the dataset with new artificial features to improve the classification model. In this case, the function $\boldsymbol{x_i * x_j, x_i^2}$, that adds all the possible multiple couples and squares, was chosen. Any model may consist of not more than **n** terms. In this analysis, the model complexity was limited to 200 terms. To calculate the performance of the predictions, the 20% of the dataset was validated.

		Predicted class		
		active	inactive	total
	active	36	0	36
	inactive	3	11	14
Actual class	total	39	11	50
	precision	0,923	1	
	accuracy	0,94	0,94	
Correctly classified instances		47	94%	
incorrectly classified instances		3	6%	
RMSE		0,237		



		Predicted class		
		active	inactive	total
	active	9	0	9
	inactive	3	1	4
Actual class	total	12	1	13
	precision	0,75	1	
	accuracy	0,769	0,769	
Correctly classified instances		10	77%	
incorrectly classified instances		3	23%	
RMSE		0,421		

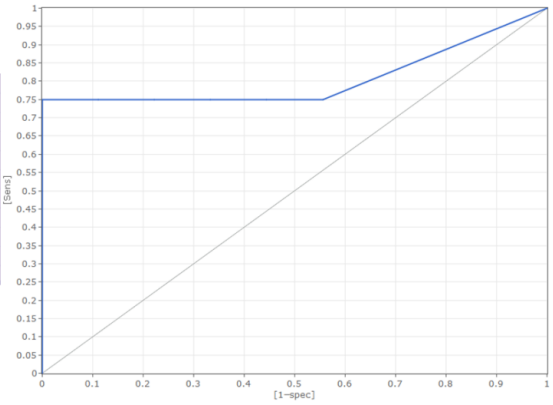


Figure 31 Results of the classification analysis on 63 AMP active on *E.coli*. At the top, there are the confusion matrix and the ROC curve calculated for observations used to train the model. It is equal to 0.97. In the chart below, there is the ROC calculated for the withheld observations. It is equal to 0.81. To the left of the graphs are shown the confusion matrix.

Classification analysis generates a ROC curve (receiver operating characteristic curve). It is a graphical plot that illustrates the ability of a binary classifier system. The ROC curve is the plot of the sensitivity against the values of 1 − specificity. The analysis of the ROC curve represents a method for comparing two continuous distributions and it is based on the estimation of an index, the area under the curve (A). It expresses the probability of a model to identify true positive and true negatives in a system.

In this case, the ROC calculated for observations used to create the model is equal to 0.97, while the ROC calculated for the withheld observations is equal to 0.81 (figure 31).

The equation of the model obtained is the following:

$$\boldsymbol{eq.19}\quad \boldsymbol{y} = -2.32 + (\boldsymbol{a} * \boldsymbol{b} * 6.17) + \big((\boldsymbol{Ch5}) * \boldsymbol{c} * (-0.05)\big) + ((\boldsymbol{d})^{\boldsymbol{2}} * 6.88) + \big(\boldsymbol{e} * \boldsymbol{b} * (-10.35)\big) + \big((\boldsymbol{\Delta Charge}) * \boldsymbol{e} * 0.66\big) + ((\boldsymbol{Ch5}) * \boldsymbol{a} * (-0.23) + (\boldsymbol{f} * \boldsymbol{g} * 0.52) + \big(\boldsymbol{a} * \boldsymbol{h} * (-6.25)\big) + \big(\boldsymbol{g} * \boldsymbol{a} * (-0.42)\big) + (\boldsymbol{f} * \boldsymbol{i} * 1.82) + ((\boldsymbol{l})^{\boldsymbol{2}} * 6.00)$$

Where

a= normalized composition from fungi and plant [90]

b= influence of Water on Protein Structure [91]

c= hydrophobicity coefficient [92]

d= average flexibility indices [93]

e= charge transfer capability [94]

f= steric parameter [95]

g= secondary structure stability [96]

h-l= prediction factors of the secondary structure of globular proteins [97]

i= normalized positional residue frequency at helix termini [98]

Ch5= peptide charge at pH5

ΔCharge= difference between the charge of the peptide at pH9 and pH5

Regression Analysis: results

On the same dataset of AMP active on *E.coli* (dataset C) was performed a regression analysis. Also in this case, experimental criteria were set (figure 32).

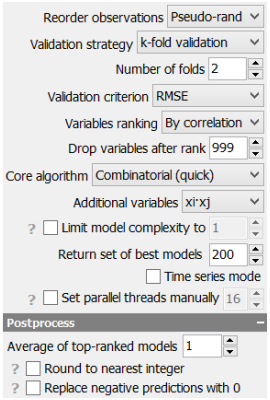


Figure 32 Screenshot of the experimenter's layout used for the regression analysis performed on 63 AMP active on *E.coli*

In this case, it was chosen the option that permits a *pseudo-random* creation of the training and the validation sets. A combinatorial algorithm (section 4.2.1) and the function $\boldsymbol{x_i} * \boldsymbol{x_j}$ that adds all possible multiplied pairs, were chosen. The model complexity was limited to 200 terms. The 10% of the dataset was used to evaluate the performance of the prediction. The result of the prediction is shown in the graph in figure 33. The coefficient of determination (R^2) calculated for the observations used to create the model is equal to 0.63, while the R^2 calculated for the withheld observations is equal to 0.75.

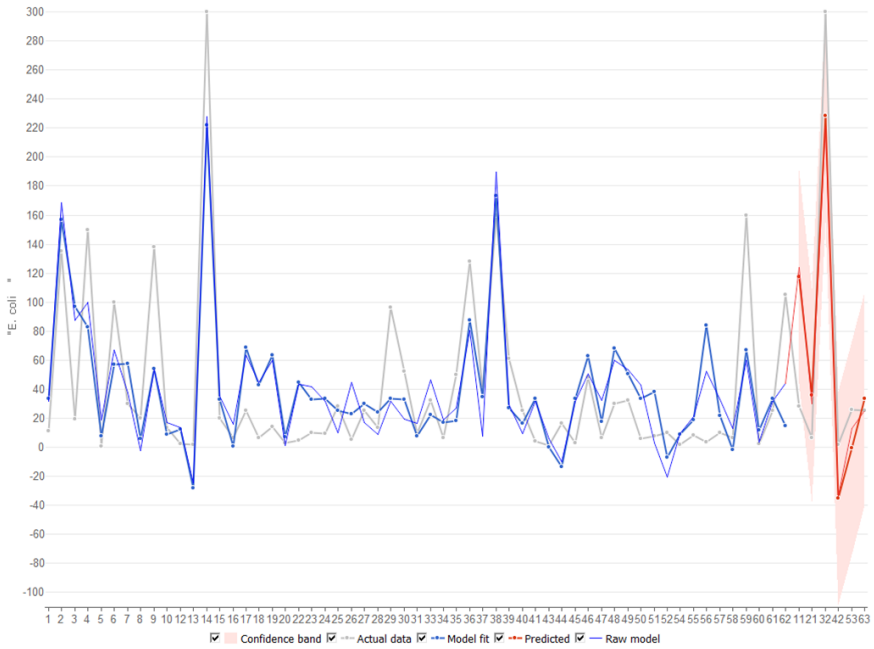


Figure 33 Performance of the model obtained. On the y axis there are the MIC values. On the x axis there are the AMP sequences. The gray line represents the MIC values for each of the 63 peptides in the dataset. The blue line represents how the model obtained (the equation) learned from the peptides of the training set. The red line represents how much the equation can be applied to the validation set (not used in the learning phase). The coefficient of determination (R^2) calculated for the observations used to create the model (training set) is equal to 0.63, while the R^2 calculated for the withheld observations (validation set) is equal to 0.75.

The equation of the model obtained is the following:

$$\boldsymbol{eq.20}\quad \boldsymbol{y} = 31.71 + \left(\boldsymbol{a} * \boldsymbol{b} * (-6065.61)\right) + \left(\boldsymbol{a} * \boldsymbol{c} * (-18.33)\right) + \left(\boldsymbol{a} * \boldsymbol{d} * 354.84\right) + \left(\boldsymbol{a} * \boldsymbol{e} * 3652.88\right) + \left(\boldsymbol{a} * \boldsymbol{f} * (-275.41)\right) + \left(\boldsymbol{a} * \boldsymbol{g} * 801.77\right) + \left(\boldsymbol{a} * \boldsymbol{h} * 2178.82\right) + \left(\boldsymbol{a} * \boldsymbol{i} * 9591.25\right) + \left(\boldsymbol{a} * \boldsymbol{l} * (-8013.42)\right) + \left(\boldsymbol{a} * \boldsymbol{m} * (-2001.28)\right)$$

Where

a= charge transfer capability [94]

b= effect of protein size on the hydrophobic behavior of amino acids [99]

c= parameter that considers the different amino acids composition between the cytoplasmic and extracellular sides in membrane proteins [100]

d= optimized side chain interaction parameter [101]

e= normalized frequency of alpha-helix in peptides [102]

f= normalized frequency of turn in all-alpha class [102]

g= normalized frequency of turn in alpha/beta class [102]

h- i- l- m= prediction factors of the secondary structure of globular proteins [97]

4.3.2 *E.coli*: Dataset D

Classification Analysis: results

A classification analysis was performed on a dataset of 62 AMP active on *E.coli* (dataset D), where the peptides with a MIC lower than 30 µM were called "active" and the peptides with a MIC greater than 30 µM were called "inactive". The experimenter's layout is shown in the figure 34.

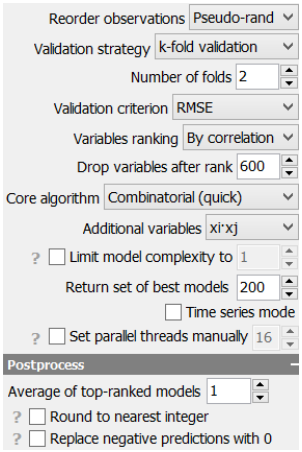


Figure 34 Screenshot of the experimenter's layout used for the classification analysis performed on 62 AMP active on *E.coli*

In this analysis, unlike the previous classification analysis, the *pseudo-random* raws were chosen for the creation of the training and the validation sets. Also in this case, the dataset was divided into 2 parts for the validation of the model (k fold validation). Variables were ranked according to the correlation values. The core algorithm chosen in this analysis is a combinatorial algorithm (section 4.2.1). The core algorithm and the variables ranking were validated by the root-mean-square error (RMSE) criterion (eq.18). At the voice "additional variables", the function $\boldsymbol{x_i} * \boldsymbol{x_j}$ was chosen. This function, in the creation of the model, adds all possible multiplied pairs. The model complexity was limited to 200 terms and the 30% of the dataset was chosen to validate the model. "Drop variables after rank"

is another function of the experimenter's layout that permits to reduce the number of the variables to **n** most important variables according to the selected ranking algorithm. Preliminary reduction of variables may reduce the quality of models, but it is definitely useful for quicker processing of high-dimensional datasets. In this case, the maximum number of variables was reduced to 600. In this analysis, the ROC curve calculated for observations used to create the model is equal to 0.75 while the ROC calculated for the withheld observations is equal to 0.83 (figure 35). The equation of the model obtained is the following:

$$\boldsymbol{eq.21}\quad \boldsymbol{y} = -1.23 + (\boldsymbol{a} * \boldsymbol{b} * 2.51) + (\boldsymbol{a} * \boldsymbol{c} * 2.86) + (\boldsymbol{d} * \boldsymbol{e} * 0.003) + (\boldsymbol{d} * \boldsymbol{f} * 0.005) + (\boldsymbol{d} * \boldsymbol{g}) * (-0.00072) + (\boldsymbol{d} * \boldsymbol{h} * 0.000749)$$

Where

a= Direction of hydrophobic moment [103]

b= Hydrophobicity index [104]

c= Amino acid side-chain partition energies [105]

d= Molecular weight [106]

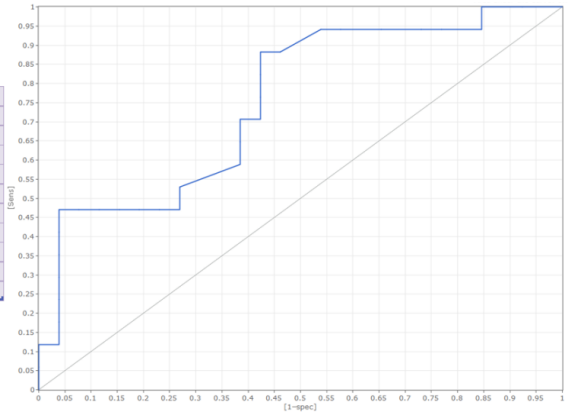
e= Alpha-helix indices for beta-proteins [107]

f= Composition [108]

g= Hydrophilicity value [109]

h= Relative mutability [110]

		Predicted class		
		active	inactive	total
	active	22	4	26
	inactive	9	8	17
Actual class	total	31	12	43
	precision	0,71	0,667	
	accuracy	0,698	0,698	
Correctly classified instances		30	70%	
incorrectly classified instances		13	30%	
RMSE		0,537		



		Predicted class		
		active	inactive	total
	active	15	1	16
	inactive	1	2	3
Actual class	total	16	3	19
	precision	0,938	0,667	
	accuracy	0,895	0,895	
Correctly classified instances		17	89,5%	
incorrectly classified instances		2	10,5%	
RMSE		0,324		

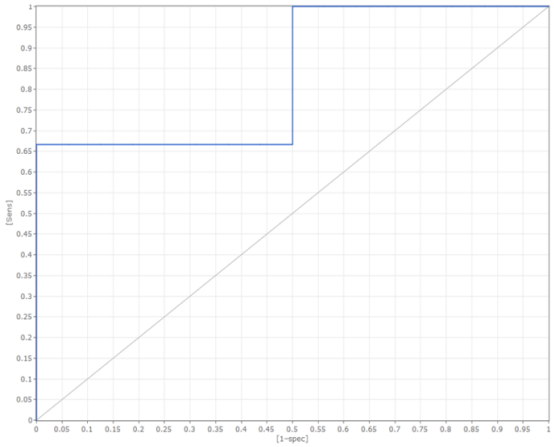


Figure 35 Results of the classification analysis on 62 AMP active on *E.coli*. At the top, there is the ROC curve calculated for observations used to the model. It is equal to 0.75. In the chart below, there is the ROC calculated for the withheld observations. It is equal to 0.83. To the left of the graphs are shown the confusion matrix.

Regression Analysis: results

A regression analysis was performed on the same dataset of 62 AMP active on *E.coli* (Dataset D). Also in this case, some experimental criteria were set (figure 36).

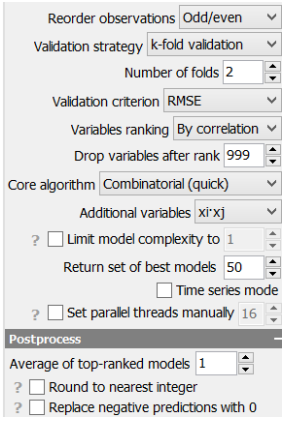


Figure 36 Screenshot of the experimenter's layout used for the regression analysis performed on 62 AMP active on *E.coli*

This time, odd and even rows were chosen for the creation of the training and the validation sets. The dataset was divided into 2 parts for the training and the validation phases and it was chosen a combinatorial algorithm (section 4.2.1) to create the model. The core algorithm and the variables ranking (by correlation) are validated by the root-mean-square error (RMSE) criterion (eq.18). At the voice "additional variables", the function $\boldsymbol{x_i} * \boldsymbol{x_j}$ was chosen. This function adds all the possible multiplied pairs in the creation of the model. The 10% of the dataset was chosen to evaluate the performance of the prediction.The coefficient of determination (R^2) calculated for the observations used to create the model is equal to 0.69, while the R^2 calculated for the withheld observations is equal to 0.81 (figure 37).

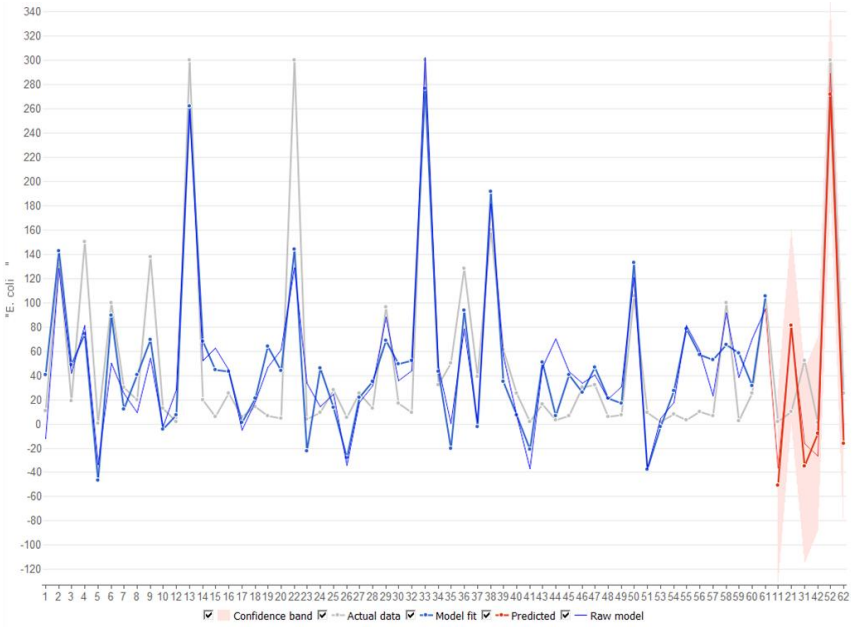


Figure 37 Performance of the model obtained. On the y axis there are the MIC values. On the x axis there are the AMP sequences. The gray line represents the MIC values for each of the 62 peptides in the dataset. The blue line represents how the model obtained (the equation) learned from the peptides of the training set. The red line represents how much the equation can be applied to the validation set (not used in the learning phase). The coefficient of determination (R^2) calculated for the observations used to create the model (training set) is equal to 0.69, while the R^2 calculated for the withheld observations (validation set) is equal to 0.81.

The equation of the model obtained is the following:

$$\boldsymbol{eq.22}\quad \boldsymbol{y} = 142.18 + (\boldsymbol{a} * \boldsymbol{b} * 809.64) + (a * \boldsymbol{c}) * (-4563.92) + (\boldsymbol{a} * \boldsymbol{d} * 329) + \left(\boldsymbol{a} * \boldsymbol{e} * (-1402.55)\right) + \left(\boldsymbol{a} * \boldsymbol{f} * (-4558.16)\right) + (\boldsymbol{a} * \boldsymbol{g} * 698.3) + \left(\boldsymbol{a} * \boldsymbol{h} * (-106.008)\right) + (\boldsymbol{a} * \boldsymbol{i} * 4572.37) + \left(\boldsymbol{a} * \boldsymbol{l} * (-560.07)\right) + (\boldsymbol{a} * \boldsymbol{m} * 6776.05) + \left(\boldsymbol{a} * \boldsymbol{n} * (-3646.4)\right) + \left(\boldsymbol{a} * \boldsymbol{o} * (-1208.86)\right) + \left(\boldsymbol{p} * \boldsymbol{b} * (-21.18)\right)$$

Where

a= charge transfer capability [94]

b= retention coefficient [111]

c= effect of protein size on the hydrophobic behavior of amino acids [99]

d= hydrophobicity of amino acid composition of mitochondrial proteins [90]

e= surface-interior diagram of globular proteins [112]

f= transfer energy in organic solvent/water [113]

g= optimized transfer energy parameter [101]

h= optimized side chain interaction parameter [101]

i= normalized frequency of alpha-helix in peptides [102]

l = normalized frequency of turn in peptides [102]

m= normalized frequency of alpha-helix in all-alpha class [102]

n= normalized frequency of alpha-helix in alpha+beta class [102]

o= normalized frequency of alpha-helix in alpha/beta class [102]

p= normalized frequency of N-terminal helix [102]

4.3.3 *E.coli*: Dataset E

Classification Analysis: results

In this classification analysis of 56 AMP active on *E.coli* (Dataset E), it was used the same experimental protocol of the classification analysis performed on the 62 AMP active on *E.coli* (section 4.3.2). The only difference is that, in this case, no additional variables were used (figure 38). Also in this case, the 30% of the dataset was used for the validation phase.

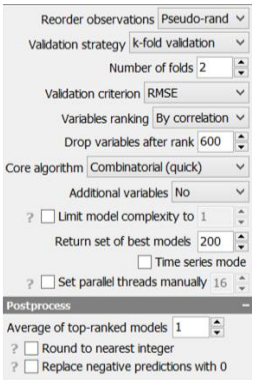


Figure 38 Screenshot of the experimenter's layout used for the classification analysis performed on 56 AMP active on *E.coli*

In this analysis, the ROC curve calculated for observations used to create the model is equal to 0.94 while the ROC calculated for the withheld observations is equal to 0.77.

The equation of the model obtained is the following:

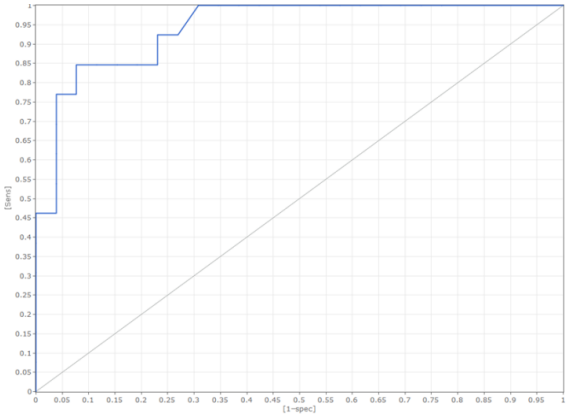
$$\boldsymbol{eq.23}\ \ \boldsymbol{y} = 2.19 + \big((\boldsymbol{number\ of\ hydrophobic\ amino\ acids}) * (-0.09)\big) + (\boldsymbol{a} * 0.055) + \big(\boldsymbol{b} * (-0.70)\big)$$

Where

a= transfer free energy to lipophilic phase [114]

b= amphiphilicity index [115]

		Predicted class		
		active	inactive	total
Actual class	active	24	2	26
	inactive	3	10	13
	total	27	12	39
	precision	0,889	0,833	
	accuracy	0,872	0,872	
Correctly classified instances		34	87%	
incorrectly classified instances		5	13%	
RMSE		0,357		



		Predicted class		
		active	inactive	total
Actual class	active	10	3	13
	inactive	3	1	4
	total	13	4	17
	precision	0,769	0,25	
	accuracy	0,647	0,647	
Correctly classified instances		11	65%	
incorrectly classified instances		6	35%	
RMSE		0,594		

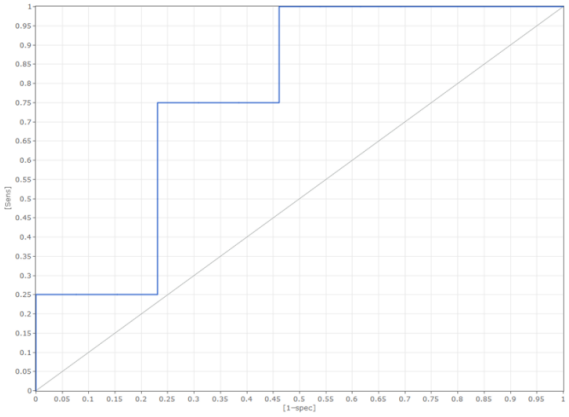


Figure 39 Results of the classification analysis on 56 AMP active on *E.coli*. At the top, there is the ROC curve calculated for observations used to train the model. It is equal to 0.94. In the chart below , there is the ROC calculated for the withheld observations. It is equal to 0.77. To the left of the graphs are shown the confusion matrix.

Regression Analysis: results

A regression analysis was performed on the same dataset of the 56 peptides active on *E.coli* (dataset E).

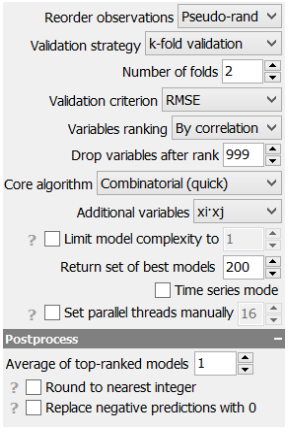


Figure 40 Screenshot of the experimenter's layout used for the regression analysis performed on 56 AMP active on *E.coli*

For this analysis, it was chosen an experimental protocol that involves in the creation of a training and a validation set with a random choice of rows (figure 40). The validation strategy is based on the division of the dataset into two groups and the validation criterion depends on the RMSE (eq.18). In this case, a combinatorial algorithm for model generation was chosen. New artificial features were chosen to improve the classification model. The function $\boldsymbol{x_i} * \boldsymbol{x_j}$ that adds all the possible multiplied pairs was chosen. The 20% of the AMP dataset was chosen to validate the final model.

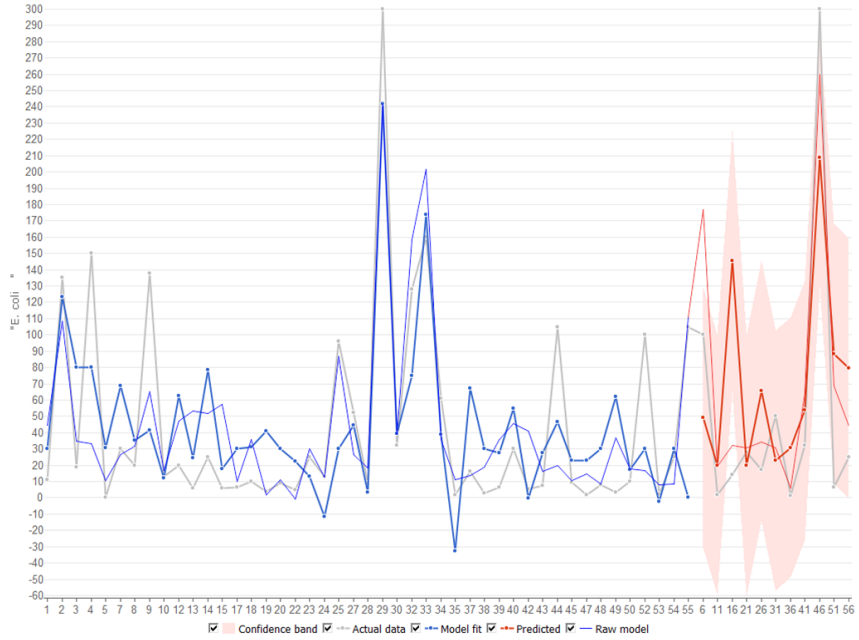


Figure 41 Performance of the model obtained. On the y axis there are the MIC values. On the x axis there are the AMP sequences. The gray line represents the MIC values for each of the 56 peptides in the dataset. The blue line represents how the model obtained (the equation) learned from the peptides of the training set. The red line represents how much the equation can be applied to the validation set (not used in the learning phase). The coefficient of determination (R^2) calculated for the observations used to create the model (training set) is equal to 0.66, while the R^2 calculated for the withheld observations (validation set) is equal to 0.81.

The equation of the model obtained is the following:

$$\boldsymbol{eq.24}\quad \boldsymbol{y} = 10.65 + (\boldsymbol{a} * \boldsymbol{b} * 1123.1) + \big(\boldsymbol{a} * \boldsymbol{c} * (-20.07)\big) + \big(\boldsymbol{a} * \boldsymbol{d} * (-440.49)\big) + (\boldsymbol{a} * \boldsymbol{e} * 2888.76) + \big(\boldsymbol{a} * \boldsymbol{f} * (-1054.48)\big) + \big(\boldsymbol{a} * \boldsymbol{g} * (-570.52)\big) + (\boldsymbol{h} * \boldsymbol{i} * 3.53)$$

Where

a= solvation energy in protein folding and binding [103]

b= the effect of burial of amino acid residues on protein stability [116]

c= volume changes on protein folding [117]

d= hydrophobicity index [104]

e- f= partition energies and distribution of residues in soluble proteins [105]

g= optimized relative partition energies [118]

h= molecular weight [106]

i= negative charge [119]

The coefficient of determination (R^2) calculated for observations used to create the model is equal to 0.66 while the R^2 calculated for the withheld observations is equal to 0.81 (figure 41).

4.3.4 *E.coli*: Dataset F

Classification Analysis: results

This model was generated from a classification analysis performed on a dataset of 36 AMP active on *E.coli* (Dataset F).

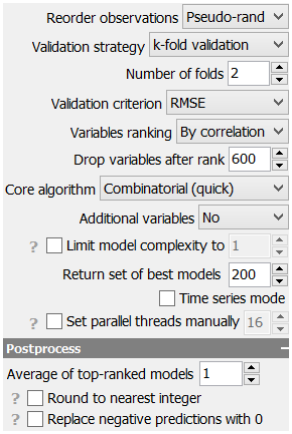


Figure 42 Screenshot of the experimenter's layout used for the classification analysis performed on 36 AMP active on *E.coli*

The experimental protocol used in this classification analysis is like that the experimenter's layout used in the analysis performed on the dataset of 56 antimicrobial peptides active on *E.coli* (section 4.3.3) (figure 42). The peptides with a MIC lower than 30 µM were called "active" and the peptides with a MIC greater than 30 µM were called "inactive". Also in this case, the 30% of the dataset of AMP was chosen to validate the classification model. In this analysis, the ROC curve calculated for observations used to create the model is equal to

0.99 while the ROC calculated for the withheld observations is equal to 0.83 (figure 43). The equation of the model obtained is the following:

$$\boldsymbol{eq.25}\ \boldsymbol{y} = 10.23 + (\boldsymbol{a} * (-19.44) + (\boldsymbol{b} * 4.66) + (\boldsymbol{c} * 1.74) + (\boldsymbol{d} * (-9.25) + (\boldsymbol{e} * 9.53)$$

Where

a= partial specific volume [120]

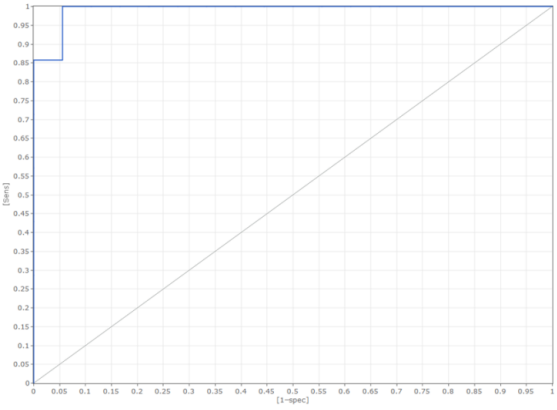
b= normalized frequency of beta-sheet [121]

c= weights for coil [97]

d= relative population of conformational state A [122]

e= interactivity scale obtained by maximizing the mean of correlation coefficient over single-domain globular proteins [123]

		Predicted class		
		active	inactive	total
Actual class	active	18	0	18
	inactive	1	6	7
	total	19	6	25
	precision	0,947	1	
	accuracy	0,96	0,96	
Correctly classified instances		24	96%	
incorrectly classified instances		1	4%	
RMSE		0,196		



		Predicted class		
		active	inactive	total
Actual class	active	8	1	9
	inactive	1	1	2
	total	9	2	11
	precision	0,889	0,5	
	accuracy	0,818	0,818	
Correctly classified instances		9	82%	
incorrectly classified instances		2	18%	
RMSE		0,426		

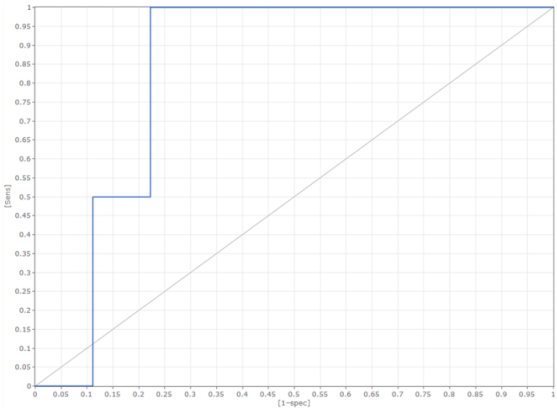


Figure 43 Results of the classification analysis on 36 AMP active on *E.coli*. At the top, there is the ROC curve calculated for observations used to train the model. It is equal to 0.99. In the chart below , there is the ROC calculated for the withheld observations. It is equal to 0.83. To the left of the graphs are shown the confusion matrix.

Regression Analysis: results

On the same dataset (dataset F), a regression analysis was performed using another experimental protocol (figure 44). Odd and even rows were chosen for the creation of the training and the validation sets. As described in the section 4.3.1, the "k-fold validation" option permits to split the dataset into k parts (2 in this case) and to train a model k times using k-1 parts, each time measuring model performance using a new remaining part. The GMDH neural network algorithm (section 4.2.1) was chosen to create the model. The core algorithm and the variables ranking (by correlation) are validated by the root-mean-square error (RMSE) criterion (eq.18). For this analysis, a linear neural function was used (eq.22):

$$\boldsymbol{eq.26} \quad \boldsymbol{a_0 + (a_1 * x_i) + (a_2 * x_j)}$$

This algorithm computes the weighted sum of the inputs. The upper limit for the number of network layers created by the algorithm is set to 33 (default) and the initial layer width that defines how many neurons are added to the set of inputs at each new layer, is set to 1 (default). Finally, to calculate the performance of the predictions, the 20% of the dataset was chosen for the validation.

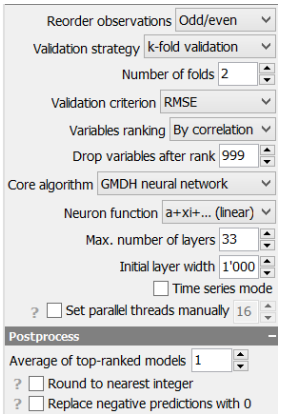


Figure 44 Screenshot of the experimenter's layout used for the regression analysis performed on 36 AMP active on *E.coli*

The coefficient of determination (R^2) calculated for observations used to create the model is equal to 0.82 while the R^2 calculated for the withheld observations is equal to 0.85 (figure 45).

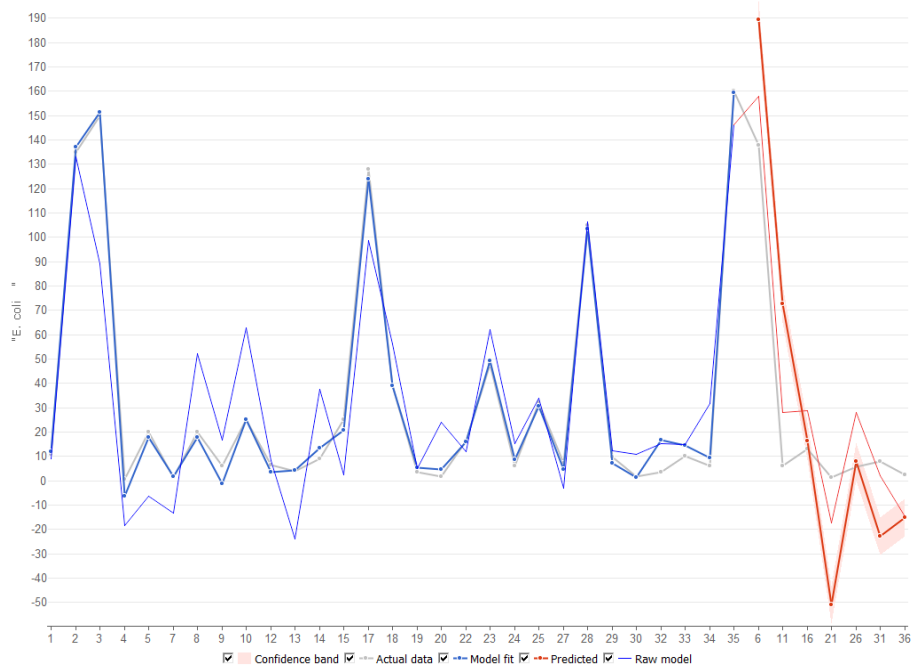


Figure 45 Performance of the model obtained. On the y axis there are the MIC values. On the x axis there are the AMP sequences. The gray line represents the MIC values for each of the 56 peptides in the dataset. The blue line represents how the model obtained (the equation) learned from the peptides of the training set. The red line represents how much the equation can be applied to the validation set (not used in the learning phase). The coefficient of determination (R^2) calculated for the observations used to create the model (training set) is equal to 0.82, while the R^2 calculated for the withheld observations (validation set) is equal to 0.85

The equation of the model obtained is the following:

$$\boldsymbol{eq.27}\quad \boldsymbol{y} = 9.34 - \boldsymbol{N1300} * 0.75 + \boldsymbol{N279} * 1.48$$

$$\boldsymbol{N279} = -8.33 + (\boldsymbol{N514} * 0.66) + (\boldsymbol{N900} * 0.59)$$

$$\boldsymbol{N900} = -444.94 + (\boldsymbol{a} * 405) + (\boldsymbol{N1095} * 1.59)$$

$$\boldsymbol{N1095} = -307.99 + (\boldsymbol{b} * 14.26) + (\boldsymbol{c} * 117.90)$$

$$\boldsymbol{N514} = -20.4 + (\boldsymbol{N1073} * 0.89) + (\boldsymbol{N1461} * 0.72)$$

$$\boldsymbol{N1461} = 479.76 + (\boldsymbol{d} * 394.49) - (\boldsymbol{e} * 395.36)$$

$$\boldsymbol{N1073} = -214.06 - (\boldsymbol{f} * 307.17) + (\boldsymbol{g} * 1975.63)$$

$$\boldsymbol{N1300} = 15.96 + (\boldsymbol{h} * 2.02) - (\boldsymbol{i} * 31.85)$$

Where

a= normalized frequency of middle helix [121]

b= NMR chemical shift of alpha-carbon [124]

c= the surface and inside volumes in globular proteins [125]

d= prediction factor of the secondary structure of globular proteins [97]

f= normalized hydrophobicity for α-proteins [126]

e= frequency of Helix-capping at helix termini C4 [98]

g= eigenvector of contact matrices and hydrophobicity profiles in proteins [123]

h= protein surface accessibility [127]

i= amphiphilicity index [115]

4.3.5 *S.aureus*: Dataset G

Classification Analysis: results

A dataset of 56 antimicrobial peptides active on *S.aureus* (dataset G) was created to perform the classification analysis.

The peptides with a MIC lower than 30 μM were called “active” and the peptides with a MIC greater than 30 μM were called "inactive". The molecular descriptors used in this analysis are 46, including length, charge, helicity, flexibility, polar and non-polar amino acids, etc.

In this analysis, the following parameters were set:

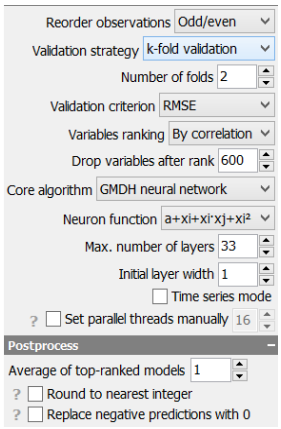


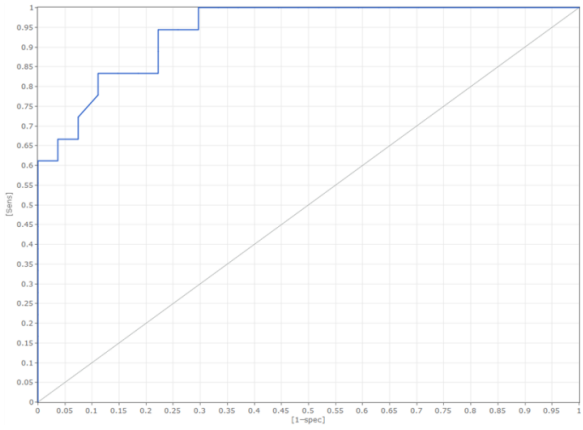
Figure 46 Screenshot of the experimenter's layout used for the classification analysis performed on 56 AMP active on *S.aureus*

"Reorder observations" is an option used to makes the training and the validation sets equally informative. In this case, odd and even raws were chosen in the creation of the training and the validation sets. For the option "validation strategy", that allows the choice of the model validation, the "k-fold validation" option that splits dataset into k parts (2 in this case) was chosen. Variables were chosen according to the correlation values. The core algorithm chosen in this analysis provides a GMDH neural network approach (section 4.2). In this case, a polynomial quadratic function was used:

$$eq.28 \quad a_0 + a_1 * x_i + a_2 * x_j + a_3 * x_i * x_j + a_4 * x_i^2 + a_5 * x_j^2$$

The upper limit for the number of network layers created by the algorithm is set to 33 (default) and the initial layer width that defines how many neurons are added to the set of inputs at each new layer, is set to 1 (default). Finally, to calculate the performance of the predictions, the 20% of the dataset was validated.

		Predicted class		
		active	inactive	total
Actual class	active	24	3	27
	inactive	4	14	18
	total	28	17	45
	precision	0,857	0,824	
	accuracy	0,844	0,844	
Correctly classified instances		38	84%	
incorrectly classified instances		7	16%	
RMSE		0,394		



		Predicted class		
		active	inactive	total
Actual class	active	4	1	5
	inactive	1	5	6
	total	5	6	11
	precision	0,8	0,833	
	accuracy	0,818	0,818	
Correctly classified instances		9	82%	
incorrectly classified instances		2	18%	
RMSE		0,426		

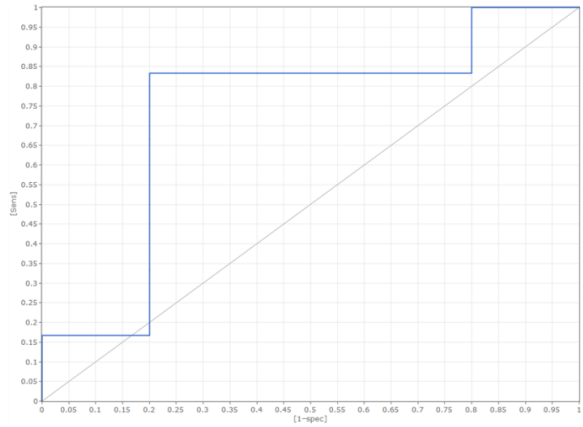


Figure 47 Results of the classification analysis on 56 AMP active on *S.aureus*. At the top, there is the ROC curve calculated for observations used to train the model. It is equal to 0.94. In the chart below, there is the ROC calculated for the withheld observations. It is equal to 0.73. To the left of the graphs are shown the confusion matrix.

In this analysis, the ROC curve calculated for observations used to create the model is equal to 0.94 while the ROC calculated for the withheld observations is equal to 0.73 (figure 47).

The equation of the model obtained is the following:

$$\boldsymbol{eq.29}\ \ \boldsymbol{y} = 0.12 - ((\boldsymbol{number\ of\ Leucine}) * 0.045) + ((\boldsymbol{number\ of\ Leucine}) * \boldsymbol{N2} * 0.05) + (\boldsymbol{N2} * 0.88)$$

Where

$$\boldsymbol{N2} = 0.058 - (\boldsymbol{N7} * 0.37) + (\boldsymbol{N3} * 1.22)$$

$$\boldsymbol{N3} = 0.015 - \big((\boldsymbol{number\ of\ Isoleucine}) * \boldsymbol{N4} * 0.11\big) + (\boldsymbol{N4} * 1.13)$$

$$\boldsymbol{N4} = -0.11 + \big((\boldsymbol{number\ of\ Serine}) * 0.33\big) - ((\boldsymbol{number\ of\ Serine})^2 * 0.077) + (\boldsymbol{N5} * 1.02)$$

$$\boldsymbol{N5} = 3.71 - (\boldsymbol{Isoelectric\ point} * 0.64) + (\boldsymbol{Isoelectric\ point} * \boldsymbol{N6} * 0.20) + ((\boldsymbol{Isoelectric\ point})^2 * 0.027)) - (\boldsymbol{N6} * 1.14)$$

$$\boldsymbol{N6} = -0.023 + ((\boldsymbol{number\ of\ Histidine}) * ((\boldsymbol{N7})2 * 1.59)$$

$$\boldsymbol{N7} = 1.04 - ((\boldsymbol{number\ of\ Tryptophan}) * (\boldsymbol{number\ of\ hydrophobic\ amino\ acids}) * 0.03) - ((\boldsymbol{number\ of\ hydrophobic\ amino\ acids}) * 0.05)$$

Regression Analysis: results

On the same set of AMP active on *S.aureus* (dataset G), a regression analysis was performed using the same experimental protocol of the classification analysis (figure 48). The molecular descriptors used in this analysis are 46, including length, charge, helicity, flexibility, polar and apolar amino acids, etc. The coefficient of determination (R^2) calculated for observations used to create the model is equal to 0.84 while the R^2 calculated for the withheld observations is equal to 0.72.

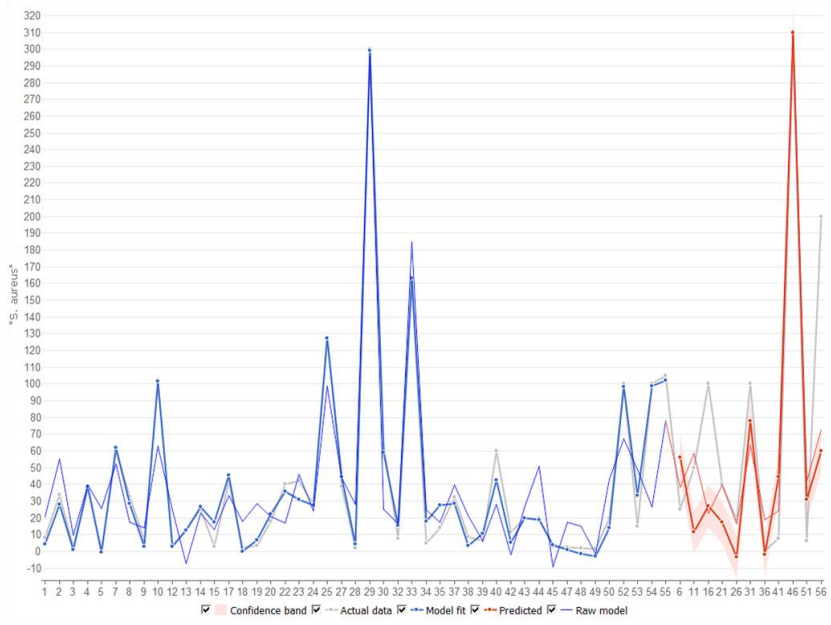


Figure 48 Performance of the model obtained. On the y axis there are the MIC values. On the x axis there are the AMP sequences. The gray line represents the MIC values for each of the 56 peptides in the dataset. The blue line represents how the model obtained (the equation) learned from the peptides of the training set. The red line represents how much the equation can be applied to the validation set (not used in the learning phase). The coefficient of determination (R^2) calculated for the observations used to create the model (training set) is equal to 0.84, while the R^2 calculated for the withheld observations (validation set) is equal to 0.72

The equation of the model obtained is the following:

$$\boldsymbol{eq.30}\quad \boldsymbol{y} = -31.19 + (\boldsymbol{N888} * 0.83) + (\boldsymbol{N518} * 0.98)$$

$$\boldsymbol{N518} = 13.08 + (\boldsymbol{N622} * \boldsymbol{N751} * 0.01)$$

$$\boldsymbol{N751} = 109.43 - ((\boldsymbol{FLEXIBILITY}) * 8.003) + ((\boldsymbol{FLEXIBILITY}) * (\textbf{number of asparagines}) * 14.56) - ((\textbf{number of asparagines}) * 143.47)$$

$$\boldsymbol{N622} = 333.12 - (\boldsymbol{pH5} * 54.73) + (\boldsymbol{pH5} * (\boldsymbol{Isoelectric\ point})) * 5.2 - ((\boldsymbol{Isoelectric\ point}) * 29.5)$$

$$\boldsymbol{N888} = 36.63 - ((\boldsymbol{\Delta CHARGE}) * 4.06) - (\boldsymbol{\Delta CHARGE}) * (\textbf{number of methionines}) * 21.64) + ((\textbf{number of methionines}) * 44.93)$$

Cytotoxic activity: an overview

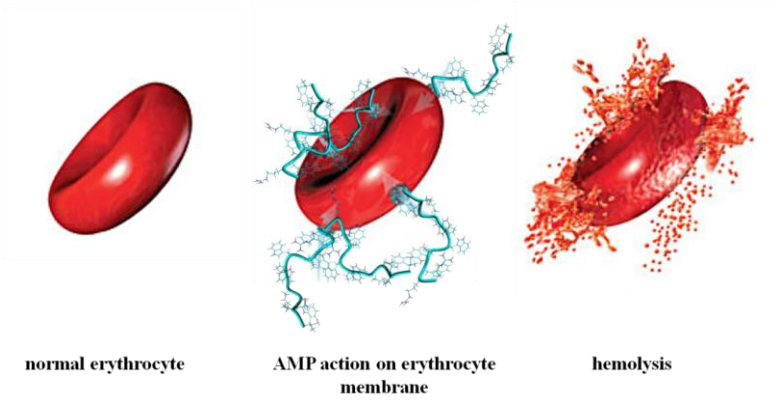


Figure 49 Representation of the action of antimicrobial peptides on blood erythrocytes

For the new predictive analysis with GMDH method, they were selected AMP sequences for which, in the literature, were also known the hemolytic activity values. Due the fact that antimicrobial peptides are considered alternative drugs for living organisms, to know their cytotoxic activity is very important. In fact, the aim is to design antimicrobial peptides active against pathogenic microorganisms (bacteria, viruses and fungi) and non-toxic for humans (or other living organisms). These sequences were searched in a database, DBAASP (https://dbaasp.org), in which there are information for structure/activity studies about antimicrobial and hemolytic (cytotoxic) activities of AMP. The DBAASP search page allows users to search peptides according to their structural characteristics, source, synthesis type and target species. A total of 84 sequences of antimicrobial peptides for which were known the HC50 (the concentration of antimicrobial that kills the 50% of red blood cells) [128] were found. If available in the database, the MIC values against organisms such as *S.aureus*, *E.coli*, *P. aeruginosa* and *C. albicans* were also extracted. For all the sequences, the values of the 573 1D and 2D molecular descriptors were calculated, as described in the section 2.1 (Chapter II).

4.3.6 Erythrocytes: Dataset H

Classification Analysis: results

This classification analysis was performed on 47 antimicrobial peptides active on erythrocytes (dataset H). In this analysis, all available molecular descriptors (573 molecular descriptors) were used. The experimental protocol is based on the use of the combinatorial algorithm (section 4.2.1) without the addition of new functions. Observations, for the creation of training and test set, were ordered by the odd and even rows. The validation strategy consists in a process that, at each cycle, divides the dataset into two groups until it finds the optimal model. The variables ranking is due to the correlation values and the validation criterion is based on the RMSE (eq.18).

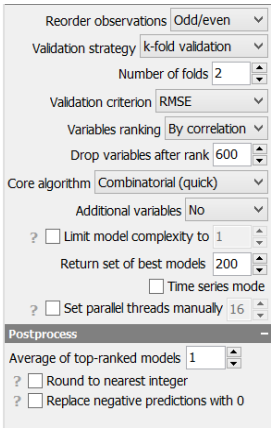


Figure 50 Screenshot of the experimenter's layout used for the classification analysis performed on 47 AMP active on erythrocytes

It was used the 20% of the dataset of AMP to validate the generated models. In this analysis, the ROC curve calculated for observations used to create the model is equal to 0.96 while the ROC calculated for the withheld observations is equal to 0.87.

The equation of the model obtained is the following:

$$\boldsymbol{eq.31} \quad \boldsymbol{y} = 49.20 + (\boldsymbol{a} * 1.88) + (\boldsymbol{b} * 3.19) + \big(\boldsymbol{c} * (-0.086)\big) + \big(\boldsymbol{d} * (-0.12)\big) + \big(\boldsymbol{e} * (-5.43)\big) + \big(\boldsymbol{f} * (-1.56)\big) + \big(\boldsymbol{g} * (-0.61)\big) + \big(\boldsymbol{h} * (-0.49)\big)$$

Where

a= length of the side chain [119]

b= average relative probability of inner beta-sheet [129]

c= side chain angle theta [130]

d= side chain torsion angle phi [130]

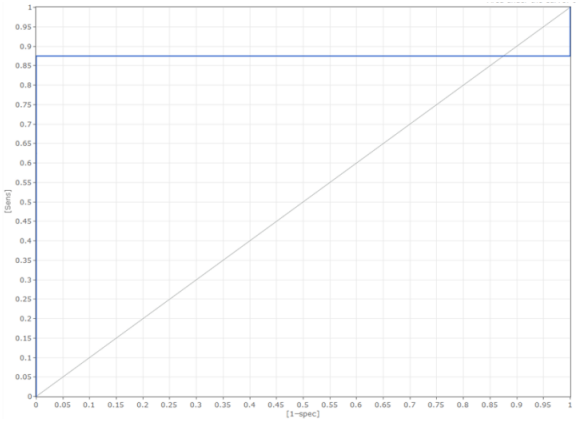
e= radius of gyration of side chain [130]

f= Van der Waals parameter [130]

g= SD of AA composition of total proteins [90]

h= hydrophobic packing and spatial arrangement of amino acid residues [131]

		Predicted class		
		active	inactive	total
	active	8	2	10
	inactive	1	27	28
Actual class	total	9	29	38
	precision	0,889	0,931	
	accuracy	0,921	0,921	
Correctly classified instances		35	92%	
incorrectly classified instances		3	8%	
RMSE		0,28		



		Predicted class		
		active	inactive	total
	active	0	1	1
	inactive	1	7	8
Actual class	total	1	8	9
	precision	0	0,875	
	accuracy	0,778	0,778	
Correctly classified instances		7	78%	
incorrectly classified instances		2	22%	
RMSE		0,471		

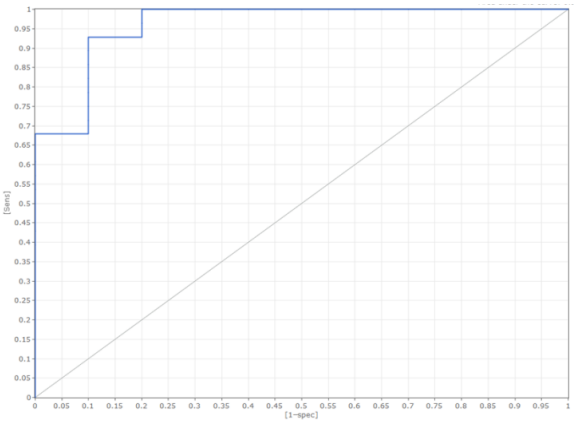


Figure 51 Results of the classification analysis on 47 AMP active on erythrocytes. At the top, there is the ROC curve calculated for observations used to create the model. It is equal to 0.96. In the chart below , there is the ROC calculated for the withheld observations. It is equal to 0.87. To the left of the graphs are shown the confusion matrix.

Regression Analysis: results

To conduct the regression analysis on the 47 AMP active on erythrocytes (dataset H), the same experimental protocol of the classification analysis was used (figure 50). The coefficient of determination (R^2) calculated for observations used to create the model is equal to 0.54 while the R^2 calculated for the withheld observations is equal to 0.59 (figure 52).

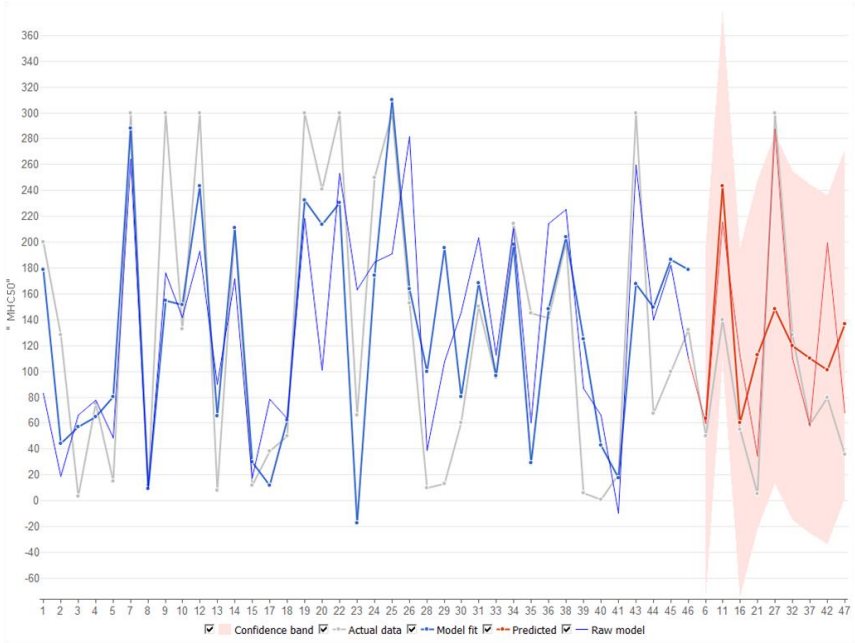


Figure 52 Performance of the model obtained. On the y axis there are the MIC values. On the x axis there are the AMP sequences.The gray line represents the MIC values for each of the 47 peptides in the dataset. The blue line represents how the model obtained (the equation) learned from the peptides of the training set. The red line represents how much the equation can be applied to the validation set (not used in the learning phase). The coefficient of determination (R^2) calculated for the observations used to create the model (training set) is equal to 0.54, while the R^2 calculated for the withheld observations (validation set) is equal to 0.59

The equation of the model obtained is the following:

$$\boldsymbol{eq.32)}\ \ \boldsymbol{y} = -7620.53 + (\boldsymbol{a} * 26.63) + \boldsymbol{b} * (-1.83) + \boldsymbol{c} * (-97.31) + (\boldsymbol{d} * 42.56) + \boldsymbol{e} * (-399.23) + (\boldsymbol{f} * 391.9) + (\boldsymbol{g} * 5453.39) + (\boldsymbol{h} * 483.82)$$

Where

a= steric properties of the side chains [119]

b-c-d-e-f = factors that describe the conformational properties of amino acid residues in globular proteins [132]

g= normalized flexibility parameter [133]

h= parameter that describes a protein domain linker [134]

4.4 Final comments

One of the limits for the correct execution of a prediction analysis is the choice of the molecular descriptors. These analyses generate models based on an input-output relationship. Input data are represented by molecular descriptors, while the output is the model that best responds, through the input data, to the problem that we have. In particular, the aim of this project was to establish a good relationship between the properties of AMP (input data) and their activity (question to answer) (figure 53).

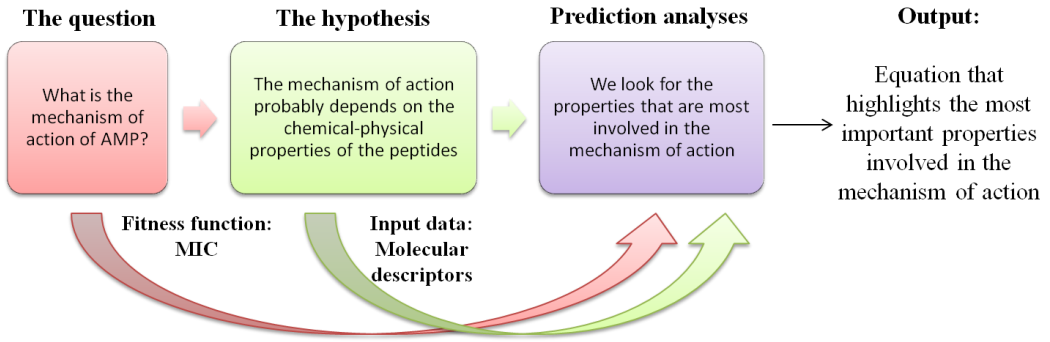


Figure 53 Prediction analyses look for a structure-activity relationship

The results of a predictive analysis must be easily interpretable and molecular descriptors must be appropriate for the final model. There are many and heterogeneous molecular descriptors of AMP and this aspect represents a limit to the efficiency of a prediction analysis. To enrich the Yadamp database (Chapter II), data about AMP were extracted from different papers in the literature, or calculated. The activity data (MIC) were also extracted from papers in the literature. The results of the predictive analyses (described above) were satisfactory and they were implemented in the Yadamp database. However, they still suffered from the elevated number of data, the excessive heterogeneity and also possible experimental errors. For example, the results of the analyses with genetic algorithms has a high R^2, but it is often accompanied by a high LOF (lack-

of-fit). The risk of overfitting is high when many data are available and the network is excessively trained.

Starting from the idea to create even more homogeneous antimicrobial peptide subsets and to generate accurate prediction models, PCA (principal component analysis) and cluster analyses were performed (Chapter V).

Chapter V

5.1 PCA and Cluster Analysis: how does they work?

"PCA is one of the most important results from applied linear algebra" [135]

The principal component analysis (PCA) is the main application for reducing the size of a dataset without losing information. PCA also allows to study the relationships between different descriptors and prepares data for further analyses, for example regression studies. The aim of the PCA is to extract important information from the dataset and to represent it as a set of new orthogonal variables called principal components [136]. The original variables are transformed into an orthogonal set of linear combinations, where, each principal component is a combination of the original variables, **v**, defined using a loading coefficient, **a** (eq.33-34).

$$eq.33 \quad PC_1 = a_{1,1}v_1 + a_{1,2}v_2 + \ \dots + a_{1,n}v_n$$

$$eq.34 \quad PC_2 = a_{2,1}v_1 + a_{2,2}v_2 + \ \dots + a_{2,n}v_n$$

Most of the variance of a dataset is usually contained in the first few components. The use of these principal components allows to proceed with subsequent analyses using a smaller dataset. If the variables are independent, PCA application is not productive. So, the basic idea is that the variables are inter-correlated (figure 54).

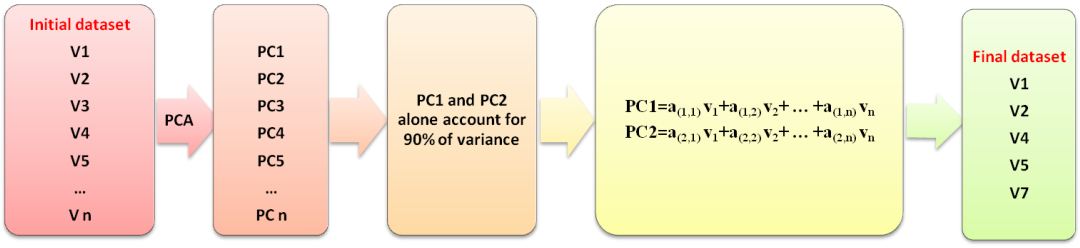


Figure 54 Schematic rapresentation that illustrates how PCA can potentially help in reducing data dimensions with a hypothetical dataset of **m** variables.

Cluster analysis, like PCA, is also a multivariate analysis technique. Through this approach, it is possible to group data by minimizing the internal *logical distance* of each group and maximizing the distance between them. *Logical distance* is expressed through similarity and dissimilarity values. The aim is to identify groups in which the elements in each group are more similar to each other than the elements in other groups. The matrix of the starting data is transformed into a matrix of distances between the pairs of observations. Cluster analysis is basically based on two points: the definition of a distance measure between observations and the choice of the method by which the groups are formed. Using a Euclidean distance measure, each distance matrix element, $\boldsymbol{d_{ij}}$, is given by:

$$\boldsymbol{eq.35)} \quad \boldsymbol{d_{ij}} = \sqrt{\sum_{k=1} \left(\boldsymbol{x_{ik}} - \boldsymbol{x_{jk}}\right)^2}$$

Where **i** and **j** are two points in an **n**-dimensional space. The distance matrix was calculated (eq.35) and then used to classify samples into clusters of similar members.

The second key point is the choice of a classification method (or algorithm). The most common classification methods are:

- Aggregate hierarchical methods
- Divisional hierarchical methods
- Non-hierarchical methods

Hierarchical methods consist of subsequent divisions of the data. In the case of aggregate hierarchical methods, the **n** initial data are fused into wider groups (at the end we have one group); in the case of divisive methods (or "scissors") are defined partitions of the initial set (at the end, **n** clusters contain each element). The main feature that distinguishes hierarchical from non-hierarchical methods is that, in the first case, when an object enters a cluster, it is no longer removed. The non-hierarchical method partitions the units into a predefined number of groups. The non-hierarchical method is essentially divided into two phases (figure 55):

1. The determination of an initial partition of the **n** individuals in **G** groups;

2. The subsequent displacement of the units between the **G** groups to obtain the partition that best suits the concepts of homogeneity within the groups;

The main limit of non-hierarchical methods is to have in advance an idea of the number of groups.

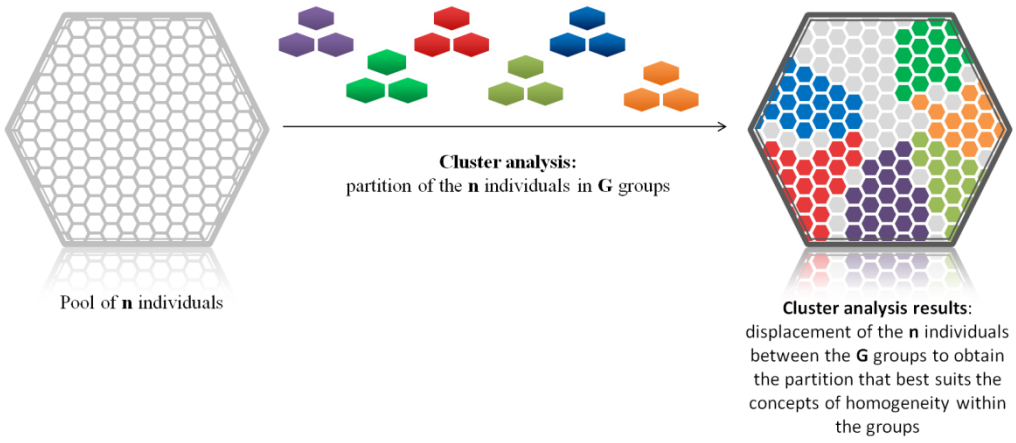


Figure 55 Schematic representation that illustrates how the cluster analysis works

In this work, non-hierarchical cluster analysis with k-means method was used. The goal is to minimize the total intra-cluster variance. Each cluster is identified by a centroid or midpoint (a representative cluster point). The algorithm follows an iterative procedure. Initially, it creates K partitions and assigns an entry point to each of them (randomly or using some heuristic information). At this point the algorithm calculates the centroid of each group. Then, the algorithm constructs a new partition by associating each entry point to the cluster in which centroid is closer to it. There is a continuous recalculation of the centroids until the algorithm converges (the algorithm found the maximum similarity within the groups).

5.1.2 PCA Analysis: results

In the graph (figure 56), the principal component vector 1 (x axis) is related to the principal component vector 2 (y axis) values, generated from a dataset of 1151 AMP with a length between 5 and 35 amino acids, active on *S.aureus*. These are the first two principal components vectors generated by PCA and contain the maximum amount of information in terms of variance. The PCA analysis was conducted using the software Materials Studio 7 and performed considering all the 573 molecular descriptors relative to AMP. To display the results of this PCA, the Software Tableau was used (www.tableau.com).

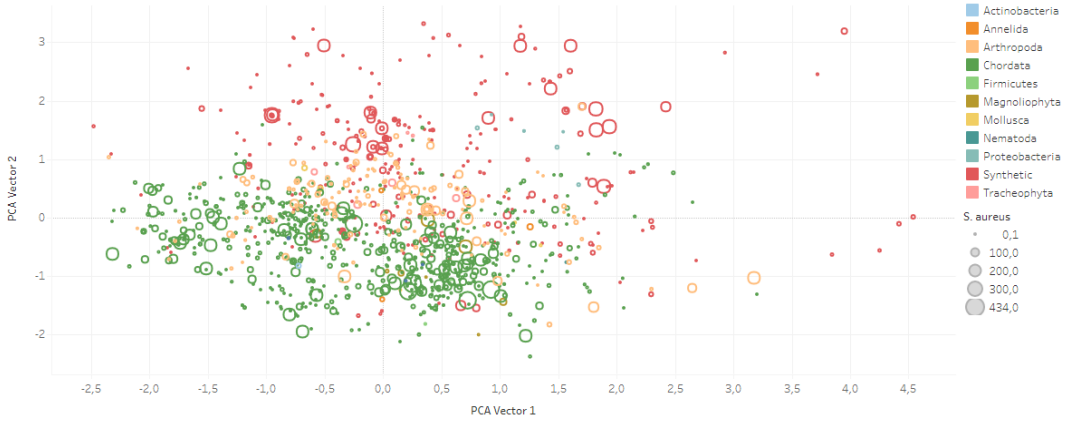


Figure 56 PCA on antimicrobial peptides active against *S.aureus*, with a length between 5 and 35 aa (1151 AMP). The colors refer to the phylum and the size of the circles to the MIC. The red color indicates synthetic peptides; the green color indicates the phylum of the peptides extracted from Chordata; the yellow color indicates the phylum of peptides extracted from Arthropoda; in the graph there is a clear division of the different phyla and a closeness between the variability in the phylum Chordata and the variability in the phylum Arthropoda. PCA is able to find correlations within a phylum from chemical physical characteristics

In the graph in figure 56, the colors refer to the phyla and the size of the circles refers to the MIC. The red color indicates the synthetic peptides, the green color indicates the peptides extracted from the phylum Chordata and the yellow color indicates the peptides extracted from the phylum Arthropoda. We can appreciate a clear division of the two phyla (Chordata and Arthropoda) and the group of

synthetic peptides. This division suggests that the peptides of these three phyla have different chemical-physical characteristics and different strategies of killing against the same microorganism (in this case *S.aureus*). However, the transmembrane proteomes, as demonstrated, are not very different. Therefore, it might be interesting to investigate the sequences of AMP in different phyla to identify common patterns and, eventually, a killing action against the same microorganisms. These considerations are important in order to design selective antimicrobial peptides. In addition, the study of these relationships could be a new development in the field of phylogenetics. In fact, the phylogenetic relationship among the kingdoms Animalia is a longstanding controversy and the proposed phylogenetic trees are very different [137].

5.1.3 Cluster Analysis: results

Starting from the results of PCA, non-hierarchical Cluster Analyses (see paragraph 5.1) were performed on the antimicrobial peptides active against different target organisms. The purpose was to find, within each phylum, sets of AMP homogeneous in their physical and chemical characteristics.

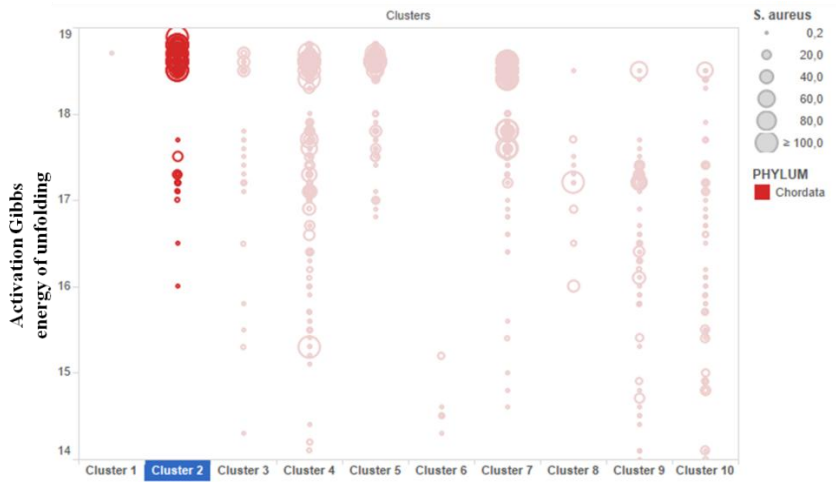


Figure 57 Cluster Analysis performed on 3054 AMP Red color indicates the phylum Chordata and the size of the circles indicates the MIC of the AMP on *S.aureus*. The cluster 2 shows a trend where increasing values of Activation Gibbs energy of unfolding correspond to a less activity of the peptides.

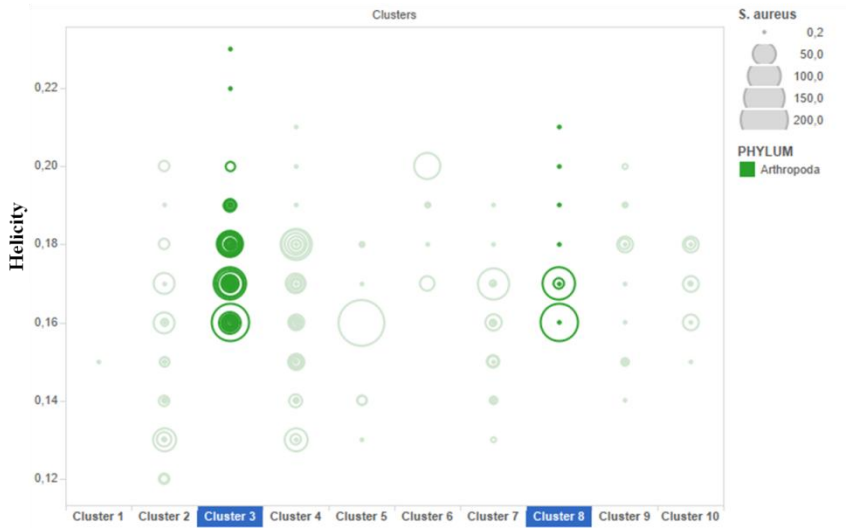


Figure 58 Cluster Analysis performed on 3054 AMP Green color indicates the phylum Arthropoda and the size of the circles indicates the MIC of the AMP against *S.aureus*. The clusters 3 and 8 show a trend where increasing values of the helicity parameter correspond to more active peptides.

Here, some of the results obtained are shown. This analysis was performed on 3054 antimicrobial peptides and the graphs were obtained by the Tableau software (figure 57-58). The figure 57 shows the relationship between the activation Gibbs energy of unfolding [122] and the antimicrobial activity of peptides against *S.aureus* in each cluster. In the figure 58, it is highlighted the relationship between the helicity parameter and the antimicrobial activity of the peptides in each cluster. In both graphs the AMP are divided in 10 clusters, the size of the shapes indicates the MIC against *S.aureus* and the color indicates the phylum from which the peptides were extracted. As shown in the figure 57, for the phylum Chordata in the Cluster 2, when the values of the activation Gibbs energy of unfolding increase, the activity of antimicrobial peptides decreases. This correlation has a logic: when the energy of unfolding and, therefore, the conformational instability of the peptides increases, their antimicrobial activity decreases. In the figure 58, for the phylum Arthropoda in the Cluster 3 and 8, the helicity and the activity of antimicrobial peptides are directly proportional. This means that a high probability of AMP to assume a more stable secondary structure corresponds to a higher antimicrobial activity.

These correlations and others allowed to create homogeneous datasets of AMP and to perform more accurate QSAR analyses.

Chapter VI

6.1 How to make prediction analysis more informative? Search for new molecular descriptors

The descriptors 1D and 2D currently available fail to capture all of the peptides properties. One of the goals of this project is to find new molecular descriptors of AMP. The peptides are extremely flexible molecules. When AMP interact and insert into the target membrane, they undergo conformational changes: in water, their structure is hydrophilic; when they interact with membranes, they expose a hydrophobic region [138].

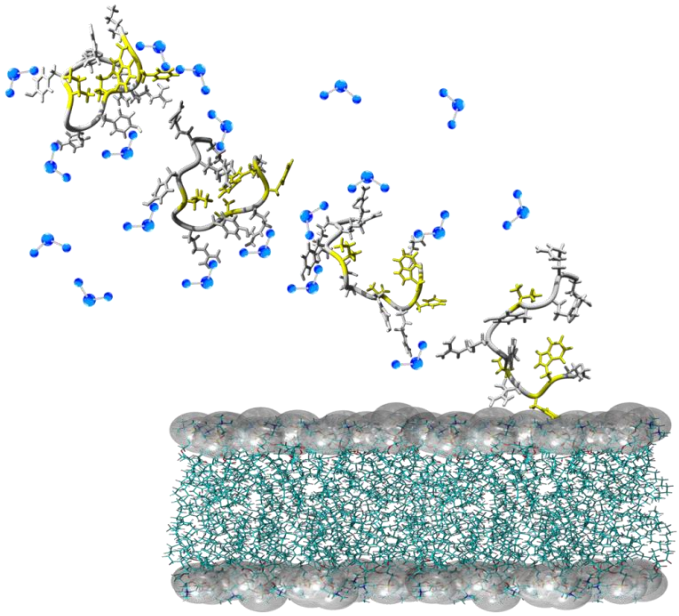


Figure 59 An image that illustrates a peptide that interacts with a target membrane and exposes its hydrophobic residues (in yellow). Water molecules are in blue.

Therefore, the binding energy among the peptides when they associated on the membrane surface and also the binding energy between the peptides and the membrane or protein receptors, are molecular descriptors that would be interesting to calculate.

6.1.1 Creation of a new tool for Molecular Docking: YADA

Figure 60 YADA software interface

Most of the processes of signal transduction in biological systems is based on the interaction between molecules [139]. The receptors can bind some ligands (substrates, inhibitors, activators or neurotransmitters) and this interaction is expressed as binding energy. Specific residues of the target receptor are involved in this recognition and form the active site. Coenzymes and metal ions can act as cofactors of the reaction between receptors and ligands and specific inhibitors can stop it.

There are currently more than 35.000 crystallographic or NMR structures of proteins available in the Protein Data Bank (https://www.rcsb.org). With the discovery of new techniques, such as X-ray crystallography, the number of macromolecules continues to grow over time. Most of these molecules has important roles in life processes and they are considered potential therapeutic targets [140]. We need to understand what are the mode of binding of a ligand against a target protein to establish relationships between structure and activity in the development of new drugs.

Docking is a viable alternative to experimental techniques [141].

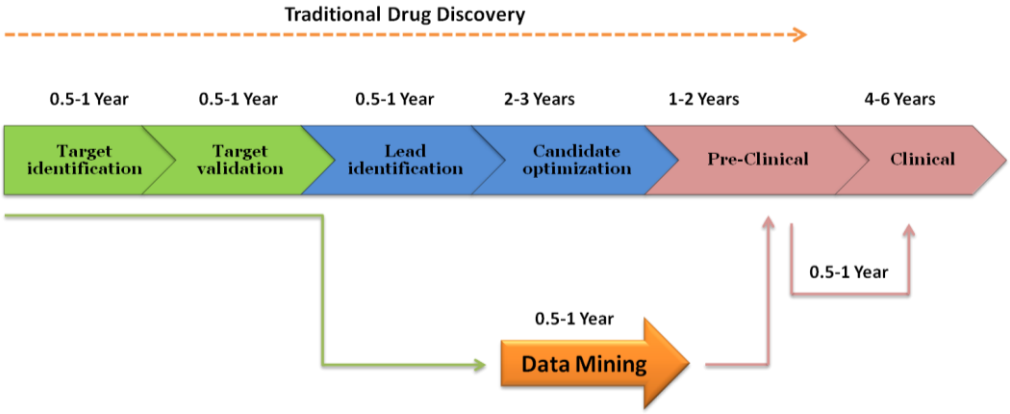


Figure 61 Traditional Drug Discovery

The molecular docking studies the conformation of a molecule complexed with other species and calculates how strongly a ligand binds the target. The binding affinity of protein−ligand complexes is an open problem in computational bioscience [142]. The molecular docking uses specific algorithms to predict the geometry of the complex receptor-ligand and parameterized functions to estimate their affinity. An algorithm allows to generate a series of poses which are analyzed by a scoring function to identify the true binding mode(s) and to estimate their binding affinity (figure 62). There are several methods of docking: rigid-body, flexible-ligand docking, and flexible ligand-flexible target [143].

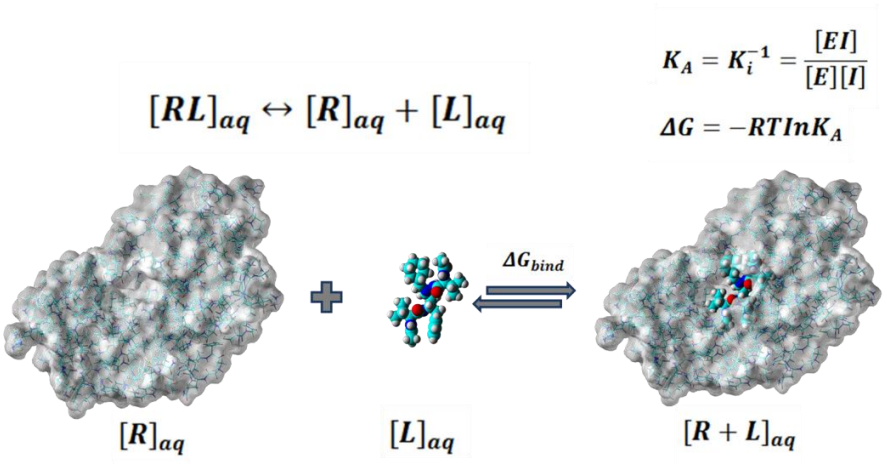


Figure 62 Thermodynamic of protein-ligand interactions

Vina (Vina Is Not Autodock) is the most used molecular docking program. Original Vina is a program of docking that predicts a series of poses between receptor and ligand and calculates the binding affinity [144]. When the protein and its ligands are known, the molecular docking returns a series of poses of this complex, using a specific algorithm. The limitation of an algorithm are the speed and the ability to find all the poses in the space. The thermodynamic of the receptor-ligand complex is represented by the scoring function that allows us to distinguish the better poses than worse [140]. Often, the active site of a receptor is unknown and the alternative is a blind docking. The blind docking scans the entire surface of the receptor to find all the possible active sites (figure 63). This procedure presents several limitations in terms of effectiveness and time.

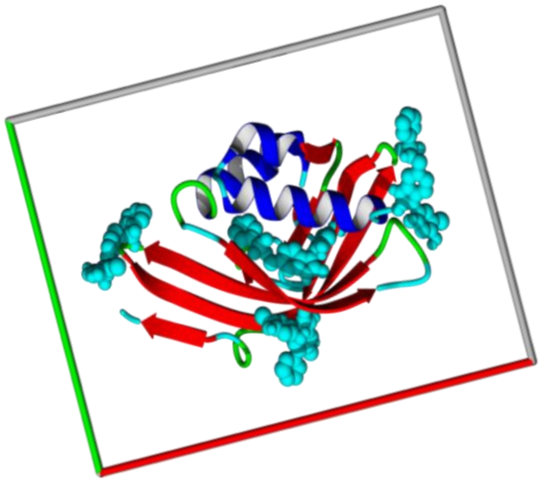


Figure 63 Blind docking explores all the surface of the target protein to find the best pose of a ligand

No commercially available or free-to-use software for molecular docking consider the importance of conserved sequence in proteins [145]. The sequence and the function of a molecule are usually closely related but, despite numerous studies, it is not yet clear how the conserved amino acid residues are involved in the evolution and the role that they have in protein function [146]. During extensive docking analysis, it was observed that conserved residues often lie on binding sites [145].

This work can be divided into three parts:

1) Preliminary consideration: observation of protein-ligand complexes;

2) Modifying Vina code and creation of a new docking system, called Yada (Yet Another Docking Approach); Yada is available for Windows and Linux and it is free to download at www.yada.unisa.it [146];

3) Optimization and validation of Yada;

It was done a careful literature search to demonstrate the hypothesis about the importance of conserved residues in the binding site of a protein. The preliminary analysis was very satisfactory.

In order to perform an extensive statistical analysis of the location of binding regions on protein surfaces, pdb structures from the PDBBind database (http://www.pdbbind.org/) were checked. PDBBind is a collection of binding affinities for the protein-ligand complexes in the Protein Data Bank (PDB) [147]. Using Yasara, a molecular-graphics, -modeling and -simulation program, data about the structure, the conservation and the binding site of the receptors (http://www.yasara.org/) were obtained. For example, the figure 64 shows the surface of the Gamma-glutamyltranspeptidase from *Bacillus subtilis* (3WHR) in the bounded and unbounded form. The conserved residues are yellow.

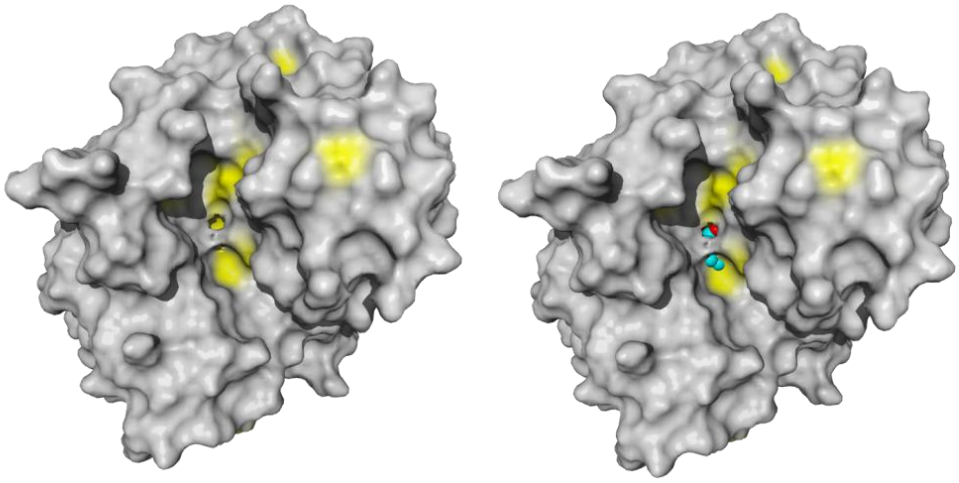


Figure 64 Gamma-glutamyltranspeptidase surface from *Bacillus subtilis* (**3WHR**) in the bounded and unbounded form. The conserved residues are yellow.

In the figure 65 is also represented the surface of a protein, the Glutatione transferase A1-1 (1GSD)**,** with the conserved residues colored in yellow, in bounded and unbounded form.

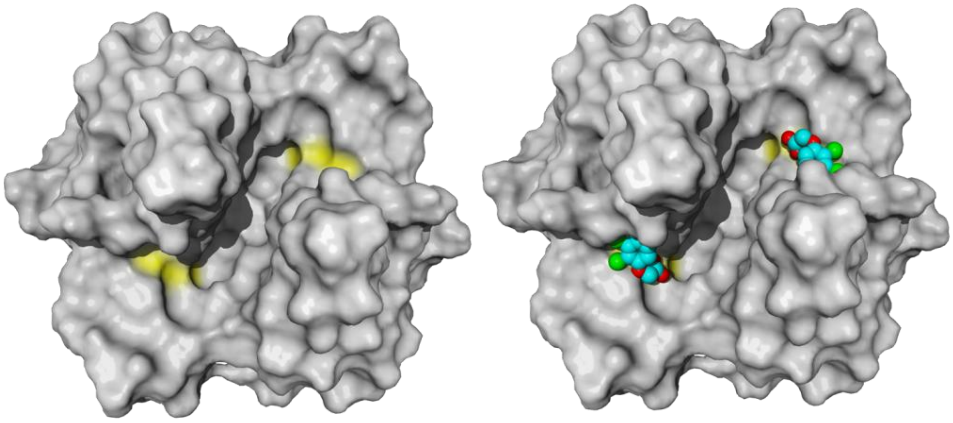


Figure 65 Glutatione transferase A1-1 (**1GSD**)**,** with the conserved residues colored in yellow, in bounded and unbounded form.

In the figure 66 is represented the Arabidopsis Hexokinase 1 (AtHXK1) structure in ligand-free form (4QS8) and in glucose-bound form (4QS7). Conserved residues are in yellow.

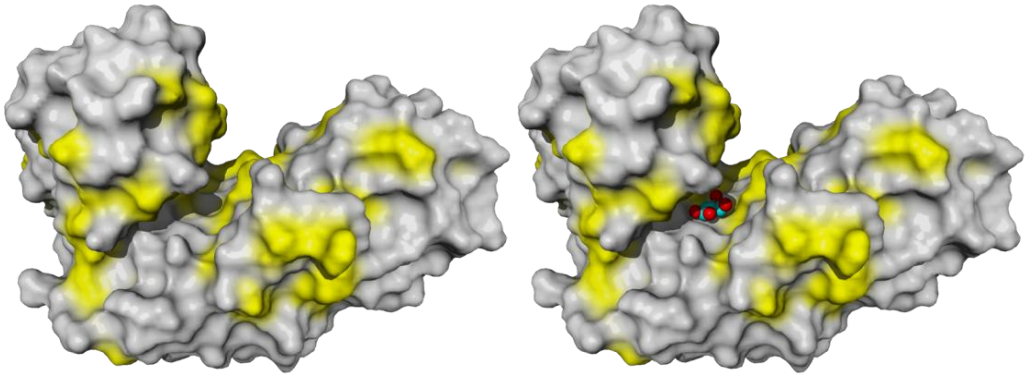


Figure 66 Arabidopsis Hexokinase 1 (AtHXK1) surface in ligand-free form (4QS8) and in glucose-bound form (4QS7). Conserved residues are in yellow.

In these cases and in others not shown here, it is clear that ligands bind conserved residues of their target proteins.

The original dataset on which the preliminary considerations were made, was randomly checked and reduced to eliminate dimers or structures with cofactor to reach a total of 305 pdb entries (Appendix C). Each amino acid of a protein was indicated with an integer corresponding to the conservation, as listed in the database Pdbfinder2 (http://swift.cmbi.ru.nl/gv/pdbfinder/) [148].

The conservation values can be obtained running a Blast search on the receptor sequence and then counting the number of punctual mutations. A value of 9 means that a particular amino acid is found conserved more than 95% of the times. For each residue, it was calculated the distance between residues and the ligand barycenter. As preliminary observation, they were counted the residues with conservation 9, 8, and 7 with the minimum distance from the ligand. Highly conserved residues (dist_9) tend to be closer to the ligand than less conserved residues (dist_7). At the same time, at longer distances less conserved residues tend to prevail (figure 67). The graph in the figure shows that most ligands have a distance less than 5.5 Å from residues with a high degree of conservation. This confirmed the hypothesis that the conserved residues are involved in the binding

site of the receptors. Then, the presence of conserved residues is an important condition to predict a binding site for a ligand.

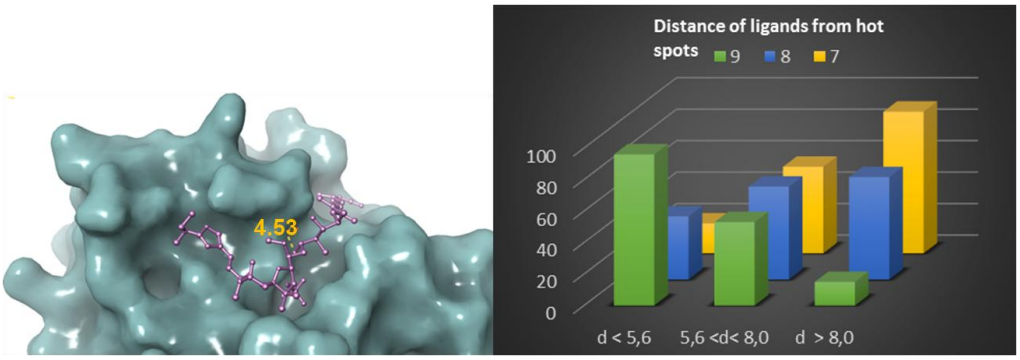


Figure 67 On the left the distance between the experimental position of a ligand and conserved residues in the binding site of the protein. On the right the distances of the experimental position of the ligands and the amino acidic residues, with a degree of conservation of 9 (high conservation), 8 (medium conservation) and 7 (low conservation).

6.1.2 Improving of the binding energy calculation

Starting from the initial considerations on the importance of conserved residues in a molecule, the idea was to drive ligands toward conserved regions on the surface, adding an extra term to the force field. To implement this new docking algorithm it was decided to start from an already existing docking program: Vina. It is an efficient and open source program for molecular docking [145] . It was followed an approach similar to the one used by Fan et al. [149]: pose prediction and ligand ranking were considered separately. We defined as hotspot (HS) the barycenter of spatially related conserved residues. The conserved regions can be easily obtained by multiple sequence analysis, but an easier way consisted in downloading essential information from the server PDBFinder 2 [148]. The distance of a pose from the HS was used to modify the Vina function. The calculation of the binding energy was modified adding a term that depended on the reciprocal of the distance between a ligand and the nearest HS. The new energy took into account also the conservation value of the residues [145].

To build a model of binding energy, besides the chemical-physical parameters already present in Vina, the distances between the ligands and the 20 amino acids and the shortest distances between ligands and the conserved residues, were considered. The descriptors were correlated with their experimental binding energy. It was used the method of genetic algorithms (look at the 3.1.1 paragraph), implemented in the Material Studio 7.0 package. The smoothness parameter was kept at the default value of 1.0 and the length of an individual was of 3 descriptors. 500 individuals were let evolve over 5000 new generations. The best equation was taken based on the highest squared correlation coefficient (R^2).

The new formula (eq.36) generated to calculate the binding energy in Yada is:

$$\boldsymbol{eq.36}\quad \boldsymbol{yaEnergy} = (\boldsymbol{VinaEnergy} * 0.536) - (\boldsymbol{ConsRes} * 0.273) + 6.097$$

where

yaEnergy: binding energy calculated by Yada

VinaEnergy: binding energy calculated by Vina

ConsRes: an integer with value between 0 and 9 that indicates to the conservation degree of the closest HS to the pose

6.1.3 Preliminary results and future perspectives

One of the typical problems of docking software (and Vina makes no exception) is that the pose ranking is made in terms of energy. Vina uses a semi-empiric calculation of the pose energy. Unfortunately, the calculation of free energy is far from being optimal and, consequently, the ranking process is poor. In YADA the ranking process is separated by the energy evaluation and it is possible to evaluate the goodness of the free energy of binding using the distance of the ligand from the conserved residues of the protein (hot spots). Another element to consider is the solvation aspect. This is usually treated implicitly, that is, by the use of implicit solvents or by modification of other scoring functions. Here, it was considered explicitly structural water in binding site.

The validation of a docking software is always a critical task. Several works already discussed this point [139]. The new ranking function (YaRank) was derived by the application of GA on a dataset of 180 proteins. The new energy function (YaEnergy) was calibrated on a set of more than 200 experimental free energies of binding (Appendix C). The accuracy of the new approach was tested on a set of 126 proteins and the results were compared to Vina, one of the most popular molecular docking tool (Appendix C). The docking procedure was total blind docking, 250 runs, Amber03 ff, without water molecules. We considered three aspects in blind docking: the goodness of the first pose in terms of RMSD between the docked pose, the experimental data, the free energy of binding and the execution time.

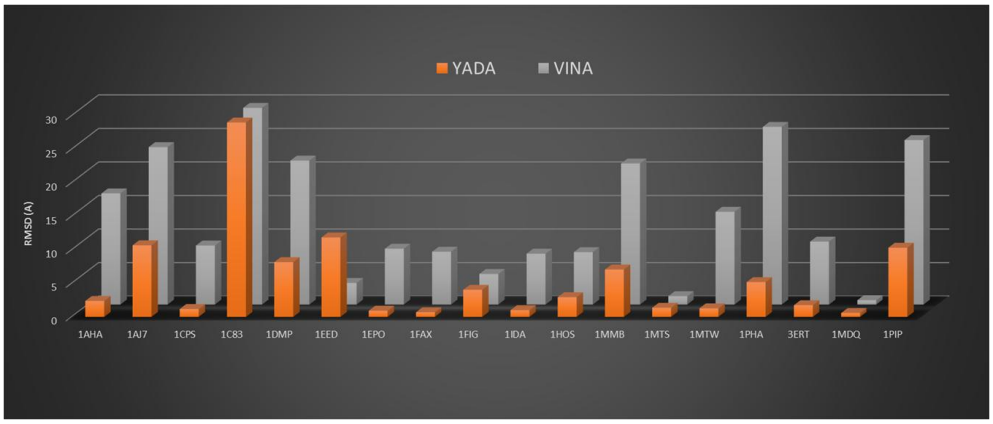


Figure 68 The RMSD of the best poses generated by YADA and VINA

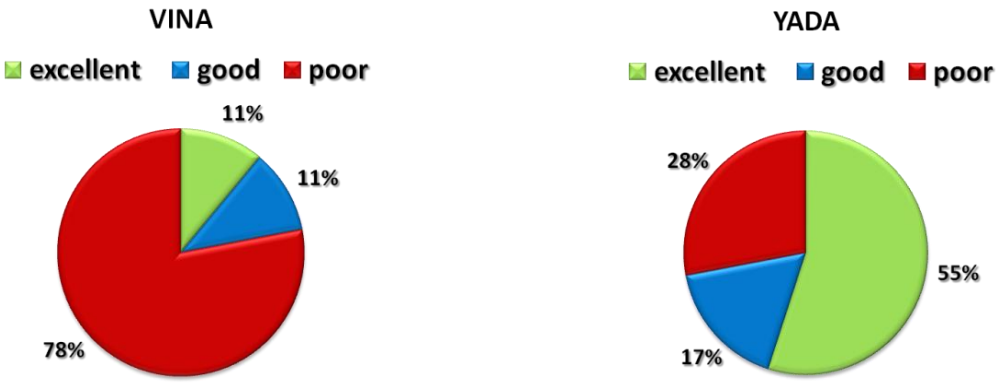


Figure 69 the RMSD of the best poses generated by YADA and VINA

In the diagram (figure 68) it is possible to see the RMSD of the best poses generated by YADA and VINA. The RMSD obtained with YADA is smaller and so better than VINA in most cases. Also, if we consider as excellent a performance with a ligand of RMSD < 2 Å, as good a performance of RMSD between 2 Å and 7 Å and as poor a performance of RMSD > 7 Å, we note that YADA compared to VINA have a poor performance only in 17% of cases (figure 69).

As example of a different use from conservation regions, Yada was recently used to monitor the residue mobility of a series of protein. The first normal modes of

vibration have been calculated through the Maestro program [150] and the mobility of each residue was calculated as RMSD respect to the crystallographic structure (Appendix C). In the figure 70, the vibrational nodes, the invariant points on the protein surface, are shown in yellow. Yada can directly exploit these pivotal HS to assist the docking procedure.

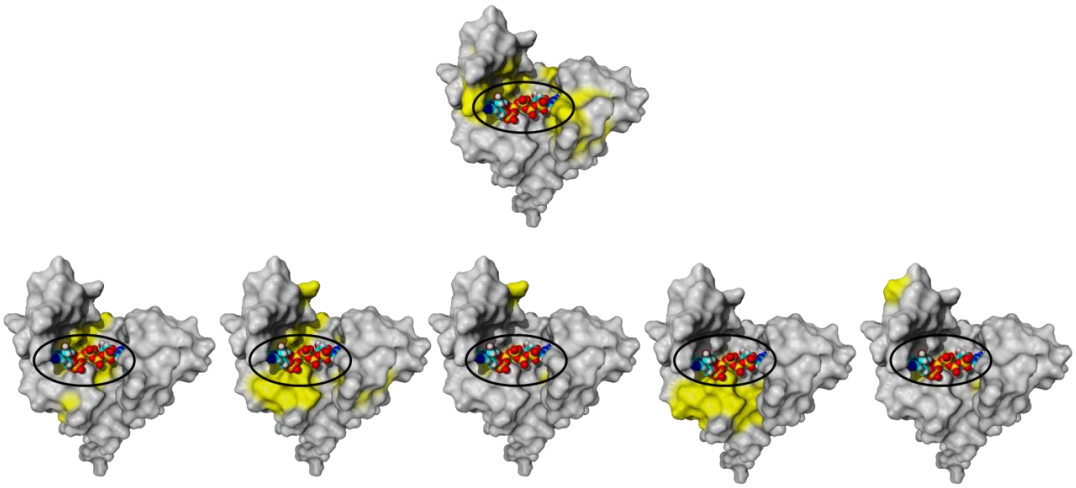


Figure 70 Low vibrational modes of the protein *E.coli* Guanylate Kinase (**2ANC**). The vibrational nodes, the invariant points on the protein surface, are shown in yellow.

The approach is very flexible and it will permit the extension to AMP interacting with other peptides, protein receptors and target membranes.

Chapter VII

7.1 Experimental studies: interaction between three new selective peptides and lipid vesicles

After careful investigations using the "Yadamp predict" tool (see paragraph 2.1.1), three amino acid sequences potentially active on Gram + bacteria were identified. Through a combinatorial calculation, 10,000 amino acid sequences were generated. The Yadamp predict tool was used to predict their activity. Between these, three sequences were chosen: they had different amino acid sequences, obtained the consent of different algorithms and contained tryptophan. The presence of tryptophan in the sequence was an important requirement because the idea was to exploit its spectrophotometric properties to monitor the interaction of peptides with lipid vesicles. It is an aromatic amino acid with a maximum absorption and emission of 280 nm and ~360 nm, respectively. The properties of these three small and cationic peptides, called p458, p459 and p460, are listed in the figure 71.

Length	Molecular weight	Hydrophobicity	Charge	
10	1472.87 (g/mol)	45.31	+3	**p458**
11	1398.79 (g/mol)	37.43	+4	**p459**
11	1410.88 (g/mol)	38.53	+5	**p460**

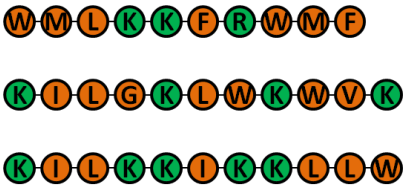


Figure 71 Properties of the three peptides designed (p458, p459 and p460)

The peptide/membranes interaction experiments were performed at the University of the Balearic Islands, in the Laboratory of molecular biology of the professor Pablo Vicente Escribá Ruiz. It was studied the binding between the peptides p458, p459 and p460 and unilamellar vesicles of 400 nm. To determine the lipid

percentage to use for the preparation of the different types of liposomes, experimental data reported in the literature were considered (see the paragraph 7.1.3) [151]. It was developed a protocol to evaluate the peptide-membrane binding by fluorescence and absorbance analyses, exploiting the aromatic properties of the amino acid tryptophan (Trp).

Fluorescence and absorbance analyses showed that these three peptides preferentially interact with negatively charged lipid vesicles (DMPG[3]-DMPG:CL[4] 3:1-PC[5]:CL 1:1) rather than with zwitterionic PC vesicles, principal component of eukaryotic membranes. The results suggest that the three peptides probably interact with the target membranes through different mechanisms of action. Furthermore, it seems that this interaction depends not only on the chemical-physical characteristics of the peptides, but also on the structural changes that the membranes undergo.

7.1.1 Synthesis of 3 new selective peptides

The peptides p458 (**WMLKKFRWMF**), p459 (**KILGKLWKWVK**) and p460 (**KILKKIKKLLW**) were synthesized by Ontores, in Zhejiang, China (http://www.ontoresinc.com/). The lyophilized peptides were dissolved in Milli-Q water, gently shaking until complete dissolution. Some aliquots of the solutions were prepared in order to work with small quantities and to avoid a possible degradation of the starting solutions.

7.1.2 Buffer preparation

A solution (100 mL) of HEPES and EDTA in pure water was prepared (pH=7.4), using the concentrations in the figure 72. A volume of 40 mL of this starting solution was extracted. To this solution, the required quantity of KCl (potassium chloride) was added. From the same starting solution of HEPES and EDTA,

[3] DMPG: 1, 2-dimyristoyl-sn-glycero-3-phospho-(1'-rac-glycerol)
[4] CL: Bovine heart cardiolipin
[5] PC: L-α-phosphatidylcholine

another volume of 40 mL was taken. To this solution, the required quantity of sucrose was added. In both cases, milli-Q water was added to a final volume of 50 mL[6] (figure 72).

Substances	Final concentration
HEPES (4-(2-hydroxyethyl) -1-piperazineethanesulfonic acid)	25 mM
EDTA (Ethylenediaminetetraacetic acid)	0.125 mM
KCl (Potassium chloride)	100 mM
Final volume = 50 mL	

Substances	Final concentration
HEPES (4-(2-hydroxyethyl) -1-piperazineethanesulfonic acid)	25 mM
EDTA (Ethylenediaminetetraacetic acid)	0.125 mM
Sucrose ($C_{12}H_{22}O_{11}$)	200 mM
Final volume = 50 mL	

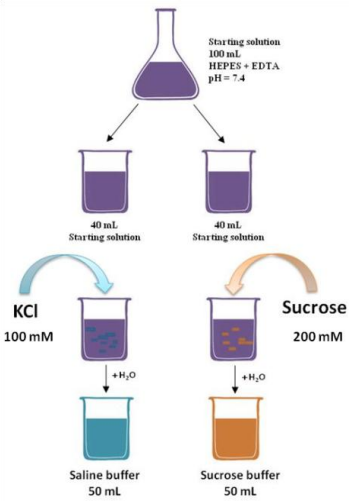


Figure 72 Saline and sucrose buffers preparation

7.1.3 Preparation of lipid vesicles

The hen egg L-α-phosphatidylcholine (PC) (figure 73) and the 1, 2-dimyristoyl-sn-glycero-3-phospho-(1'-rac-glycerol) (DMPG) (figure 74) were purchased from Avanti Lipids (https://avantilipids.com/). The L-α-phosphatidylcholine is a neutral phospholipid, particularly abundant on the outer sheet of the eukaryotic plasma membrane [151]. DMPG is a negatively charged phospholipid, especially present in the membrane of Gram + bacteria [151]. Bovine heart cardiolipin (CL) (figure 75) was purchased from Sigma Aldrich (www.sigmaaldrich.com). CL has two negative charges and it is mainly present in the Gram + bacteria membrane [151].

[6] For each experiment, it is advisable to prepare the two solutions (saline buffer and sucrose buffer) starting from the same mother solution, to have the same conditions.

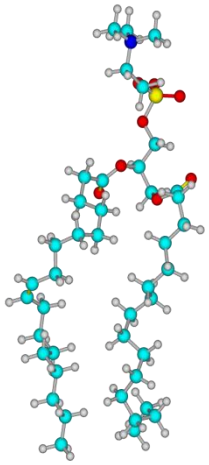


Figure 73 L-α-phosphatidylcholine (PC)

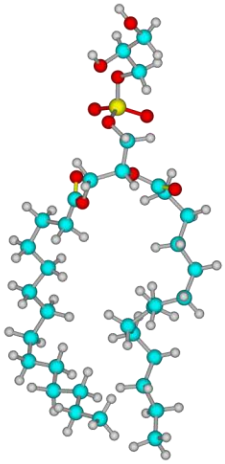


Figure 74 1, 2-dimyristoyl-sn-glycero-3-phospho-(1'-rac-glycerol) (DMPG)

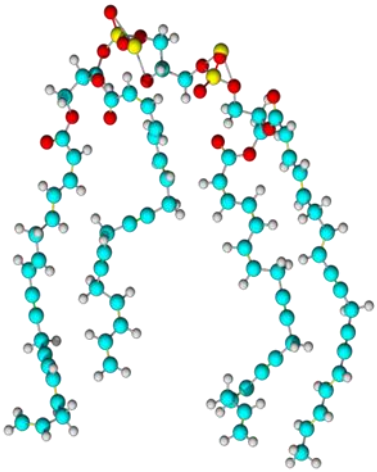


Figure 75 heart cardiolipin (CL)

The following lipid vesicles were prepared: PC, DMPG, DMPG:CL 3:1 and PC:CL 1:1. The same experimental protocol was used for the preparation of all these membranes. First, the concentration of the starting solution of lipids was proved using the Fiske method, based on the determination of the total amount of phosphorus [152]. At this point, the required quantity of the lipids was dissolved with a solution of chloroform:methanol 1:1, in a test tube. These organic solvents ensured a homogeneous lipid mixture. The chloroform was removed from the product by blowing a slow stream of argon over the chloroform solution. To remove the last traces of chloroform, the test tube was put on a vacuum system overnight (Figure 76).

Figure 76 On the left the argon flow system; on the right the vacuum pump

The next day, the dry lipid film formed at the bottom of the vial was suspended in sucrose buffer[7] (figure 77).

[7] The buffer volume was arbitrary. The important thing was to prove the concentration of lipids.

Figure 77 The dry lipid film formed at the bottom of the vial was suspended in sucrose buffer

The vial was shaken through an eppendorf vortex and then placed in a bath at ~40°C for 5 minutes (this procedure was repeated for three times). The concentration of lipids in the vial was tested by determining the total amount of phosphorus, using the Fiske method [152]. Next, 10 cycles were carried out. Each of them provided freezing in liquid hydrogen for 1 minute and defrosting in a bath at ~40°C for 5 minutes. At this initial stage, the vesicles formed in the vial were MLV (multilamellar lipid vesicles) with the same size. To form 400 nm (0.4 μm) unilamellar lipid vesicles (LUV), the suspension of MLV was subjected to the lipid extrusion method, using an Avanti polar lipids extruder. This procedure (https://avantilipids.com/tech-support/liposome-preparation/luvet/) consisted in forcing the lipid suspension through a polycarbonate filter with a defined pore size (in our case 400 nm). The aim was to produce vesicles having the same diameter of the membrane pores. The extrusion was conducted at a temperature of ~50 °C. In this way, vesicles of 400 nm suspended in sucrose buffer were obtained (figure 78).

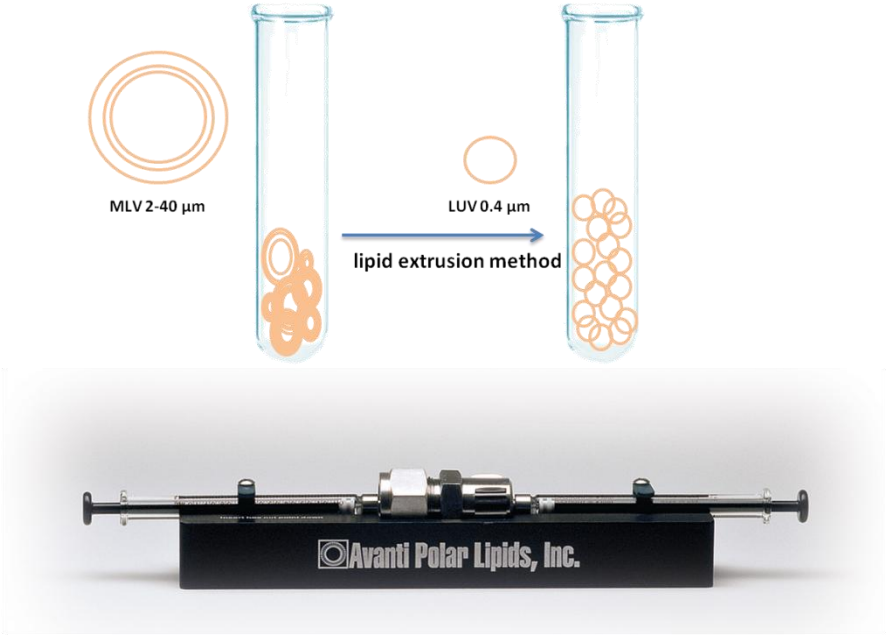


Figure 78 Avanti polar lipids extruder used to generate LUV of 0.4 µM from MLV

To balance the internal and the external environment of the liposome, saline buffer (see paragraph 3.2) was added to the vesicles solution, in a molar ratio of 15:35 liposome solution:saline buffer[8]. The liposome solution was centrifuged at 40.000 rpm for 50 minutes at 25 °C. The supernatant resulting from the centrifugation was then removed and the pellet was recovered in 300 µL of saline buffer. The concentration of lipids was then tested through the Fiske method [152]. At this point, the liposomes were ready for binding experiments. Until their use, the liposomes were retained at a temperature of +4 °C (for up to 48 h).

[8] The saline buffer was added to decrease the sucrose in the outer medium and to avoid the swelling of the vesicles.

7.2 Binding tests: fluorescence and absorbance analyses

7.2.1 Molar ratio peptide:lipids

For the binding tests, a peptide concentration of 25 μM[9] was used. Each peptide was tested on PC liposomes in molar ratios peptide:PC of 1:20 and 1:50, on DMPG liposomes in a molar ratio peptide:DMPG of 1:50, on liposomes of DMPG:CL 3:1 in molar ratios peptide:lipids of 1:20 and 1:50 and, finally, on PC:CL 1:1 liposomes in molar ratios peptide:lipids of 1:20 and 1:50.

7.2.2 Incubation

Peptides and liposomes were incubated for 30 minutes at 25 °C, gently shaking through a shaker-vortex multi piastra.

7.2.3 Fluorescence Measures

Fluorescence analyses were performed immediately after the incubation. 800 μL of the sample were put into a cuvette with an optical path of 1 cm and the sample was excited at a wavelength (λ) of 280 nm. The tryptophan emission spectra were recorded in a range of 300-400 nm at room temperature with a spectrophotometer (Cary Eclipse).

7.2.4 Absorbance Measures

After fluorescence analyses, the samples were prepared for absorbance tests. The idea was to separate the fraction of peptides bound to the target membrane from the unbound fraction. Then, each sample was separated into two ultracentrifuge tubes (400 μL per tube).

[9] I decided to use this concentration value after a series of preliminary tests during which I looked for the ideal concentration to detect the presence of peptide in absorbance and fluorescence analyses.

The ultracentrifuge was set at 25.000 rpm for 1 h at 25 °C. At the end, the supernatant fraction was separated from the pellet fraction. The pellet, which contained liposomes with the bound peptides, was resuspended and recovered in 400 μL of saline buffer and 10X sodium cholate[10]. Then, it was added ethanol (EtOH)[11] to this solution, in a molar ratio of 1:2 saline buffer:ethanol. The supernatant fraction was also treated with 10X sodium cholate and EtOH to work in the same conditions of the pellet fraction. The final volume of each sample was about 1.2 mL. Each sample was slightly agitated and they were put it into a cuvette with an optical path of 1 cm. The absorbance was read at 280 nm. A detailed and illustrated explanation of the protocol can be found in Appendix D.

7.3 Results

7.3.1 Results of fluorescence tests

Each spectrum was analyzed to determine the maximum emission value (λ_{max}). The λ_{max} of the amino acid tryptophan is strongly sensitive to the chemical around. This value depends on the position of the tryptophan. In fact, if the tryptophan is in a polar zone we can see a shift to the visible (red shift). If the environment in which tryptophan is located is hydrophobic, we can see a shift to UV (blue shift). For this reason, a blue shift phenomenon could indicate an interaction between the peptides and the hydrophobic chains of the lipids. The peptides p458, p459 and p460 alone have a maximum fluorescence emission of 356 nm, 355 nm and 361 nm, respectively (figure 79). Therefore, the results in the figure 79 represent the control.

[10] Sodium Cholate is a water-soluble ionic detergent commonly used for membrane protein and lipid isolation, cell lysis and liposome preparation

[11] Ethanol was added to break the possible micelles that sodium cholate could form and which could then interfere with the absorbance measurements

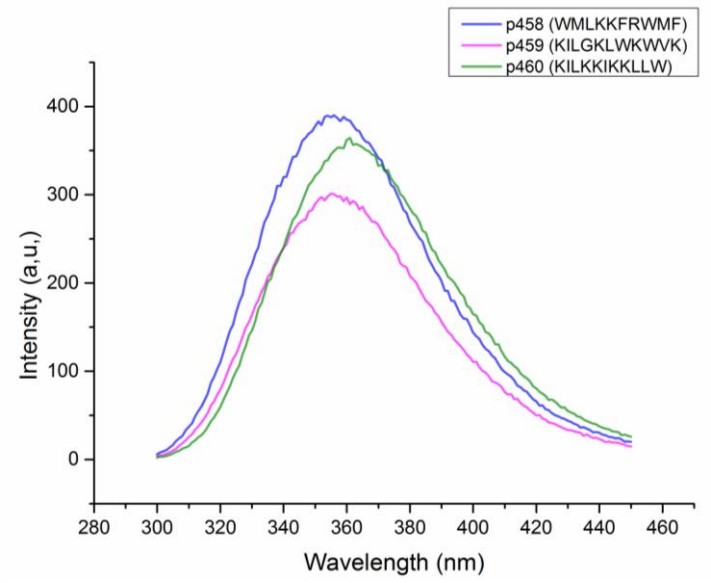


Figure 79 Maximum fluorescence emission of the p458,p459 and p460 peptides alone. They respectively have a λ_{max} of 356 nm, 355 nm and 361 nm

Fluorescence analyses were performed on the solutions in which the peptides were incubated with the target membranes. Therefore, from a spectrum we expect to find two peaks: the λ_{max} of the unbound peptide (control) and the λ_{max} relative to the interaction between the peptide and the target membrane. In fact, only a part of the peptides binds to the membrane while the other remains in solution. Unfortunately, a spectrophotometer fails to be so sensitive and it reveals the result of the sum of smaller peaks. For this reason, a deconvolution analysis on the fluorescence results obtained was performed. The aim was to detect the hidden peaks that are not revealed by the instrument. A deconvolution indicates a correction technique based on the application of a special algorithm. The deconvolution algorithm allows to reconstruct the missing elements on a statistical basis, to remove the noise factors and to create a higher quality image (figure 80).

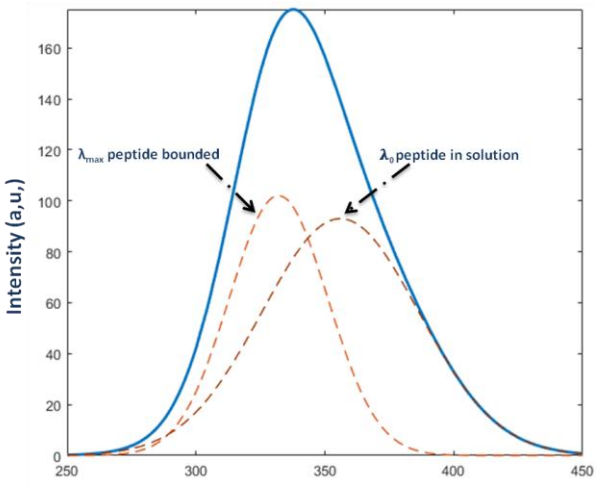


Figure 80 An example of how a deconvolution analysis works

To perform the deconvolution analysis a application of the software Matlab 2017 was used (Curve Fitting Toolbox 3.5.6). Curve Fitting Toolbox provides an app and functions for fitting curves and surfaces to data. After the calculation of all the maximum fluorescence emission values, the variations in fluorescence emission (Δλ) were determined by this formula:

$$eq.37 \quad \Delta\lambda = \lambda_0 - \lambda_{max}$$

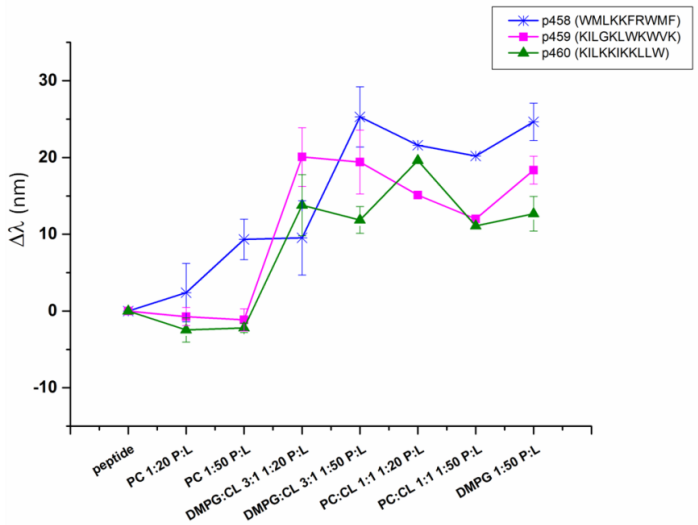


Figure 81 Fluorescence variation (Δλ) that occurs when each peptide interacts with a specific model membrane, at different ratio peptide:lipids.

The parameter λ_0 indicates the maximum fluorescence emission of each peptide, in the absence of lipid vesicles (figure 79). The graph in figure 81 shows the fluorescence variation ($\Delta\lambda$) that occurs when each peptide interacts with a specific model membrane, at different ratio peptide:lipids. Instead, the charts below (figure 82-84) show in detail the λ_{max} of each peptide when it interacts with the different lipid membranes.

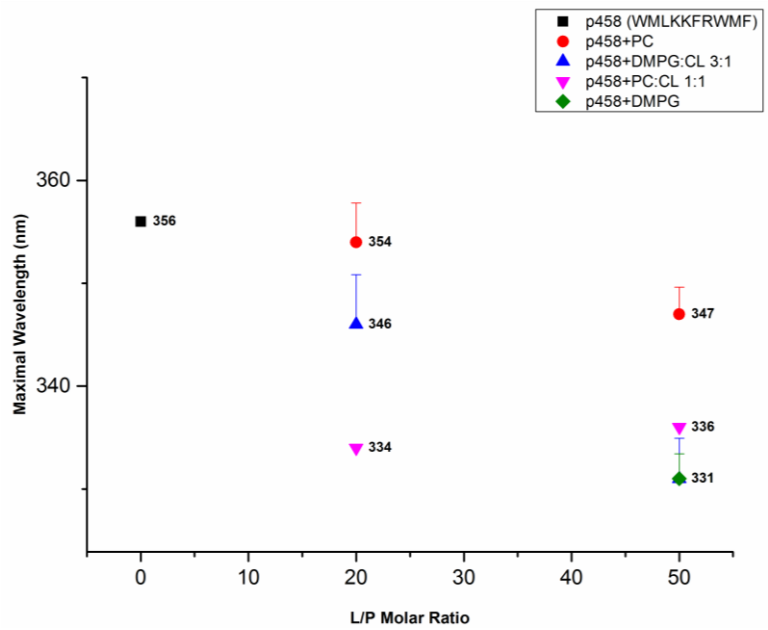


Figure 82 Fluorescence emission of the peptide p458 when it interacts with PC, DMPG, DMPG:CL 3:1 and PC: CL 1:1 vescicles. The maximal wavelength of Trp fluorescence is plotted as a function of L/P molar ratio.

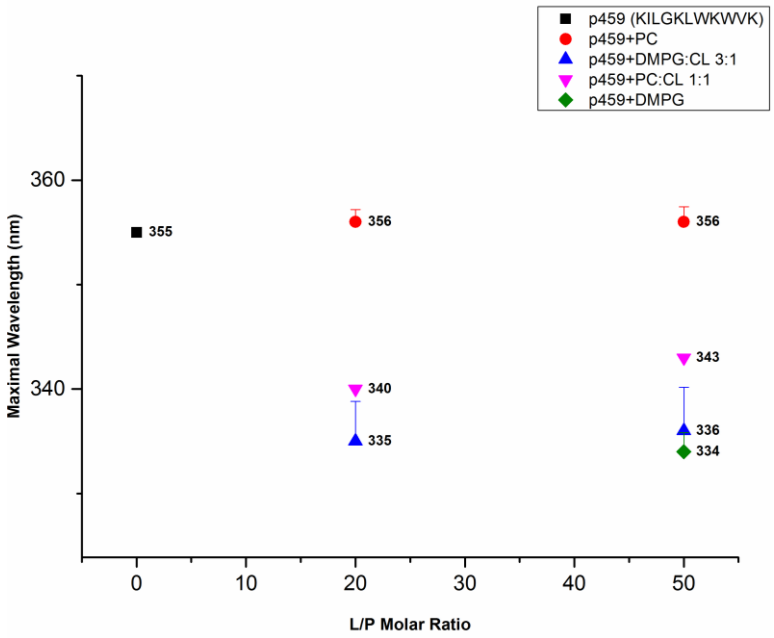


Figure 83 Fluorescence emission of the peptide p459 when it interacts with PC, DMPG, DMPG:CL 3:1 and PC: CL 1:1 vesicles. The maximal wavelength of Trp fluorescence is plotted as a function of L/P molar ratio

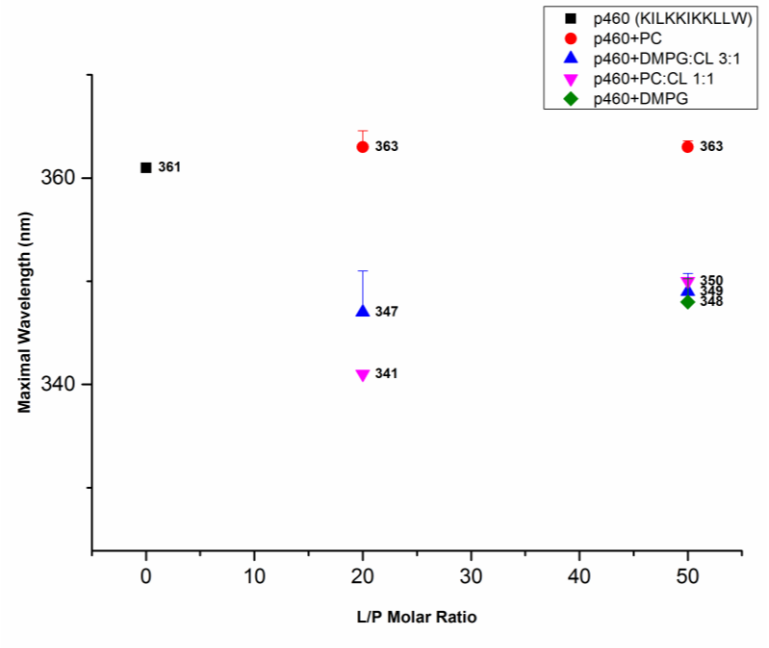


Figure 84 Fluorescence emission of the peptide p460 when it interacts with PC, DMPG, DMPG:CL 3:1 and PC:CL 1:1 vescicles. The maximal wavelength of Trp fluorescence is plotted as a function of L/P molar ratio

7.3.2 Results of absorbance tests

After the fluorescence analysis, the samples were centrifuged to separate membrane-bound peptides (pellet fraction) from unbound peptides, which then remained in suspension (supernatant fraction). The sedimentation rate of suspended particles depends on their size and density. Sucrose buffer (see paragraph 7.1.2) was used to increase the density of the lipid vesicles. In fact, after ultracentrifugation, they all deposit on the bottom of the vial. The aim was to perform absorbance measurements on the pellet and supernatant fractions separately, in order to determine the amount of peptide bound to the liposomes (pellet fraction). The law of Lambert Beer correlates the amount of light absorbed by the sample (A) to the concentration (M) of the sample and to the optical path of the cuvette (l):

$$\boldsymbol{eq.38} \quad \boldsymbol{A = \varepsilon_{\lambda} l M}$$

The parameter $\boldsymbol{\varepsilon_{\lambda}}$ is the molar extinction coefficient, or molar attenuation coefficient. It is a measurement of how strongly a chemical species attenuates light at a given wavelength. It is an intrinsic property of the species. The molar extinction coefficient of the amino acid Tryptophan is 5690 M^{-1} cm^{-1}. The absorption of tryptophan, amino acid present in the sequences of the three AMP studied, was measured when it was excited at 280 nm. The absorption is directly proportional to the concentration of the peptides. The absorbance (Abs) results were normalized by expressing the amount of peptide in the pellets as a percentage of the total material recovered from the starting material (eq.39):

$$eq.39 \quad \%\, peptide_{pellet} = \frac{Abs_{pellet}}{Abs_{pellet} + Abs_{surnatant}}$$

The histogram in the figure 85 shows a complete overview of the absorbance results obtained on the pellet fractions of the samples in which it was performed the binding between each peptide and the target membranes.

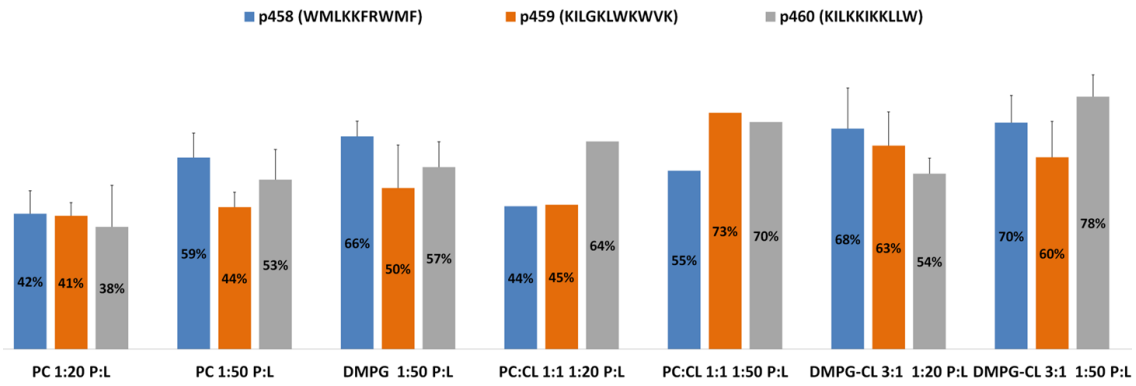


Figure 85 A complete overview of the absorbance results obtained on the samples in which it was performed the binding between each peptide and the target membranes

7.4 Discussion

Absorbance and fluorescence results allow us to formulate the first hypotheses about the behavior of these three candidate drugs. The results of the analyses performed on PC vesicles represent the average of three independent experiments. The peptide p458 alone has a λ_{max} of 356 nm (figure 79). When this peptide interacts with PC membrane in ratios of 1:20 and 1:50 P:L, it has a λ_{max} of 354 nm and 347 nm, respectively (figure 82): there is a slight blue shift and the behavior of this peptide appears to be dependent on the lipid concentration. Absorbance test confirms this aspect: the 42% of the peptide binds PC in a ratio of 1:20 P:L, while the 59% of the peptide binds PC in a ratio 1:50 P:L (figure 85). Due to this direct dependence on the lipid concentration, the idea was to test this peptide even on a higher lipid concentration. The absorbance test reveals that the 60% of the peptide binds the target membrane in a ratio of 1:75 peptide:PC and fluorescence test reveals that the λ_{max} is equal to 347 nm (results not shown). Thus, the peptide p458 has a concentration-dependent behavior against the PC only up to a certain limit (ratio 1:50 P:L), after which its degree of interaction does not change. So, more than half of the p458 interacts with PC membranes in a ratio of 1:50 P:L, but the peptide probably interlaces a little or does not interlace between the hydrophobic lipid chains (the blue shift is very slight). The peptide p459 alone has a λ_{max} of 355 nm (figure 79) and when it interacts with PC membrane in ratios of 1:20 and 1:50 P:L the λ_{max} assumes a value of 356 nm in both cases (figure 83). Absorbance analyses show that the 41% of the peptide binds PC membranes in 1:20 P:L ratio, while the 44% binds PC membranes in a ratio 1:50 P:L (figure 85). This behavior, unlike the p458, does not seem to be affected by the variation in lipid concentration. In fact, the binding between p459 and PC membranes in a ratio of 1:75 P:L does not show a substantial variation (results not shown): the 46% of p459 binds to the PC membrane and the λ_{max} is equal to 360.7 nm. Then, the behavior of the peptide p459 when it contacts PC membranes is not dependent on the lipid concentration and the interaction (just over the 40% of the peptide) is only with the membrane surface: fluorescence analyses do not reveal a blue shift (figure 81).

Absorbance assays performed on the samples in which the binding between the peptide p460 and PC membranes occurred shows a behavior dependent on the lipid concentration. The 38% of the p460 binds PC vesicles in a ratio of 1:20 P:L, while the 53% of p460 binds PC vesicles in a ratio of 1:50 P:L (Figure 85). The 57% of p460 binds PC vesicles in a ratio of 1:75 P:L (result not shown). This behavior does not result in a blue shift in fluorescence analysis. Then, probably, the amount of p460 that interacts with PC membranes is directly proportional to the lipid concentration, but it does not interfere with the hydrophobic lipid chains: fluorescence analyses do not reveal a blue shift (figure 81). Therefore, the peptides p458 and p460 interact with phosphatidylcholine in a way directly proportional to the lipid concentration, but the p458 tends to intercept slightly the hydrophobic chains while the p460 remains on the surface. Instead, the amount of the peptide p459 that binds the PC vesicles does not depend on the lipid concentration (it remains low even at higher lipid concentrations) and it does not seem to interfere with the hydrophobic chains. These preliminary results are very interesting because the PC is the major lipid component of the eukaryotic membranes and we are looking for candidate drugs that are not toxic to living organisms.

The results of the binding between each peptide and DMPG vesicles represent the average of 4 independent experiments. All the fluorescence results obtained on the samples in which each peptide was incubated with DMPG vesicles in a ratio 1:50 P:L show a blue shift in the emission (figure 81). This variation is particularly considerable for p458 and p459: when they interact with DMPG vesicles their λ_{max} decreases by 25% and 20% respectively (figure 82-83). Instead, the peptide p460 decreases its λ_{max} by ~10% when it contacts the DMPG vesicles. The results obtained with the absorbance analyses on the interaction between p458 and DMPG show that the 66% of the peptide binds the DMPG membrane (figure 85). This result, added to the blue shift phenomenon that we observe in the fluorescence analysis (figure 81), suggests that the p458 interacts a lot with DMPG and interferes with the hydrophobic chains. Probably, the p459 also interferes with the hydrophobic chains of the lipids, but with less affinity than the

p458: the absorbance analysis reveals that only the 50% of the peptide binds the DMPG vesicles (figure 85). The 57% of the peptide p460 binds DMPG membranes (figure 85), but this does not result in a high blue shift (figure 81). Probably, the peptide p460 adopts a mechanism of action that need more time to fully intercept the membrane. The interaction of these three cationic peptides with DMPG vesicles negatively charged was foregone: computational and in vitro analyses showed the importance of the *charge* factor. Consequently, the idea was to understand if the charge was a necessary and sufficient property for the binding.

Then, the peptides were tested on membranes with a negative charge higher than the charge of the DMPG vesicles. DMPG:CL 3:1 vesicles were prepared and the binding with p458, p459 and p460 in ratios of 1:20 and 1:50 P:L was performed. The idea to use cardiolipin is due to the fact that previous studies have shown the importance of this lipid in the membrane. Cardiolipin modulates the membrane composition to adapt to stress conditions and membrane fluidity [153]. The results obtained represent the average of 3 independent experiments.

Compared to the binding on the DMPG membranes, among the three peptides only the p460 exhibits a greater affinity when it interacts with the DMPG:CL 3:1 membranes: the 78% of the p460 binds the DMPG:CL 3:1 vesicles against the 57% of the peptide that binds the DMPG vesicles (figure 85). However, the results of the fluorescence analyses are comparable (figure 84): when p460 binds DMPG vesicles in a ratio 1:50 P:L his λ_{max} is equal to 349 nm; when it binds DMPG vesicles at the same ratio P:L the λ_{max} is equal to 348 nm. So, a greater amount of peptide binds the DMPG:CL 3:1 membranes, but the mechanism of action appears the same and involves a slight intercalation between the hydrophobic chains. The peptide p458, on the other hand, is not affected by the increase of the membrane charge. In fact, the amount of the peptide that binds the vesicles of DMPG and the amount of the peptide that binds the DMPG: CL 3:1 (ratio of 1:50 P:L) vesicles are comparable (figure 85). The fluorescence results are also the same: in both cases the λ_{max} is equal to 331 nm (figure 82). This behavior suggests that the

charge is not the only factor involved in the interaction between the p458 and the target membranes, but there are also other factors (for example structural factors).

The peptide p459 interacts a little more with DMPG:CL 3:1 membranes (ratio 1:50 P:L) than with DMPG vesicle: the amount of the peptide linked to the DMPG:CL 3:1 increases by the 10% (figure 85). The results of fluorescence analysis are also comparable (figure 84): when p459 binds DMPG vesicles in a ratio 1:50 P:L his λ_{max} is equal to 336 nm; when it binds DMPG vesicles at the same ratio P:L the λ_{max} is equal to 334 nm. Therefore, like p458, the p459 is not affected by the increase in negative charge of the target membranes and probably there are other factors to consider.

Finally, the peptides were tested on a membrane with the same negative charge of the DMPG to see if the behavior of the peptides on these membranes was the same. Compared to the binding with the DMPG vesicles, a lower amount of p458 binds to PC:CL 1:1 vesicles in a ratio of 1:50 P:L (figure 85). This is confirmed by the fluorescence analysis that shows a slighter blue shift (figure 81). This result, compared with the result on the DMPG:CL 3:1 vesicles, suggests that for the peptide p458 the charge is a necessary but not sufficient factor. A comparable amount of p459 binds the membranes of DMPG and PC:CL 1:1 in a ratio of 1:50 P:L, but this peptide shows less affinity for the DMPG:CL 3:1 membranes, despite the increase in negative charge. Fluorescence analysis, however, suggests that the p459 acts similarly on DMPG and DMPG:CL 3:1 (probably it interferes with the hydrophobic chains), but it acts in a different manner on PC:CL 1:1 membranes. In fact, in this case the blue shift is less marked. The fact that a cationic peptide, in terms of quantity, binds more to a less negative charged membrane, suggests that even in this case the charge is a necessary but not sufficient factor. Therefore, probably, the membrane composition and therefore some structural factors, influences the mechanism of action of these AMP. The p460 peptide interacts in larger amounts with PC:CL 1:1 vesicles than with DMPG vesicles, despite the same negative charge, but fluorescence results are comparable (figure 84). So, it seems that p460 has more affinity for the PC:CL 1:1

vesicles than for the DMPG (probably due to structural factors), but in both cases we see a blue shift that suggests that the peptide interacts with the hydrophobic chains of the lipids (figure 81). The interaction of p460 with DMPG:CL 3:1 vesicles also shows a blue shift and it is comparable with the blue shift that we see after the binding with the other two membranes (figure 81). However, the increased charge of the DMPG:CL 3:1 permits that a greater amount of peptide binds to the membrane: the absorbance analyses show that the 78% of the p460 binds the DMPG:CL 3:1 vesicles (figure 85). So, even in this case the charge is a necessary factor for the interaction but it is not sufficient for the insertion of the peptide into the membrane.

Chapter VIII

8.1 What has been done in this work?

This work includes two approaches: a computational and a preliminary experimental approach. Computational results were the starting point for the subsequent tests in vitro. In 2012, the database Yadamp was created from the research group in which this work was carried out. The main idea was to facilitate the access to important information on AMP, such as the activity and the chemical-physical properties of these molecules. The activity of the peptides was extracted from the literature and the physical-chemical properties calculated by online tools or by Matlab scripts. In this work, an important contribution was given to improve the Yadamp database. When Yadamp was created it contained 2133 sequences of antimicrobial peptides. During this work, 1009 new sequences of AMP were manually extracted from the scientific literature. For these sequences 573 chemical-physical parameters were calculated. In this regard, this work also intersects with another parallel project that involved the creation of a new molecular docking system: Yada. The idea was to study the interaction of AMP with other peptides, protein receptors and target membranes in terms of binding energy.

The philosophy behind Yadamp was to permit QSAR analyses and the creation of activity model against pathogenic microorganisms. Yadamp allowed the creation of homogeneous subsets of AMP: the hypothesis was that peptides with the same chemical-physical characteristics shared the same mechanism of action against target microorganisms. In this work, new computational prediction procedures have been employed and allowed to generate many activity models against pathogenic microorganisms Gram + and Gram - (Chapter III-IV-V). They were implemented in the "Yadamp predict" tool (http://yadamp.unisa.it/predict.aspx). It allows researchers to submit sequences of unknown molecules and to see if and to which organisms these molecules are potentially active. Users can also know the degree of reliability of their results through appropriate statistical validation

systems. The “Yadamp predict” tool suggested three amino acid sequences that potentially could bind Gram + bacteria (Chapter VII). These sequences were chosen on their diversity in the amino acid sequence, on the consensus that they obtained from different algorithms and on the presence of the amino acid tryptophan in their sequence. By exploiting the known spectrophotometric properties of the tryptophan, the interaction of these peptides with vesicles of 400 nm with different lipid composition was evaluated through fluorescence and absorbance analyses. The preliminary results suggest that the interaction of these three candidate sequences with the target liposomes does not depend only on the charge parameter, but probably also on structural changes of the membranes due to the lipid polymorphism. It is known that lipids, when forming a membrane, can be assembled into a variety of phases with different geometry. It depends on their chemical structure and also on external variables, such as temperature or pressure. This feature influences different cellular processes [154]. We can hypothesize a structural rearrangement of the membrane caused by the interaction with the peptides. Cardiolipin (CL), for example, is a phospholipid with two phosphate groups and four acyl chains [155]. The small size of the polar group of CL increases the propensity to form non-lamellar inverted phases. This tendency, however, is attenuated by the presence of negative charges of the mutually repulsive phosphate groups [156].

8.2 What will be done?

Starting from the preliminary results obtained in silico and in vitro, the idea is to repeat these analyses and, also, to prove the activity of these peptides by microbiological tests. At the same time, molecular dynamics simulations of peptide/membrane systems are in progress. The aim is to clarify the mechanism of action of AMP. Furthermore, the software Yada will permit the study of the interaction of AMP with other peptides, protein receptors and target membranes (Chapter VI). The calculation of the binding energy is also important to enrich the

pool of molecular descriptors available and to obtain even more efficient prediction results.

8.3 Conclusions

Drug discovery is the long and very complex process by which new drugs are found. It requires many years of research, experimental tests, clinical studies and a high economic capital. It is necessary to wait 12-14 years before having a new drug on the market. The identification of new potentially active molecules is certainly a delicate step. Usually, a series of possible candidates (lead compounds) is available. The large amount and the heterogeneity of these substances makes this process long and expensive. The identification of a lead compound and its optimization can take up to three years of work. An approach that can shorten the research time and the optimization of a drug candidate is the study of the quantitative structure-activity relationship (QSAR). It is the search of a relationship between the three-dimensional structure of a molecule and its bioactivity. The computational approach has speeded up the lead optimization process by multiple degrees in the last two decades. This is a profitable operation that allows to test thousands of compounds by a priori rejecting unattractive compounds and reducing the number of possible candidates. However, it is not easy to get the optimal results through the QSAR analysis, due to the preparation of the data and to poor application of statistical methods. For example, an optimal QSAR analysis needs an adequately sized data set. The real challenge is to verify if the predictions will come true or not: it is not obvious that what is calculated through computational techniques gives a positive result. The initial idea was to identify correlations between the structure and the activity of AMP to clarify their mechanism of action and to design new active and selective molecules. However, this approach is possible only working on homogeneous datasets of AMP and looking for new molecular descriptors. Many techniques, including PCA and cluster analysis, have been used to find homogeneous datasets of AMP (Chapter V). Furthermore, a new docking system was created to study the interaction of

these molecules with the target membranes in terms of binding energy (Chapter VI). In fact, the descriptors 1D and 2D currently available fail to capture all of the peptides properties. The peptides are extremely flexible molecules and undergo conformational changes: in water, their structure is hydrophilic; when they interact with membranes, they expose a hydrophobic region.

The optimization of computational analyses significantly reduces the time of the drug discovery process. More generally, this work gives guidelines on the automation of a large part of the drug discovery process to proceed towards in vitro and in vivo experimentation in a more targeted way.

All together, these findings support the proposed mechanism of action of the 3 peptides and pave the way for novel and more focused design of antimicrobial peptides.

Appendices

Appendix A
Matlab Scripts

Calculate_Charge.m

```
% This script computes the charge at three different pH 5, 7, 9
using pKa values taken from Lehninger Principles of Biochemistry.
load pKa_value.mat
fid = fopen('AMP.txt', 'r');
fid2 =fopen('AMP_Charge.txt','w');
fprintf(fid2, 'Calculation of charge at pH5,7 and 9\n');
fprintf(fid2, 'sequence             pH5         pH7
pH9\n');
string = fgetl(fid)
while string ~= -1
    val5=0;
    val7=0;
    val9=0;
    peptide_length = length(string);
    for j = 1:peptide_length
        switch string(j)
            case ('K')
                val5=val5 + 10.^10.5/(10.^10.5+ 10^5);
                val7=val7 + 10.^10.5/(10.^10.5+ 10^7);
                val9=val9 + 10.^10.5/(10.^10.5+ 10^9);
            case ('R')
                val5=val5 + 10.^12.4/(10.^12.4+ 10^5);
                val7=val7 + 10.^12.4/(10.^12.4+ 10^7);
                val9=val9 + 10.^12.4/(10.^12.4+ 10^9);
            case ('H')
                val5=val5 + 10.^6/   (10.^6+ 10^5);
                val7=val7 + 10.^6/   (10.^6+ 10^7);
                val9=val9 + 10.^6/   (10.^6+ 10^9);
            case ('Y')
                val5=val5 - 10.^5/(10.^10+ 10^5);
                val7=val7 - 10.^7/(10.^10+ 10^7);
                val9=val9 - 10.^9/(10.^10+ 10^9);
            case ('D')
                val5=val5 - 10.^5/(10.^3.86+ 10^5);
                val7=val7 - 10.^7/(10.^3.86+ 10^7);
                val9=val9 - 10.^9/(10.^3.86+ 10^9);
            case ('E')
                val5=val5 - 10.^5/(10.^4.25+ 10^5);
                val7=val7 - 10.^7/(10.^4.25+ 10^7);
                val9=val9 - 10.^9/(10.^4.25+ 10^9);
            case ('C')
                val5=val5 - 10.^5/(10.^8.33+ 10^5);
                val7=val7 - 10.^7/(10.^8.33+ 10^7);
                val9=val9 - 10.^9/(10.^8.33+ 10^9);
        end;
    end;
    val5 = val5 + 10.^9.69/(10.^9.69 + 10^5)- 10.^5/(10.^2.34+
10^5)
    val7 = val7 + 10.^9.69/(10.^9.69 + 10^7)- 10.^7/(10.^2.34+
10^7);
    val9 = val9 + 10.^9.69/(10.^9.69 + 10^9)- 10.^9/(10.^2.34+
10^9);
    fprintf(fid2, '%s\t\t\t%f\t%f\t%f\n', string, val5, val7,
val9);
```

```
    string = fgetl(fid);
end;
fclose('all');
Calculate_Boman_index.m
% This script computes the Boman index in accordingly with Journal
of Internal Medicine 2003; 254: 197-215
fid = fopen('AMP.txt', 'r');
fid2 = fopen('AMP_BomanIndex.txt','w');
fprintf(fid2, 'Boman index of alpha AMPs\n\n');
string = fgetl(fid);
while string ~= -1
    peptide_lenght = length(string);
    temp = 0;
    bindex=0;
    for j = 1: peptide_lenght
%         mono = 1;
            first = char(string(j));
            switch first
                case ('W')
                    aminoacidA = 2.33;
                case ('C')
                    aminoacidA = 1.28;
                case ('M')
                    aminoacidA = 2.35;
                case ('H')
                    aminoacidA = -4.66;
                case ('Y')
                    aminoacidA = -0.14;
                case ('F')
                    aminoacidA = 2.98;
                case ('Q')
                    aminoacidA = -5.54;
                case ('N')
                    aminoacidA = -6.64;
                case ('I')
                    aminoacidA = 4.92;
                case ('R')
                    aminoacidA = -14.92;
                case ('D')
                    aminoacidA = -8.72;
                case ('P')
                    aminoacidA = 0;
                case ('T')
                    aminoacidA = -2.57;
                case ('K')
                    aminoacidA = -5.55;
                case ('E')
                    aminoacidA = -6.81;
                case ('V')
                    aminoacidA = 4.04;
                case ('S')
                    aminoacidA = -3.40;
                case ('G')
                    aminoacidA = 0.94;
                case ('A')
                    aminoacidA = 1.81;
                case ('L')
                    aminoacidA = 4.92;
            end;
            temp = temp + aminoacidA;
    end;
    bindex = -temp/ peptide_lenght;
    fprintf(fid2, '%f\t%s\n', bindex, string);
    string = fgetl(fid);
```

```
end;
fclose('all');
```

Calculate_HydrophobicityMoment_Flexibility.m

```
% This script computes the mean hydrophobicity and the hydrophobic
moment in accordingly with Faraday Symp. Chem. Soc., 1982, 17, 109-
120.
% In addition it computes the Flexibility value in accordingly with
Angew. Chem. Int. Ed. 2003, 42, 2269 – 2272.
load string_Hydrophobicity
fid = fopen('AMP.txt', 'r');
fid2 =fopen('AMP_HydrophobicMoment.txt','w');
fid3 =fopen('AMP_Flexibility.txt','w');
fprintf(fid2, 'Hydrophobic moments of alpha AMPs\n');
fprintf(fid2, 'length      h_mom_a     mean_hm_a   h_mom_b
mean_hm_b   h_mom_c     mean_hm_c\n');
fprintf(fid3, 'Flexibility of alpha AMPs\n');

string = fgetl(fid);
while string ~= -1
    hxa=0;
    hxb=0;
    hxc=0;
    hya=0;
    hyb=0;
    hyc=0;
    flex=0;
    peptide_lenght = length(string);
    for j = 1:peptide_lenght
        s= sin(1.7453*j);
        c= cos(1.7453*j);
        switch string(j)
            case ('A')
                hxa=hxa+c*values(1,1);
                hya=hya+s*values(1,1);
                hxb=hxb+c*values(1,2);
                hyb=hyb+s*values(1,2);
                hxc=hxc+c*values(1,3);
                hyc=hyc+s*values(1,3);
                flex= flex+18;
            case ('R')
                hxa=hxa+c*values(2,1);
                hya=hya+s*values(2,1);
                hxb=hxb+c*values(2,2);
                hyb=hyb+s*values(2,2);
                hxc=hxc+c*values(2,3);
                hyc=hyc+s*values(2,3);
                flex= flex+4.6;
            case ('N')
                hxa=hxa+c*values(3,1);
                hya=hya+s*values(3,1);
                hxb=hxb+c*values(3,2);
                hyb=hyb+s*values(3,2);
                hxc=hxc+c*values(3,3);
                hyc=hyc+s*values(3,3);
                flex= flex+20;
            case ('D')
                hxa=hxa+c*values(4,1);
                hya=hya+s*values(4,1);
                hxb=hxb+c*values(4,2);
                hyb=hyb+s*values(4,2);
```

```
    hxc=hxc+c*values(4,3);
    hyc=hyc+s*values(4,3);
    flex= flex+20;
case ('C')
    hxa=hxa+c*values(5,1);
    hya=hya+s*values(5,1);
    hxb=hxb+c*values(5,2);
    hyb=hyb+s*values(5,2);
    hxc=hxc+c*values(5,3);
    hyc=hyc+s*values(5,3);
    flex= flex+20;   % estimated by comparison with Ser
case ('Q')
    hxa=hxa+c*values(6,1);
    hya=hya+s*values(6,1);
    hxb=hxb+c*values(6,2);
    hyb=hyb+s*values(6,2);
    hxc=hxc+c*values(6,3);
    hyc=hyc+s*values(6,3);
    flex= flex+7.2;
case ('E')
    hxa=hxa+c*values(7,1);
    hya=hya+s*values(7,1);
    hxb=hxb+c*values(7,2);
    hyb=hyb+s*values(7,2);
    hxc=hxc+c*values(7,3);
    hyc=hyc+s*values(7,3);
    flex= flex+8.2;
case ('G')
    hxa=hxa+c*values(8,1);
    hya=hya+s*values(8,1);
    hxb=hxb+c*values(8,2);
    hyb=hyb+s*values(8,2);
    hxc=hxc+c*values(8,3);
    hyc=hyc+s*values(8,3);
    flex= flex+39;
case ('H')
    hxa=hxa+c*values(9,1);
    hya=hya+s*values(9,1);
    hxb=hxb+c*values(9,2);
    hyb=hyb+s*values(9,2);
    hxc=hxc+c*values(9,3);
    hyc=hyc+s*values(9,3);
    flex= flex+4.8;
case ('I')
    hxa=hxa+c*values(10,1);
    hya=hya+s*values(10,1);
    hxb=hxb+c*values(10,2);
    hyb=hyb+s*values(10,2);
    hxc=hxc+c*values(10,3);
    hyc=hyc+s*values(10,3);
    flex= flex+2.3;
case ('L')
    hxa=hxa+c*values(11,1);
    hya=hya+s*values(11,1);
    hxb=hxb+c*values(11,2);
    hyb=hyb+s*values(11,2);
    hxc=hxc+c*values(11,3);
    hyc=hyc+s*values(11,3);
    flex= flex+10;
case ('K')
    hxa=hxa+c*values(12,1);
    hya=hya+s*values(12,1);
    hxb=hxb+c*values(12,2);
    hyb=hyb+s*values(12,2);
```

```
                hxc=hxc+c*values(12,3);
                hyc=hyc+s*values(12,3);
                flex= flex+3.4;
            case ('M')
                hxa=hxa+c*values(13,1);
                hya=hya+s*values(13,1);
                hxb=hxb+c*values(13,2);
                hyb=hyb+s*values(13,2);
                hxc=hxc+c*values(13,3);
                hyc=hyc+s*values(13,3);
                flex = flex+ 7;  % intermediate value between L and
K
              case ('F')
                hxa=hxa+c*values(14,1);
                hya=hya+s*values(14,1);
                hxb=hxb+c*values(14,2);
                hyb=hyb+s*values(14,2);
                hxc=hxc+c*values(14,3);
                hyc=hyc+s*values(14,3);
                flex= flex+7.6;
            case ('P')
                hxa=hxa+c*values(15,1);
                hya=hya+s*values(15,1);
                hxb=hxb+c*values(15,2);
                hyb=hyb+s*values(15,2);
                hxc=hxc+c*values(15,3);
                hyc=hyc+s*values(15,3);
                flex= flex+0.1;
            case ('S')
                hxa=hxa+c*values(16,1);
                hya=hya+s*values(16,1);
                hxb=hxb+c*values(16,2);
                hyb=hyb+s*values(16,2);
                hxc=hxc+c*values(16,3);
                hyc=hyc+s*values(16,3);
                flex= flex+25;
            case ('T')
                hxa=hxa+c*values(17,1);
                hya=hya+s*values(17,1);
                hxb=hxb+c*values(17,2);
                hyb=hyb+s*values(17,2);
                hxc=hxc+c*values(17,3);
                hyc=hyc+s*values(17,3);
                flex= flex+11;
            case ('W')
                hxa=hxa+c*values(18,1);
                hya=hya+s*values(18,1);
                hxb=hxb+c*values(18,2);
                hyb=hyb+s*values(18,2);
                hxc=hxc+c*values(18,3);
                hyc=hyc+s*values(18,3);
                flex = flex+8; % estimated by comparison with Phe
            case ('Y')
                hxa=hxa+c*values(19,1);
                hya=hya+s*values(19,1);
                hxb=hxb+c*values(19,2);
                hyb=hyb+s*values(19,2);
                hxc=hxc+c*values(19,3);
                hyc=hyc+s*values(19,3);
                flex = flex+8; % estimated by comparison with Phe
            case ('V')
                hxa=hxa+c*values(20,1);
                hya=hya+s*values(20,1);
```

```
                hxb=hxb+c*values(20,2);
                hyb=hyb+s*values(20,2);
                hxc=hxc+c*values(20,3);
                hyc=hyc+s*values(20,3);
                flex= flex+3;
            end;
        end;
        flex = flex/peptide_lenght;
        hydro_moment_a = sqrt(hxa*hxa + hya*hya);
        mean_hydro_moment_a = hydro_moment_a/peptide_lenght;
        hydro_moment_b = sqrt(hxb*hxb + hyb*hyb);
        mean_hydro_moment_b = hydro_moment_b/peptide_lenght;
        hydro_moment_c = sqrt(hxc*hxc + hyc*hyc);
        mean_hydro_moment_c = hydro_moment_c/peptide_lenght;
        fprintf(fid2,
'%d\t\t\t%f\t%f\t%f\t%f\t%f\t%f\t%s\n',peptide_lenght,
hydro_moment_a, mean_hydro_moment_a, hydro_moment_b,
mean_hydro_moment_b, hydro_moment_c, mean_hydro_moment_c, string);
        fprintf(fid3, '%f\n', flex);
        string = fgetl(fid);
    end;
    fclose('all');
```

Parsing_Helicity_prediction.m

```
% This script produces the Helicity prediction values after
computing by DSC algorithm presents in Discovery Studio from
Accelrys.
fid = fopen('Yadamp-DscPrediction.txt', 'r');
fid2 =fopen('AMP_Helicitiy.txt','w');
fprintf(fid2, 'Helicity prediction\n\n');
string = fgetl(fid);
number = 0;
j=1;
flag = 0;
tot = 0;
while string ~= -1
    string = fgetl(fid);
    peptide_lenght = 0;
    if string
        if string(1) == 'D'
            string = fgetl(fid);
            for i= 14:67
                number = double(string(i));
                if number <= 32
                    flag =0;

                else
                    peptide_lenght = peptide_lenght +1;
                        tot = tot + number-48;
                    flag = 0;
                end
            end
            tot = tot/ (peptide_lenght);
            fprintf(fid2, '%f\t%s\n', tot, string);
            tot =0;
            flag = 0;
        end
    end
end
fclose('all');
```

Calculate_InstabilityIndex.m

```
% This script computes the Instability index in accordingly with
Protein Engineering vol.4 no.2 pp.155-161. 1990.
load aminoacids_instability.mat
load instability_matrix.mat
fid = fopen('AMP.txt', 'r');
fid2 =fopen('AMP_InstabilityIndex.txt','w');
fprintf(fid2, 'Instability index of alpha AMPs\n\n');
string = fgetl(fid);

while string ~= -1
    iindex=0;
    temp=0;

    peptide_lenght = length(string);

    dimers = nmercount(string, 2);
    dimers_number = length(dimers);

for j = 1:dimers_number

            dimer  = sscanf(char(dimers(j)), '%c');
            first = char(dimer(1));
            switch first
                case ('W')
                    aminoacidA = 1;
                case ('C')
                    aminoacidA = 2;
                case ('M')
                    aminoacidA = 3;
                case ('H')
                    aminoacidA = 4;
                case ('Y')
                    aminoacidA = 5;
                case ('F')
                    aminoacidA = 6;
                case ('Q')
                    aminoacidA = 7;
                case ('N')
                    aminoacidA = 8;
                case ('I')
                    aminoacidA = 9;
                case ('R')
                    aminoacidA = 10;
                case ('D')
                    aminoacidA = 11;
                case ('P')
                    aminoacidA = 12;
                case ('T')
                    aminoacidA = 13;
                case ('K')
                    aminoacidA = 14;
                case ('E')
                    aminoacidA = 15;
                case ('V')
                    aminoacidA = 16;
                case ('S')
                    aminoacidA = 17;
                case ('G')
                    aminoacidA = 18;
                case ('A')
```

```
                    aminoacidA = 19;
                case ('L')
                    aminoacidA = 20;
            end;
            temp(1) = aminoacidA;
            second   = char(dimer(2));
            switch second
                case ('W')
                    aminoacidA = 1;
                case ('C')
                    aminoacidA = 2;
                case ('M')
                    aminoacidA = 3;
                case ('H')
                    aminoacidA = 4;
                case ('Y')
                    aminoacidA = 5;
                case ('F')
                    aminoacidA = 6;
                case ('Q')
                    aminoacidA = 7;
                case ('N')
                    aminoacidA = 8;
                case ('I')
                    aminoacidA = 9;
                case ('R')
                    aminoacidA = 10;
                case ('D')
                    aminoacidA = 11;
                case ('P')
                    aminoacidA = 12;
                case ('T')
                    aminoacidA = 13;
                case ('K')
                    aminoacidA = 14;
                case ('E')
                    aminoacidA = 15;
                case ('V')
                    aminoacidA = 16;
                case ('S')
                    aminoacidA = 17;
                case ('G')
                    aminoacidA = 18;
                case ('A')
                    aminoacidA = 19;
                case ('L')
                    aminoacidA = 20;
            end;
            temp(2) = aminoacidA;
        iindex = iindex + INSTMATRIX(temp(1),temp(2));
    end;
    iindex = 10/peptide_lenght*iindex;

    fprintf(fid2, '%f\t%s\n', iindex, string);
    string = fgetl(fid);

end;
fclose('all');
```

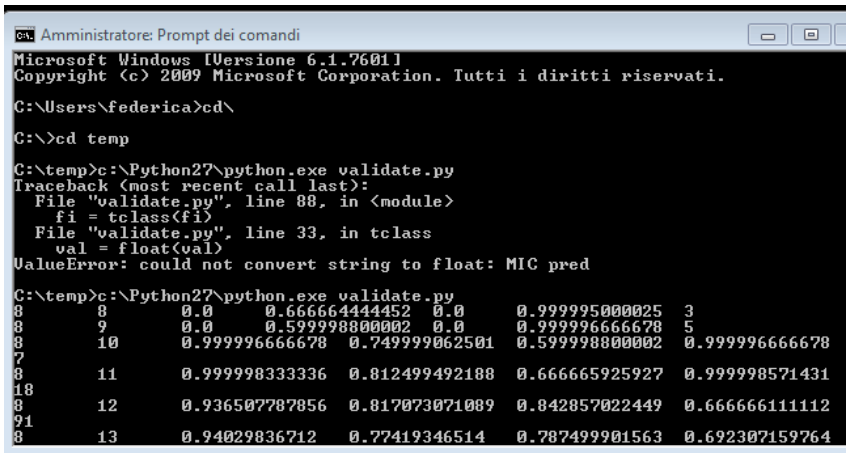
Appendix B

Statistical validation of the models obtained: calculation of accuracy, precision, sensitivity and specificity parameters and calculation of the index score

```
len_start = 8
len_stop = 83
scoremat =
for i in range(1,6):
scoremat[i] =
scoremat[1][1] = 2
scoremat[1][2] = 1
scoremat[1][3] = 0
scoremat[1][4] = -1
scoremat[1][5] = -2
scoremat[2][1] = 2
scoremat[2][2] = 2
scoremat[2][3] = 1
scoremat[2][4] = 0
scoremat[2][5] = -1
scoremat[3][1] = 0
scoremat[3][2] = 1
scoremat[3][3] = 1
scoremat[3][4] = 0
scoremat[3][5] = -1
scoremat[4][1] = -1
scoremat[4][2] = 0
scoremat[4][3] = 0
scoremat[4][4] = 1
scoremat[4][5] = 0
scoremat[5][1] = -2
scoremat[5][2] = -1
scoremat[5][3] = -1
scoremat[5][4] = 0
scoremat[5][5] = 2

def tclass(val):
val = float(val)
if (val<=2):
      return 1
if (val >2 and val <=5):
      return 2
if (val >5 and val <=10):
      return 3
if (val >10 and val<=30):
      return 4
if (val > 30):
      return 5

for k in range(8,84):
```

```
for i in range(k,84):

#statistiche
      TN = 0
      TP = 0
      FP = 0
      FN = 0
      PRECISION=0
      SENSITIVITY=0
      ACCURACY=0
      SPECIFICITY = 0
      SCORE=0
#media valori osservati sperimentali
      mv = 0
      r2 = 0
      numerator = 0
      denominator = 0
      denom = []
      numpep = 0
      #print "7\t",i,"\t",
      #print "Len 7 =>",i,"Num pep:",
      #open file
      inp = open("input.txt")
      while 1:
            line = inp.readline();
            line = line.replace("\n","")
            if (line==""):
                  break;
            seq = line.split("\t")[0]
            l = len(seq)
            if (l>=k and l <=i):
                  lenw = line.split("\t")[1]
                  yi= line.split("\t")[2]
                  fi= line.split("\t")[3]
                  fi = fi.replace(",",".")
                  yi = yi.replace(",",".")
                  fi = tclass(fi)
                  yi = tclass(yi)
                  mv +=float(yi)
                  numpep += 1
                  numerator += pow(float(yi) - float(fi),2)
                  denom.append(yi)
                  #print yi,fi

                  gmic = yi
                  gpredict = fi

                  SCORE += scoremat[fi][yi]

                  #attivo (gruppo a*,a,b,c)
                  if (gmic <= 4):
                        if (gpredict<=4): #active predicted
as active
                              TP+=1
                        else:
                              FN+=1       #active predicted
as not active
                        pass
```

```
                  if (gmic >= 5):
                        if (gpredict>=5): #not active
predicted as not active
                              TN+=1
                        else:
                              FP+=1       #not active
predicted as active
      if (numpep==0):
            continue;
      mv = mv/numpep
      for p in denom:
            denominator += pow(float(p)-mv,2)
      PRECISION = float(TP)/ (float(TP)+float(FP)+0.00001)

      SENSITIVITY= float(TP)/ (float(TP)+float(FN)+0.00001)
      ACCURACY= (float(TP)+float(TN))/ (numpep+0.00001)
      SPECIFICITY= float(TN)/ (float(TN)+float(FP)+0.00001)
      SCORE = SCORE
      #attivare queste due righe per calcolare lo score
      print k,"\t",i,"\t",SCORE
      print i,"\t",k,"\t",SCORE
      #attivare la riga seguente per precision etc...
      #print
k,"\t",i,"\t",PRECISION,"\t",ACCURACY,"\t",SENSITIVITY,"\t",
SPECIFICITY
#print numpep,"R2=",1-
(numerator/denominator),"TP:",TP,"TN:",TN,"FP:",FP,"FN:",FN
      #print "\tPRECISION:",PRECISION,
      #print "\tSENSITIVITY:",SENSITIVITY,
      #print "\tACCURACY:",ACCURACY,
      #print "\tSPECIFICITY:",SPECIFICITY
inp.close()
```




```
C:\temp>c:\Python27\python.exe validate.py > fileoutput.txt
```

Appendix C

List of 25 PDB used for vibrational analysis

1dgf	1dgg	1dgh	2wwo	2wwn
3whs	2pfk	1pfk	3b8d	3tu9
1pv7	4b5z	4b60	2anc	2f3r
1gsd	1gsf	4qs8	4qs7	3whr
1ruz	1rvt	1d4c	1d4e	1pv6

List of 305 PDB used for Yada calibration

1a07	1a0q	1a1b	1a1e	1a28	1a42	1a4g	1a4k	1a4q	1a6w	1a9u
1aaq	1abe	1abf	1acj	1acl	1acm	1aco	1aec	1aha	1ai5	1aj7
1ake	1aoe	1apt	1apu	1aqw	1ase	1atl	1azm	1b58	1b59	1b6n
1b9v	1baf	1bbp	1bgo	1bl7	1blh	1bma	1bmq	1byb	1byg	1c12
1c1e	1c2t	1c5c	1c5x	1c83	1cbs	1cbx	1cdg	1cf8	1cil	1cin
1ckp	1cle	1com	1coy	1cps	1cqp	1ctr	1ctt	1cvu	1cx2	1d0l
1d3h	1d4p	1dbb	1dbj	1dbm	1dd7	1dg5	1dhf	1did	1die	1dmp
1dog	1dr1	1dwb	1dwc	1dwd	1dy9	1eap	1ebg	1eed	1ei1	1ejn
1ela	1elb	1elc	1eld	1ele	1eoc	1epb	1epo	1eta	1etr	1ets
1ett	1etz	1f0r	1f0s	1f3d	1fax	1fbl	1fen	1fgi	1fig	1fkg
1fki	1fl3	1flr	1frp	1ghb	1glp	1glq	1gpy	1hak	1hdc	1hdy
1hef	1hfc	1hiv	1hos	1hpv	1hri	1hsb	1hsl	1htf	1hti	1hvr
1hyt	1ibg	1icn	1ida	1igj	1imb	1ivb	1ivc	1ivd	1ive	1ivq
1jao	1jap	1kel	1kno	1lah	1lcp	1ldm	1lic	1lkk	1lmo	1lna
1mmq	1mnc	1mrg	1mrk	1mts	1mtw	1mup	1nco	1ngp	1nis	1nsd
1okl	1okm	1pbd	1pdz	1pgp	1pha	1phd	1phf	1phg	1poc	1ppc
1pph	1ppi	1ppl	1pso	1ptv	1qbr	1qbt	1qbu	1pph	1qh7	1ql7

1qpe	1qpq	1rbp	1rds	1rne	1rnt	1rob	1rt2	1sln	1slt	1snc
1srf	1srg	1srh	1srj	1stp	1tdb	1tka	1tlp	1tmn	1tng	1tnh
1tni	1tnl	1tph	1tpp	1trk	1tyl	1ukz	1ulb	1uvs	1uvt	1vgc
1vrh	1wap	1xid	1xie	1xkb	1ydr	1yds	1ydt	1yee	25c8	2aad
2ack	2ada	2ak3	2cgr	2cht	2cmd	2cpp	2ctc	2dbl	2er7	2fox
2gbp	2h4n	2ifb	2lgs	2mcp	2mip	2pcp	2phh	2pk4	2plv	2qwk
2r04	2r07	2sim	2tmn	2tsc	2yhx	2ypi	3cla	3cpa	3erd	3ert
3gpb	3gch	3hvt	3mth	3nos	3pgh	3ptb	3tpi	4aah	4cox	4cts
4dfr	4er2	4est	4fab	4fbp	4lbd	4phv	4tpi	5abp	5cpp	5er1
5p2p	6abp	6cpa	6rnt	6rsa	7cpa	7tim	8gch	1lpm	1lyb	1mmb
1lst	1lyl	1mbi	1mcq	1mcr	1mdr	1ml1	1mld	1qcf	1pph	

List of 126 PDB from the Astex set used for the validation

pdb	RMSD VINA (Å)	RMSD YADA (Å)
1a28	18.95	27.43
1a6w	0.52	2.15
1a9u	0.47	0.89
1abe	2.52	1.02
1abf	21.97	4.47
1acj	23.88	1.39
1acl	32.85	1.24
1aec	15.52	5.66
1aha	16.32	1.13
1aj7	4.53	23.06
1ake	21.13	24.38
1aoe	0.31	0.32
1apt	1.21	1.93
1apu	0.96	0.71
1b59	18.77	1.59
1bbp	34.57	21.58
1bgo	5.61	2.01
1bl7	0.36	0.81
1blh	2.7	1.67
1bmq	2.99	3.51
1byb	24.68	0.37
1c12	20.34	2.27
1c2t	26.81	21.79
1c5c	5.11	4.78
1c5x	0.67	1.19
1c83	28.91	26.51
1cbs	0.54	0.72
1cle	4.78	13.71
1coy	15.86	14.04

1cqp	7.12	0.51
1cvu	21.13	24.22
1d3h	12.67	10.02
1d4p	0.48	0.51
1dbb	1.14	1.11
1dbj	2.77	31.64
1dbm	0.39	0.51
1dg5	5.09	6.22
1dhf	5.14	6.86
1die	3.28	22.61
1dmp	19.05	0.62
1dog	4.97	1.94
1dwb	0.67	9.61
1dwc	1.03	0.42
1dwd	1.4	0.43
1eap	0.81	0.84
1ebg	28.43	20.7
1ejn	11.86	11.52
1epb	13.96	1.52
1eta	26.44	26.94
1etr	0.9	0.58
1ets	0.35	9.44
1ett	6.58	1.87
1etz	31.05	5.15
1fen	21.36	0.36
1fgi	59.91	8.73
1fig	3.48	3.48
1fkg	0.41	0.4
1fki	35	13.39
1flr	18.73	5.92
1glp	8.74	0.74
1glq	12.99	1.81
1gpy	4.22	3.11
1hak	5.68	8.06
1hos	10.55	0.46
1hpv	6.05	0.58
1htf	12.99	7.51
1hvr	7.61	1.79
1ida	0.22	0.4

1igj	7.02	5.34
1kel	23.32	32.34
1ldm	2.93	7.89
1lic	4.64	0.45
1lmo	4.52	1.41
1mdr	1.03	1.48
1mrg	5.03	1.1
1nco	0.41	0.54
1ngp	30.06	2.44
1pbd	0.11	0.15
1pgp	18.09	19.77
1qbr	5.93	0.09
1qbt	11.99	0.39
1qbu	11.97	0.32
1qcf	2.9	3.14
1qh7	0.37	1.52
1rbp	16.08	0.34
1rds	2.3	0.32
1rne	4.16	0.42
1rt2	41.87	4.54
1srf	1.57	2.83
1srg	1.75	0.48
1srh	2.66	7.49
1srj	26.35	0.84
1stp	0.57	0.8
1tph	4.19	5.86
1ulb	14.59	1.27
1uvs	0.98	8.52
1uvt	0.64	1.14
1vrh	44.47	3.23
1ydr	30.21	4.49
1yds	9.33	0.27
1yee	17.72	21.09
25c8	56.31	19.85
2ack	9.84	1.33
2cgr	60.13	32.33
2cmd	0.18	0.38
2dbl	4.57	3.87
2ifb	2.64	0.65

2mcp	15.3	2.85
2pcp	18.19	19.22
2pk4	0.3	0.34
2plv	32.61	8.47
2r04	46.3	6.27
2sim	0.35	0.74
2tsc	3.55	25.4
2yhx	1.89	8.11
2ypi	1.85	2.43
3ert	0.7	0.49

3gch	2.86	1.15
3gpb	25.24	1.89
4cts	0.22	17.54
4fab	4.45	4.18
4lbd	16.2	0.23
4phv	0.27	0.21
5abp	10.05	9.44
6abp	22.49	2.97
7tim	5.95	4.27

Correlation between experimental binding energy and Yada calculation

pdb	Experimental binding energy	Predicted values
1AAQ	11.5	6.8
1BB0	11.4	7.3
1BMN	11.5	8.2
1BWB	11.5	10.7
1CEA	6.76	4.6
1D4J	11.4	9.7
1FQ5	11.5	10.1
1FWU	5.0	6.3
1FWV	5.0	5.6
1G7V	8.7	7.3
1GWW	4.8	6.9
1GX0	5.7	6.8
1GX8	8.7	7.2
1GZC	4.7	6.3
1H61	5.9	7.7
1IT6	11.4	8.5
1K21	11.4	7.5
1K22	11.5	8.3
1LZQ	11.4	7.8
1M2R	8.8	9.4
1MU6	11.4	8.9
1SRE	5.2	7.2
1T7R	8.1	9.9
1TOJ	4.6	8.4
1TSY	6.7	8.2
1UTJ	5.2	8.7
1VJC	4.8	7.4
1W5X	11.5	10.1
1Y2F	6.7	7.0
1YSG	4.8	8.1
1ZC9	4.3	8.0
1ZPA	11.5	8.2
2IKO	7.4	8.3
2IWS	8.7	8.6
2J77	6.6	7.0
2J95	11.5	9.0
2NMY	11.4	9.6
2NMZ	11.4	9.3
2NNK	11.4	9.5
2NNO	8.7	7.2
2NNP	11.4	9.9
2OIQ	6.8	10.3
2P3I	4.7	5.9
2UWL	11.5	8.5
2V00	4.9	7.5
2VSL	11.5	5.5
2VT3	5.6	7.0
2VXN	6.8	7.3
2W0S	4.5	7.3
2W47	6.7	7.9
2WIB	6.7	8.2
2WJ2	11.4	6.6
2XGS	4.4	7.9

2XYE	11.4	8.5
2XYF	11.5	8.5
2Y82	11.5	10.4
2ZXA	11.4	8.1
3A1C	5.4	7.4
3ARQ	8.7	9.6
3ATV	7.2	8.2
3B24	5.9	9.0
3B68	11.5	9.8
3BXG	6.7	8.4
3C2O	4.5	8.8
3CFN	6.8	7.1
3CJ5	8.6	7.6
3D0E	11.5	8.6
3D1V	4.7	7.3
3O83	11.4	10.5
3EOS	11.5	8.6
3F33	4.5	3.8
3F34	4.3	4.0
3F35	4.3	3.6
3F82	11.4	10.1
3HAU	8.7	7.7
3OXC	11.5	9.3
3PJT	6.6	9.4
3PJU	6.6	9.7
3Q7Q	7.7	8.7
3QBC	6.7	8.7
3QLM	8.7	8.5
3QX9	5.5	7.2
3S76	6.8	8.1
3SW2	11.4	9.8
3TCP	11.4	7.8
3U7S	11.5	8.1
3UUG	6.1	8.2
3V2P	8.7	7.3
3W07	4.6	7.6
3ZK6	8.6	8.9
4A6B	11.4	9.4
4AGD	11.5	8.3
4B32	4.5	6.1
4B33	4.5	5.7
4BAO	11.4	8.2
4BUP	6.7	7.8
4DMW	6.7	7.5
4DY6	6.0	8.6
4E6Q	11.4	9.3
4F1Q	6.7	6.4
4F3H	8.7	8.2

4G3E	11.4	9.5
4G3F	11.4	9.2
4G3G	11.5	8.2
4GBY	4.7	6.9
4HPI	6.6	6.3
4I67	8.7	6.7
4IPJ	6.6	6.8
4J7E	6.6	7.4
4JFL	6.7	7.9
4LBP	6.7	8.0
4LIL	6.6	7.2
9HVP	11.4	8.8
1AJ7	5.3	5.9
1AU2	10.8	11.1
1BXQ	10.1	9.1
1C83	6.6	8.8
1CET	3.9	7.6
1CPS	9.1	8.4
1DMP	13.0	10.8
1EED	6.5	8.6
1EPO	10.9	9.6
1FAX	10.1	9.4
1FIG	8.5	6.1
1G2K	10.9	10.6
1HIH	11.0	9.5
1HOS	11.7	9.1
1IVP	10.3	7.9
1LBF	10.7	8.6
1LOQ	5.0	8.1
1LYX	6.2	8.7
1M0O	3.2	7.3
1NJ5	10.0	9.4
1P6E	4.0	7.7
1QB6	8.3	9.1
1QBN	8.0	9.3
1TFT	11.3	7.6
1TNK	2.0	8.2
1TNL	2.6	8.9
1TOM	11.3	9.2
1UJ5	4.2	8.7
1W31	4.9	8.7
1WVJ	9.2	9.5
1XS7	10.4	7.9
1YYY	6.9	11.1
2BPV	10.5	9.5
2BXU	9.8	8.7
2E9U	11.1	9.8
2E9V	10.8	9.2

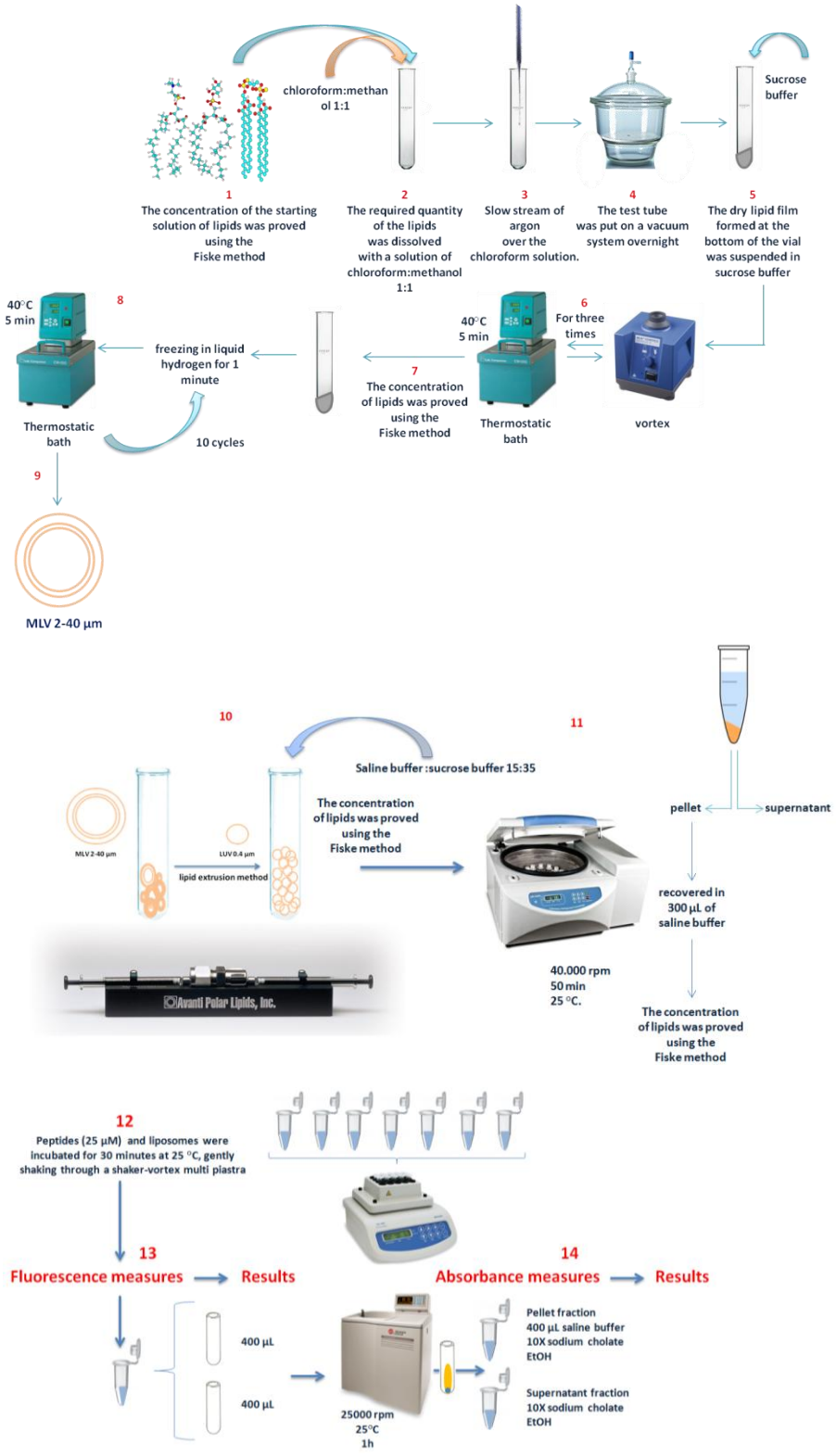
2V2C	4.7	9.3
3D1Y	11.2	10.9
3EOU	9.6	8.9
3ERT	13.1	9.3
3MI2	10.6	9.5
3PRS	10.7	9.3
3VFQ	7.1	9.0

4AYY	11.2	10.0
4BT5	3.8	7.8
4FJZ	8.6	9.3
4GR3	10.7	7.6
4TIM	2.9	8.8

Friedman LOF	24.1
R-squared	0.24
Adjusted R-squared	0.23
Cross validated R-squared	0.23
Significance-of-regression F-value	24.5
Critical SOR F-value (95%)	4.11

Appendix D

Experimental protocol



Bibliography

1. Thomsen, T.T., *Peptide Antibiotics for ESKAPE Pathogens: Past, Present and Future Perspectives of Antimicrobial Peptides for the Treatment of Serious Gram-Negative and Gram-Positive Infections*, 2016, Department of Biology, Faculty of Science, University of Copenhagen.
2. Livermore, D., *The need for new antibiotics.* Clinical Microbiology and Infection, 2004. **10**(s4): p. 1-9.
3. Deng, T., et al., *The heterologous expression strategies of antimicrobial peptides in microbial systems.* Protein Expression and Purification, 2017. **140**: p. 52-59.
4. Hwang, P.M. and H.J. Vogel, *Structure-function relationships of antimicrobial peptides.* Biochemistry and Cell Biology, 1998. **76**(2-3): p. 235-246.
5. Bechinger, B. and S.-U. Gorr, *Antimicrobial Peptides: mechanisms of action and resistance.* Journal of dental research, 2017. **96**(3): p. 254-260.
6. Ong, P.Y., et al., *Endogenous antimicrobial peptides and skin infections in atopic dermatitis.* New England Journal of Medicine, 2002. **347**(15): p. 1151-1160.
7. Zasloff, M., *Magainins, a class of antimicrobial peptides from Xenopus skin: isolation, characterization of two active forms, and partial cDNA sequence of a precursor.* Proceedings of the National Academy of Sciences, 1987. **84**(15): p. 5449-5453.
8. Capone, R., et al., *Antimicrobial protegrin-1 forms ion channels: molecular dynamic simulation, atomic force microscopy, and electrical conductance studies.* Biophysical journal, 2010. **98**(11): p. 2644-2652.
9. Rozek, A., C.L. Friedrich, and R.E. Hancock, *Structure of the bovine antimicrobial peptide indolicidin bound to dodecylphosphocholine and sodium dodecyl sulfate micelles.* Biochemistry, 2000. **39**(51): p. 15765-15774.
10. Mandard, N., et al., *Solution structure of thanatin, a potent bactericidal and fungicidal insect peptide, determined from proton two-dimensional nuclear magnetic resonance data.* The FEBS Journal, 1998. **256**(2): p. 404-410.
11. Wakabayashi, N., et al., *A pH-dependent charge reversal peptide for cancer targeting.* European Biophysics Journal, 2017. **46**(2): p. 121-127.
12. Semrau, S., et al., *Membrane lysis by gramicidin S visualized in red blood cells*

and giant vesicles. Biochimica et Biophysica Acta (BBA)-Biomembranes, 2010. **1798**(11): p. 2033-2039.

13. Sani, M.-A. and F. Separovic, *How membrane-active peptides get into lipid membranes.* Accounts of chemical research, 2016. **49**(6): p. 1130-1138.
14. Perez Espitia, P.J., et al., *Bioactive peptides: synthesis, properties, and applications in the packaging and preservation of food.* Comprehensive Reviews in Food Science and Food Safety, 2012. **11**(2): p. 187-204.
15. Matsuzaki, K., et al., *Relationship of membrane curvature to the formation of pores by magainin 2.* Biochemistry, 1998. **37**(34): p. 11856-11863.
16. Zhang, M., J. Zhao, and J. Zheng, *Molecular understanding of a potential functional link between antimicrobial and amyloid peptides.* Soft Matter, 2014. **10**(38): p. 7425-7451.
17. Yeaman, M.R. and N.Y. Yount, *Mechanisms of antimicrobial peptide action and resistance.* Pharmacological reviews, 2003. **55**(1): p. 27-55.
18. Marr, A.K., W.J. Gooderham, and R.E. Hancock, *Antibacterial peptides for therapeutic use: obstacles and realistic outlook.* Current opinion in pharmacology, 2006. **6**(5): p. 468-472.
19. Hu, Y., et al., *Dissemination of the mcr-1 colistin resistance gene.* The Lancet infectious diseases, 2016. **16**(2): p. 146-147.
20. Moffatt, J.H., et al., *Colistin resistance in Acinetobacter baumannii is mediated by complete loss of lipopolysaccharide production.* Antimicrobial agents and chemotherapy, 2010. **54**(12): p. 4971-4977.
21. Otto, M., *Bacterial evasion of antimicrobial peptides by biofilm formation*, in *Antimicrobial peptides and human disease*. 2006, Springer. p. 251-258.
22. Nikaido, H., *Multidrug efflux pumps of gram-negative bacteria.* Journal of bacteriology, 1996. **178**(20): p. 5853.
23. Aoki, W. and M. Ueda, *Characterization of antimicrobial peptides toward the development of novel antibiotics.* Pharmaceuticals, 2013. **6**(8): p. 1055-1081.
24. Bacalum, M. and M. Radu, *Cationic antimicrobial peptides cytotoxicity on mammalian cells: an analysis using therapeutic index integrative concept.* International Journal of Peptide Research and Therapeutics, 2015. **21**(1): p. 47-55.
25. Maloy, W.L. and U.P. Kari, *Structure–activity studies on magainins and other*

host defense peptides. Biopolymers, 1995. **37**(2): p. 105-122.

26. Seo, M.-D., et al., *Antimicrobial peptides for therapeutic applications: a review.* Molecules, 2012. **17**(10): p. 12276-12286.
27. Kreil, G., *D-amino acids in animal peptides.* Annual review of biochemistry, 1997. **66**(1): p. 337-345.
28. Yin, L.M., et al., *Differential binding of L-vs. D-isomers of cationic antimicrobial peptides to the biofilm exopolysaccharide alginate.* Protein and peptide letters, 2013. **20**(8): p. 843-847.
29. Tew, G.N., et al., *De novo design of biomimetic antimicrobial polymers.* Proceedings of the National Academy of Sciences, 2002. **99**(8): p. 5110-5114.
30. Mensa, B., et al., *Antibacterial mechanism of action of arylamide foldamers.* Antimicrobial agents and chemotherapy, 2011. **55**(11): p. 5043-5053.
31. Park, I.Y., et al., *Helix stability confers salt resistance upon helical antimicrobial peptides.* Journal of Biological Chemistry, 2004. **279**(14): p. 13896-13901.
32. Li, Y., *Recombinant production of antimicrobial peptides in Escherichia coli: a review.* Protein expression and purification, 2011. **80**(2): p. 260-267.
33. Ashby, M., A. Petkova, and K. Hilpert, *Cationic antimicrobial peptides as potential new therapeutic agents in neonates and children: a review.* Current opinion in infectious diseases, 2014. **27**(3): p. 258-267.
34. Todeschini, R. and V. Consonni, *Handbook of molecular descriptors*. Vol. 11. 2008: John Wiley & Sons.
35. Cherkasov, A., et al., *QSAR modeling: where have you been? Where are you going to?* Journal of medicinal chemistry, 2014. **57**(12): p. 4977-5010.
36. Kawashima, S., et al., *AAindex: amino acid index database, progress report 2008.* Nucleic acids research, 2007. **36**(suppl_1): p. D202-D205.
37. Hammami, R., et al., *PhytAMP: a database dedicated to antimicrobial plant peptides.* Nucleic acids research, 2008. **37**(suppl_1): p. D963-D968.
38. Hammami, R., et al., *BACTIBASE second release: a database and tool platform for bacteriocin characterization.* Bmc Microbiology, 2010. **10**(1): p. 22.
39. Seshadri Sundararajan, V., et al., *DAMPD: a manually curated antimicrobial peptide database.* Nucleic acids research, 2011. **40**(D1): p. D1108-D1112.
40. Wang, G., X. Li, and Z. Wang, *APD3: the antimicrobial peptide database as a tool*

for research and education. Nucleic acids research, 2016. **44**(D1): p. D1087-D1093.

41. Thomas, S., et al., *CAMP: a useful resource for research on antimicrobial peptides.* Nucleic acids research, 2009. **38**(suppl_1): p. D774-D780.
42. Guaní-Guerra, E., et al., *Antimicrobial peptides: general overview and clinical implications in human health and disease.* Clinical Immunology, 2010. **135**(1): p. 1-11.
43. Piotto, S.P., et al., *YADAMP: yet another database of antimicrobial peptides.* International journal of antimicrobial agents, 2012. **39**(4): p. 346-351.
44. UniProtKB/Swiss-Prot. Available from: http://www.uniprot.org/uniprot
45. Wang, G., X. Li, and Z. Wang, *APD2: the updated antimicrobial peptide database and its application in peptide design.* Nucleic acids research, 2008. **37**(suppl_1): p. D933-D937.
46. NCBI. Available from: http://www.ncbi.nlm.nih.gov/taxonomy.
47. Nelson, D.L., A.L. Lehninger, and M.M. Cox, *Lehninger principles of biochemistry*. 2008: Macmillan.
48. Gasteiger, E., et al., *Protein identification and analysis tools on the ExPASy server*. 2005: Springer.
49. Bjellqvist, B., et al., *Reference points for comparisons of two-dimensional maps of proteins from different human cell types defined in a pH scale where isoelectric points correlate with polypeptide compositions.* Electrophoresis, 1994. **15**(1): p. 529-539.
50. Boman, H., *Antibacterial peptides: basic facts and emerging concepts.* Journal of internal medicine, 2003. **254**(3): p. 197-215.
51. Radzicka, A. and R. Wolfenden, *Comparing the polarities of the amino acids: side-chain distribution coefficients between the vapor phase, cyclohexane, 1-octanol, and neutral aqueous solution.* Biochemistry, 1988. **27**(5): p. 1664-1670.
52. Eisenberg, D., et al. *Hydrophobic moments and protein structure*. in *Faraday Symposia of the Chemical Society*. 1982. Royal Society of Chemistry.
53. King, R.D. and M.J. Sternberg, *Identification and application of the concepts important for accurate and reliable protein secondary structure prediction.* Protein science, 1996. **5**(11): p. 2298-2310.
54. Huang, F. and W.M. Nau, *A conformational flexibility scale for amino acids in*

peptides. Angewandte Chemie International Edition, 2003. **42**(20): p. 2269-2272.

55. Guruprasad, K., B.B. Reddy, and M.W. Pandit, *Correlation between stability of a protein and its dipeptide composition: a novel approach for predicting in vivo stability of a protein from its primary sequence.* Protein Engineering, Design and Selection, 1990. **4**(2): p. 155-161.
56. Holton, T.A., et al., *CPPpred: prediction of cell penetrating peptides.* Bioinformatics, 2013. **29**(23): p. 3094-3096.
57. Pirtskhalava, M., et al., *DBAASP v. 2: an enhanced database of structure and antimicrobial/cytotoxic activity of natural and synthetic peptides.* Nucleic acids research, 2016. **44**(D1): p. D1104-D1112.
58. Liu, S., et al., *Computational resources and tools for antimicrobial peptides.* Journal of Peptide Science, 2016.
59. Lata, S., B. Sharma, and G. Raghava, *Analysis and prediction of antibacterial peptides.* BMC bioinformatics, 2007. **8**(1): p. 263.
60. Scheetz, T., et al., *Genomics-based approaches to gene discovery in innate immunity.* Immunological reviews, 2002. **190**(1): p. 137-145.
61. Ghahramani, Z., *An introduction to hidden Markov models and Bayesian networks.* International journal of pattern recognition and artificial intelligence, 2001. **15**(01): p. 9-42.
62. Holland, J.H., *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. 1992: MIT press.
63. Leardi, R., R. Boggia, and M. Terrile, *Genetic algorithms as a strategy for feature selection.* Journal of chemometrics, 1992. **6**(5): p. 267-281.
64. Parisi, R., et al. *Models for the Prediction of Antimicrobial Peptides Activity*. in *Italian Workshop on Artificial Life and Evolutionary Computation*. 2015. Springer.
65. Jain, A.K., J. Mao, and K.M. Mohiuddin, *Artificial neural networks: A tutorial.* Computer, 1996. **29**(3): p. 31-44.
66. McCulloch, W.S. and W. Pitts, *A logical calculus of the ideas immanent in nervous activity.* The bulletin of mathematical biophysics, 1943. **5**(4): p. 115-133.
67. Mynski, M. and S. Papert, *Perceptrons: An introduction to Computational Geometry.* MA: MIT Press, Cambridge, 1969.

68. Wang, S.-C., *Artificial neural network*, in *Interdisciplinary computing in java programming*. 2003, Springer. p. 81-100.
69. Khaled, K. and N. Abdel-Shafi, *Quantitative structure and activity relationship modeling study of corrosion inhibitors: Genetic function approximation and molecular dynamics simulation methods.* International Journal of Electrochemical Science, 2011. **6**: p. 4077-4094.
70. Accelrys, Accelrys Materials Studio. 2014, San Diego, California:Accelrys Inc.
71. MATLAB, V., *8.1. 0.604 (R2013a).* MathWorks, Natick, MA, 2013.
72. Accelrys, Accelrys Materials Studio. Accelrys Inc., San Diego, California (2014)
73. Kluytmans, J., A. Van Belkum, and H. Verbrugh, *Nasal carriage of Staphylococcus aureus: epidemiology, underlying mechanisms, and associated risks.* Clinical microbiology reviews, 1997. **10**(3): p. 505-520.
74. Tong, S.Y., et al., *Staphylococcus aureus infections: epidemiology, pathophysiology, clinical manifestations, and management.* Clinical microbiology reviews, 2015. **28**(3): p. 603-661.
75. Chambers, H.F., *The changing epidemiology of Staphylococcus aureus?* Emerging infectious diseases, 2001. **7**(2): p. 178.
76. Chen, L., et al., *How the antimicrobial peptides kill bacteria: computational physics insights.* Communications in Computational Physics, 2012. **11**(3): p. 709-725.
77. Paterson, D.J., et al., *Lipid topology and electrostatic interactions underpin lytic activity of linear cationic antimicrobial peptides in membranes.* Proceedings of the National Academy of Sciences, 2017. **114**(40): p. E8324-E8332.
78. Szuba, T.M., *Computational collective intelligence*. 2001: John Wiley & Sons, Inc.
79. Walker, J.D., et al., *Guidelines for developing and using quantitative structure-activity relationships.* Environmental Toxicology and Chemistry, 2003. **22**(8): p. 1653-1665.
80. Cronin, M.T. and T.W. Schultz, *Pitfalls in QSAR.* Journal of Molecular Structure: THEOCHEM, 2003. **622**(1): p. 39-51.
81. Dearden, J., M. Cronin, and K. Kaiser, *How not to develop a quantitative structure–activity or structure–property relationship (QSAR/QSPR).* SAR and QSAR in Environmental Research, 2009. **20**(3-4): p. 241-266.
82. Chirico, N. and P. Gramatica, *Real external predictivity of QSAR models: how to*

evaluate it? Comparison of different validation criteria and proposal of using the concordance correlation coefficient. Journal of chemical information and modeling, 2011. **51**(9): p. 2320-2335.

83. Gramatica, P., *On the development and validation of QSAR models.* Computational Toxicology: Volume II, 2013: p. 499-526.
84. Netzeva, T.I., et al., *Current status of methods for defining the applicability domain of (quantitative) structure-activity relationships.* ATLA, 2005. **33**: p. 155-173.
85. Anastasakis, L. and N. Mort, *The development of self-organization techniques in modelling: a review of the group method of data handling (GMDH).* RESEARCH REPORT-UNIVERSITY OF SHEFFIELD DEPARTMENT OF AUTOMATIC CONTROL AND SYSTEMS ENGINEERING, 2001.
86. Ivakhnenko, A.G., *The group method of data handling, a rival of the method of stochastic approximation.* Soviet Automatic Control, 1968. **13**(3): p. 43-55.
87. Ivakhnenko, A., *Heuristic self-organization in problems of engineering cybernetics.* Automatica, 1970. **6**(2): p. 207-219.
88. Tenaillon, O., et al., *The population genetics of commensal Escherichia coli.* Nature Reviews Microbiology, 2010. **8**(3): p. 207-217.
89. Vogt, R.L. and L. Dippold, *Escherichia coli O157: H7 outbreak associated with consumption of ground beef, June–July 2002.* Public health reports, 2005. **120**(2): p. 174-178.
90. Nakashima, H., K. Nishikawa, and T. Ooi, *Distinct character in hydrophobicity of amino acid compositions of mitochondrial proteins.* Proteins: Structure, Function, and Bioinformatics, 1990. **8**(2): p. 173-178.
91. Wertz, D.H. and H.A. Scheraga, *Influence of water on protein structure. An analysis of the preferences of amino acid residues for the inside or outside and for specific conformations in a protein molecule.* Macromolecules, 1978. **11**(1): p. 9-15.
92. Wilce, M.C., M.-I. Aguilar, and M.T. Hearn, *Physicochemical basis of amino acid hydrophobicity scales: evaluation of four new scales of amino acid hydrophobicity coefficients derived from RP-HPLC of peptides.* Analytical chemistry, 1995. **67**(7): p. 1210-1219.

93. Bhaskaran, R. and P. Ponnuswamy, *Positional flexibilities of amino acid residues in globular proteins.* Chemical Biology & Drug Design, 1988. **32**(4): p. 241-255.

94. Charton, M. and B.I. Charton, *The dependence of the Chou-Fasman parameters on amino acid side chain structure.* Journal of theoretical biology, 1983. **102**(1): p. 121-134.

95. Charton, M., *Protein folding and the genetic code: an alternative quantitative model.* Journal of theoretical biology, 1981. **91**(1): p. 115-123.

96. Finkelstein, A., A.Y. Badretdinov, and O. Ptitsyn, *Physical reasons for secondary structure stability: α-Helices in short peptides.* Proteins: Structure, Function, and Bioinformatics, 1991. **10**(4): p. 287-299.

97. Qian, N. and T.J. Sejnowski, *Predicting the secondary structure of globular proteins using neural network models.* Journal of molecular biology, 1988. **202**(4): p. 865-884.

98. Aurora, R. and G.D. Rosee, *Helix capping.* Protein Science, 1998. **7**(1): p. 21-38.

99. Meirovitch, H., S. Rackovsky, and H. Scheraga, *Empirical studies of hydrophobicity. 1. Effect of protein size on the hydrophobic behavior of amino acids.* Macromolecules, 1980. **13**(6): p. 1398-1405.

100. Nakashima, H. and K. Nishikawa, *The amino acid composition is different between the cytoplasmic and extracellular sides in membrane proteins.* FEBS letters, 1992. **303**(2-3): p. 141-146.

101. Oobatake, M., Y. Kubota, and T. Ooi, *Optimization of Amino Acid Parameters for Correspondence of Sequence to Tertiary Structures of Proteins (Commemoration Issue Dedicated to Professor Eiichi Fujita on the Occasion of his Retirement).* 1985.

102. PALAU, J., P. ARGOS, and P. PUIGDOMENECH, *Protein secondary structure.* Chemical Biology & Drug Design, 1982. **19**(4): p. 394-401.

103. Eisenberg, D. and A.D. McLachlan, *Solvation energy in protein folding and binding.* Nature, 1986. **319**(6050): p. 199-203.

104. Wolfenden, R., P. Cullis, and C. Southgate, *Water, protein folding, and the genetic code.* Science, 1979. **206**(4418): p. 575-577.

105. Guy, H.R., *Amino acid side-chain partition energies and distribution of residues in soluble proteins.* Biophysical journal, 1985. **47**(1): p. 61-70.

106. Bigelow, C. and M. Channon, *Handbook of Biochemistry and Molecular Biology*

3rd. ed: Proteins (Fassman, GD, ed.) Vol. 1, 1976, CRC Press, Clevelend.

107. Geisow, M.J. and R.D. Roberts, *Amino acid preferences for secondary structure vary with protein class.* International Journal of Biological Macromolecules, 1980. **2**(6): p. 387-389.

108. Grantham, R., *Amino acid difference formula to help explain protein evolution.* Science, 1974. **185**(4154): p. 862-864.

109. Hopp, T.P. and K.R. Woods, *Prediction of protein antigenic determinants from amino acid sequences.* Proceedings of the National Academy of Sciences, 1981. **78**(6): p. 3824-3828.

110. Jones, D.T., W.R. Taylor, and J.M. Thornton, *The rapid generation of mutation data matrices from protein sequences.* Bioinformatics, 1992. **8**(3): p. 275-282.

111. Meek, J.L. and Z.L. Rossetti, *Factors affecting retention and resolution of peptides in high-performance liquid chromatography.* Journal of Chromatography A, 1981. **211**(1): p. 15-28.

112. NISHIKAWA, K. and T. OOI, *PREDICTION OF THE SURFACE-INTERIOR DIAGRAM OF GLOBULAR PROTEINS BY AN EMPIRICAL METHOD.* Chemical Biology & Drug Design, 1980. **16**(1): p. 19-32.

113. Nozaki, Y. and C. Tanford, *The solubility of amino acids and two glycine peptides in aqueous ethanol and dioxane solutions establishment of a hydrophobicity scale.* Journal of Biological Chemistry, 1971. **246**(7): p. 2211-2217.

114. Von Heune, G. and C. Blomberg, *Trans-membrane translocation of proteins. The direct transfer model.* Eur. J. Biochem, 1979. **97**: p. 175-181.

115. Mitaku, S., T. Hirokawa, and T. Tsuji, *Amphiphilicity index of polar amino acids as an aid in the characterization of amino acid preference at membrane–water interfaces.* Bioinformatics, 2002. **18**(4): p. 608-616.

116. Zhou, H. and Y. Zhou, *Quantifying the effect of burial of amino acid residues on protein stability.* PROTEINS: Structure, Function, and Bioinformatics, 2004. **54**(2): p. 315-322.

117. Harpaz, Y., M. Gerstein, and C. Chothia, *Volume changes on protein folding.* Structure, 1994. **2**(7): p. 641-649.

118. Miyazawa, S. and R.L. Jernigan, *Self-consistent estimation of inter-residue protein contact energies based on an equilibrium mixture approximation of*

residues. Proteins: Structure, Function, and Bioinformatics, 1999. **34**(1): p. 49-68.

119. FAUCHÈRE, J.L., et al., *Amino acid side chain parameters for correlation studies in biology and pharmacology.* Chemical Biology & Drug Design, 1988. **32**(4): p. 269-278.

120. Cohn, E. and J. Edsall, *Proteins, Amino Acids and Peptides (Reinhold, New York, 1943).* Google Scholar: p. 184.

121. Crawford, J.L., W.N. Lipscomb, and C.G. Schellman, *The reverse turn as a polypeptide conformation in globular proteins.* Proceedings of the National Academy of Sciences, 1973. **70**(2): p. 538-542.

122. Yutani, K., Ogasahara, K., Tsujita, T., & Sugino, Y. (1987). *Dependence of conformational stability on hydrophobicity of the amino acid residue in a series of variant proteins substituted at a unique position of tryptophan synthase alpha subunit.* Proceedings of the National Academy of Sciences, **84**(13), 4441-4444.

123. Bastolla, U., et al., *Principal eigenvector of contact matrices and hydrophobicity profiles in proteins.* Proteins: Structure, Function, and Bioinformatics, 2005. **58**(1): p. 22-30.

124. Kjaer, J., et al., *Prediction of phenotypic susceptibility to antiretroviral drugs using physiochemical properties of the primary enzymatic structure combined with artificial neural networks.* HIV medicine, 2008. **9**(8): p. 642-652.

125. Janin, J., *Surface and inside volumes in globular proteins.* Nature, 1979. **277**(5696): p. 491.

126. Cid, H., et al., *Hydrophobicity and structural classes in proteins.* Protein Engineering, Design and Selection, 1992. **5**(5): p. 373-375.

127. Naderi-Manesh, H., et al., *Prediction of protein surface accessibility with information theory.* Proteins: Structure, Function, and Bioinformatics, 2001. **42**(4): p. 452-459.

128. Castanho, M. and N. Santos, *Peptide drug discovery and development: translational research in academia and industry*. 2011: John Wiley & Sons.

129. Kanehisa, M.I. and T.Y. Tsong, *Local hydrophobicity stabilizes secondary structures in proteins.* Biopolymers, 1980. **19**(9): p. 1617-1628.

130. Levitt, M., *A simplified representation of protein conformations for rapid simulation of protein folding.* Journal of molecular biology, 1976. **104**(1): p. 59-107.

131. Ponnuswamy, P., M. Prabhakaran, and P. Manavalan, *Hydrophobic packing and spatial arrangement of amino acid residues in globular proteins.* Biochimica et Biophysica Acta (BBA)-Protein Structure, 1980. **623**(2): p. 301-316.

132. Robson, B. and E. Suzuki, *Conformational properties of amino acid residues in globular proteins.* Journal of molecular biology, 1976. **107**(3): p. 327-356.

133. Vihinen, M., E. Torkkila, and P. Riikonen, *Accuracy of protein flexibility predictions.* Proteins: Structure, Function, and Bioinformatics, 1994. **19**(2): p. 141-149.

134. George, R.A. and J. Heringa, *An analysis of protein domain linkers: their classification and role in protein folding.* Protein Engineering, Design and Selection, 2002. **15**(11): p. 871-879.

135. Shlens, J., *A tutorial on principal component analysis.* arXiv preprint arXiv:1404.1100, 2014.

136. Abdi, H. and L.J. Williams, *Principal component analysis.* Wiley interdisciplinary reviews: computational statistics, 2010. **2**(4): p. 433-459.

137. Nikoh, N., et al., *Phylogenetic relationship of the kingdoms Animalia, Plantae, and Fungi, inferred from 23 different protein species.* Molecular biology and evolution, 1994. **11**(5): p. 762-768.

138. Shai, Y., *Mode of action of membrane active antimicrobial peptides.* Peptide Science, 2002. **66**(4): p. 236-248.

139. Ouzounis, C., et al. *Are binding residues conserved?* in *Pacific symposium on biocomputing. Pacific symposium on biocomputing*. 1998.

140. Lazakidou, A.A., *Biocomputation and Biomedical Informatics: Case Studies and Applications: Case Studies and Applications*. 2009: IGI Global.

141. Sousa, S.F., et al., *Protein-ligand docking in the new millennium–a retrospective of 10 years in the field.* Current medicinal chemistry, 2013. **20**(18): p. 2296-2314.

142. Ballester, P.J., A. Schreyer, and T.L. Blundell, *Does a more precise chemical description of protein–ligand complexes lead to more accurate prediction of binding affinity?* Journal of chemical information and modeling, 2014. **54**(3): p. 944-955.

143. Sousa, S., et al., *Protein-ligand docking in the new millennium–a retrospective of 10 years in the field.* Current medicinal chemistry, 2013. **20**(18): p. 2296-2314.

144. Oleg, T. and J. Arthur, *Software news and update AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, eficient optimization, and multithreading.* Wiley InterScience, New York. doi, 2009. **10**: p. 1002.

145. Di Biasi, L., et al. *Novel algorithm for efficient distribution of molecular docking calculations*. in *Italian Workshop on Artificial Life and Evolutionary Computation*. 2015. Springer.

146. Piotto, S., et al., *Yada: a novel tool for molecular docking calculations.* Journal of computer-aided molecular design, 2016. **30**(9): p. 753-759.

147. Wang, R., et al., *The PDBbind database: methodologies and updates.* Journal of medicinal chemistry, 2005. **48**(12): p. 4111-4119.

148. Hooft, R.W., et al., *The PDBFINDER database: a summary of PDB, DSSP and HSSP information with added value.* Bioinformatics, 1996. **12**(6): p. 525-529.

149. Fan, H., et al., *Statistical potential for modeling and ranking of protein–ligand interactions.* Journal of chemical information and modeling, 2011. **51**(12): p. 3078-3092.

150. Release, S., *1: Maestro*, 2013, Schrodinger.

151. Epand, R.F., P.B. Savage, and R.M. Epand, *Bacterial lipid composition and the antimicrobial efficacy of cationic steroid compounds (Ceragenins).* Biochimica et Biophysica Acta (BBA)-Biomembranes, 2007. **1768**(10): p. 2500-2509.

152. Fiske, C.H. and Y. Subbarow, *The colorimetric determination of phosphorus.* J. biol. Chem, 1925. **66**(2): p. 375-400.

153. Lopez, D., *Molecular composition of functional microdomains in bacterial membranes.* Chemistry and physics of lipids, 2015. **192**: p. 3-11.

154. Koynova, R. and B. Tenchov, *Transitions between lamellar and non-lamellar phases in membrane lipids and their physiological roles.* OA Biochemistry, 2013. **1**(1): p. 1-9.

155. Gorbenko, G.P., J.G. Molotkovsky, and P.K. Kinnunen, *Cytochrome c interaction with cardiolipin/phosphatidylcholine model membranes: effect of cardiolipin protonation.* Biophysical journal, 2006. **90**(11): p. 4093-4103.

156. Cullis, P.t. and B. De Kruijff, *Lipid polymorphism and the functional roles of lipids in biological membranes.* Biochimica et Biophysica Acta (BBA)-Reviews on Biomembranes, 1979. **559**(4): p. 399-420.

Publications

- Parisi, R., Moccia, I., Sessa, L., Di Biasi, L., Concilio, S., Piotto, S. (2016) **Models for the Prediction of Antimicrobial Peptides Activity.** Communications in Computer and Information Science 587, pp. 83-91.

- Di Biasi, L., Fino, R., Parisi, R., Sessa, L., Cattaneo, G., De Santis, A., Iannelli, P. & Piotto, S. (2016). **Novel algorithm for efficient distribution of molecular docking calculations.** Communications in Computer and Information Science 587, pp. 65-74.

- Piotto, S., Di Biasi, L., Fino, R., Parisi, R., Sessa, L., Concilio, S., (2016). **Yada: a novel tool for molecular docking calculations.** *Journal of computer-aided molecular design*, *30*(9), 753-759.