UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Discriminative Estimation using Probabilistic Context Free Grammars for Protein Characterization

DEGREE FINAL WORK

Degree in Computer Engineering

Author: Marc Eres Olivares

Tutor: José Miguel Benedí Ruiz
Joan Andreu Sánchez Peiró

Course 2019 - 2020

Resum

La bioinformàtica és un camp de recerca actiu, l'objectiu principal de la qual és el desenvolupament de sistemes intel·ligents per a l'anàlisi en biologia molecular. Al llarg de l'última década s'ha produït un increment significatiu en l'ús de la teoría del llenguatge formal en aquest camp, donant lloc a diversos mètodes per a l'anàlisi i caracterització de molècules d'ADN, ARN i proteïnes. Encara així, en el camp de la proteòmica, la grandària de l'alfabet i la complexitat de les relacions entre aminoàcids han limitat l'aplicació de mètodes d'inferència gramatical a la producció de gramàtiques que no tenen un poder expressiu major que una gramática estocástica regular. No obstant això, aquestes gramàtiques regulars són incapaces de cobrir i detectar les dependències que apareixen en les estructures secundàries i terciàries de les proteïnes. És per aquest motiu que proposem un mètode d'estimació discriminatiu usant gramàtiques incontextuals per a l'anàlisi i detecció de llocs d'unió en proteïnes capaç de produir descripcions per a les seqüències d'interès.

Paraules clau: bioinformàtica; inferència gramatical ; llenguatge formal; gramàtica lliure de context probabilística; estimació descriminativa; aprenentatge automàtic; proteïnes; lectines lleguminoses; interaccions bioquímiques

Resumen

La bioinformática es un campo de investigación activo cuyo objetivo principal es el desarrollo de sistemas inteligentes para el análisis en biología molecular. A lo largo de la última decada, se ha producido un incremento significativo en el uso de la teoría del lenguaje formal en este campo, dando lugar a diversos métodos para el análisis y caracterización de moleculas de ADN, ARN y proteínas. Aun así, en el campo de la proteómica, el tamaño del alfabeto y la complejidad de las relaciones entre amino ácidos han limitado la aplicación de métodos de inferecia gramatical a la producción de gramáticas que no tienen un poder expresivo mayor que una gramática estocástica regular. Sin embargo, estas gramáticas regulares son incapaces de cubrir y detectar las dependencias que aparecen en las estructuras secundarias y terciarias de las proteínas. Es por este motivo que proponemos un método de estimación discriminativo usando gramáticas incontextuales para el análisis y detección de lugares de unión en proteínas capaz de producir descripciones para las secuencias de interés.

Palabras clave: bioinformática; inferencia gramatical; lenguaje formal; gramática libre de contexto probabilística; estimación descriminitiva; aprendizaje automático; proteínas; lectinas leguminosas; interacciones bioquímicas

Abstract

Bioinformatics is an active research area in which the objective is to develop intelligent systems for the analysis of molecular biology. Throughout the last decade, there has been a significant increase in the use of the formal language theory in the field of bioinformatics. Many methods based on formal language theory, statistical theory and learning theory have been developed for the analysis and characterization of sequences such as DNA, RNA and proteins. However, in the field of proteomics, the main problems resides in the size of the alphabet and the high complexity of the relations between amino acids. This parameters have deeply influenced the application of grammatical inference methods to the production of grammars in which the expressive power is not higher than stochastic regular grammars. Nevertheless, these stochastic regular grammars are unable to cover and detect any high-order dependencies such as nested and crossing relationships that are common in secondary and tertiary protein structures. For this reason, we propose a discriminative estimation model for the analysis and detection of protein binding sites that is capable of producing human readable descriptors for this sequences of interest.

Key words: bioinformatics; gramatical inference; formal language; stochastic context free grammars; discriminative estimation; machine learning; proteins; legume lectins; biochemical interactions

Contents

List of Figures

List of Tables

CHAPTER 1
Introduction

In this study, we are going to characterize a set of known proteins by using methods of discriminative estimation. In this manner, we will elucidate the main physicochemical characteristics of said proteins, that allow for a classification by using Stochastic Context Free Grammars (SCFG), through the structural information presented by a labelled set of proteins. For this reason, this study is divided into two intertwined parts; the biological context and the formal language context. In the former, we will investigate the central properties of proteins and how we can use these characteristics to acquire information from a proteic family. In the latter, we will inquire in the main definitions and techniques that allow us to predict if a given protein pertains to the established proteic family.

1.1 Motivation

In recent years, the demand for a fast and accurate classification of proteins has increased, as novel biochemical techniques for the analysis and synthesis of biomolecules have aroused, thus producing a surge in the number of proteins without classification. Also, the modeling and 3D representation of said proteins has an interest in the field of biotechnology and biochemistry, as been able to represent these structures further elucidates the mechanisms by which biomolecules are able to interact with each other. Meanwhile, in the field of formal languages, the capacity of regular grammars and context free grammars have not been able to represent and detect high-complexity relations in a biological context, such as nested and crossing relations between biomolecules.

Having in consideration these interests and the problematic of establishing relationships within the structure of biomolecules, we propose a method of identifying characteristics within a protein family, in order to establish a new method for the classification of proteins through the use of SCFG and discriminative estimation.

1.2 Objectives

- Understanding of Chomsky's hierarchy and the expressive power of each type of grammar.
- Review of the biochemical interactions that govern the protein world.
- Comprehension of the main mechanisms and theories that underlie the method of discriminative estimation and SCFG.
- Establishing the main characteristics that identify a proteomic family.
- Construction of a corpus that represents crucial structural information for the classification of proteins.
- Experimentation regarding the process of SCFG estimation.
- Evaluation of the results based on the chosen metrics.
- Determination of a further line of investigation.

1.3 Memory structure

This study is divided into six main chapters:

- Chapter 1 gives a brief summary of the intend and motivation behind this line of work.
- Chapter 2 introduces an overview of a selected number of articles, that have a direct relation with the field of bioinformatics by using different types of grammars or other formal language techniques.
- Chapter 3 establishes the main hierarchy in the proteomic world, alongside the physicochemical interactions that procure the stabilization of proteins.
- Chapter 4 inquires into the basis of Context Free Grammars (CFG) and Stochastic Context Free Grammars (SCFG). Also, this chapter explains the fundamental knowledge required to understand discriminative estimation.
- Chapter 5 presents the metrics and results of the investigation.
- Chapter 6 arrives to the conclusions of this study and establishes further lines of investigation, that can be drawn from the obtained results.

CHAPTER 2
Related Work

In this chapter we are going to take a look at the beginning of the field of bioinformatics. Mainly, we are going to dive into the initial techniques and goals that this field has used and achieved, but also to the current methodology and the state of the art.

Bioinformatics is a field that combines biological information with information manipulation techniques and analysis. The information is generated by "high-throughput data-generating experiments, including genomic sequence determinations and measurements of gene and protein expression patterns"[9]. The term was coined in the 1970s by Ben Hesper and Pauline Hogeweg, which felt that "information processing could serve as a useful metaphor for understanding living systems" [10]. In this early stages the work in this field was centered in understanding how living organisms were able to gather,process and use the information of their environment, and then utilize this information for the own advantage and evolution. One of the early success accomplished was the formation of the genetic code, "the central dogma of the unidirectional flow of information"[10], that culminated with the sequencing of the complete human genome (Figure 2.1).

Once the basis were set in stone (genetic code) a number of technologies were applied to understand the main evolutionary pathways of different living organisms. One of the early technologies or string metrics used in the area of genome sequencing was the Levenshtein distance. This algorithm falls into the category of edit distance, "which allows us to delete, insert and substitue simple characters in string-to-string comparison" [11].

Thanks to this algorithm we can apply different costs to the mentioned operations and detect possible differences between two given strings. "If we assign an operation cost of one to all the operations we are referring to a simple edit distance" [11]. In a biomolecular context, this algorithm is of vital importance as DNA, RNA and protein sequences can be viewed as long strings with a specific alphabet, the genetic code. Been able to search specific substrings in this long strings has been a giant step towards conquering fundamental problems such as "assembling the DNA chain from the pieces obtained by different experiments, looking for given features in DNA and protein chains, or determining how different two genetic sequences are" [11]. In a proteomic context this algorithm has been used to determine possible mutations or changes between proteins in order

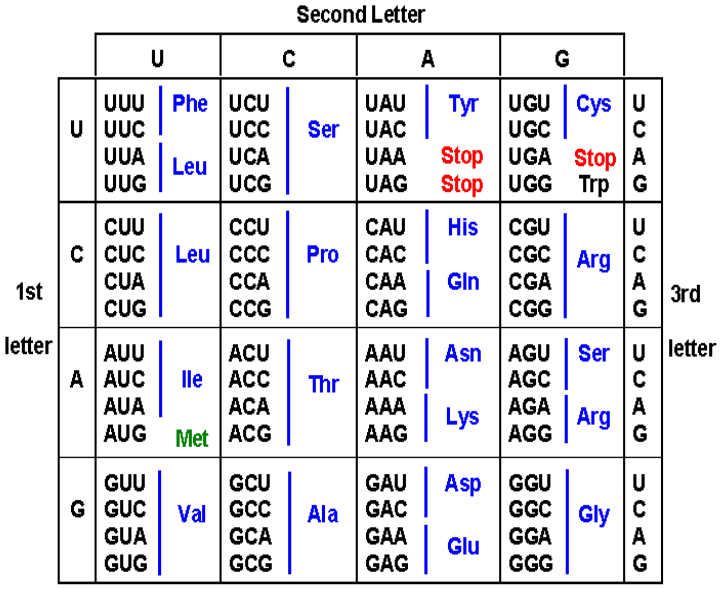


Figure 2.1: Genetic code. Referred to as the central dogma of genetic information, this codification allows the scientists to transform RNA sequences into amino acids that will later combine to form proteins. In other words it is the translation of RNA molecules to proteins. [1]

to determine the evolotunioray pathway of different species, in a field known as protein homology. Homology in DNA, RNA or proteins comes from the understanding that sequence similarity is a strong evidence that the two sequences or strings are related by evolutionary changes. With the use of the Levenhstein distance, we can align the sequences and detect the possible mutations and trace back each organism to their ancestral sequence (Figure 2.2).

Almost at the same time as the sequencing and alignment of DNA, RNA and proteins was been accomplished another investigation was been carried out by Stuart Kaufman using a different type of approach. This investigation involved the use of random Boolean networks, with the goal of understanding the main transcription regulation network of genes in living organisms. In this investigation organisms were understood as "randomly constructed molecular automaton and were examined by modeling the genes as binary devices" [12]. The results of this investigation reflected that genes could be affected by other genes and that the genetic network, which in large part tends to be stable, could undergo behavioural cycles under the stimulus of different noise altering methods. Nevertheless, the main achievement of this investigation was the capability of applying Markov chains into a genetic net to explain metabolic and epigenetic behaviour.

A Markov process is defined as a process whose "main property is that the probability of any particular behaviour of the process, when the present state is known exactly, is not altered by additional knowledge concerning its past behaviour" [3]. It should be noted that if the knowledge of the present state is not complete or unclear, then the probability of any possible predicted future will be influenced by the additional information relating to the past behaviour of the system. In mathematical terms a Markov process can be expressed as [3]:

$$Pr(a < X_t \leq b | X_{t_1} = x_1, X_{t_2} = x_2, ..., X_{t_n} = x_n) = Pr(a < X_t \leq b | X_{t_n} = x_n)$$

i	j	0	1	2	3	4	5	6	7	8
		ϵ	C	A	T	G	A	C	T	G
0	ϵ	0	0	0	0	0	0	0	0	0
1	T	1	1	1	0	1	1	1	0	1
2	A	2	2	1	1	1	1	2	1	1
3	C	3	2	2	2	2	2	1	2	2
4	T	4	3	3	2	3	3	2	1	2
5	G	5	4	4	3	2	3	3	2	1

Figure 2.2: Levenshtein matrix by using a cost operation of one. In this example we can see how the two DNA strings ($A = CATGACTG$ and $B = TACTG$) are compared. As we can see this process corresponds to a dynamic programming computation where the value of $D[i,j]$ is calculated by its previous cells $D[i,j-1]$, $D[i-1,j]$ and $D[i-1,j-1]$. Particularly, if $i = 0$ then $D[i,j] = 0$, if $j = 0$ then $D[i,j] = i$, if $A[j-1] = B[i-1]$ then $D[i,j] = D[i-1,j-1]$ otherwise $D[i,j] = 1 + min(D[i-1,j], D[i,j-1], D[i-1,j-1])$. [2]

In this context, a Markov chain (X_n) is a Markov stochastic process in which "the possible states are countable or a finite set" [12]. Regarding the value of X_n it can also be defined as the "outcome of the nth trial" [3]. The probability of $X_n + 1$, being in a state y where we already know that X_n is in the state z (Figure 2.3), is mathematically defined as [3]:

$$Pr_{y,z}^{n,n+1} = Pr(X_{n+1} = y | X_n = z)$$

Following this notation we can infer that the transition probabilities are functions of initial and final state, but also of the time of transition as well. "When one-step transition probabilities are independent of the time variable, We say that the Markov process has stationary transition probabilities" [3]. In general, most Markov chains can be defined as stationary transition probabilities.

The use of Markov chains in a biomolecular context has been extensive in the last years. For example, Chao and Kou applied Markov chains in biophysical experiments based on enzymatic systems. In this investigation the researchers were able to use continuous time Markov chains in order to elucidate the "correlation between experimental fluorescence intensity and enzymatic reaction times, focusing on the role of substrate concentration with enzymatic reactions" [13]. The results demonstrated that the use of Markov chains were able to capture the change of conformation of the enzymes in a time period of nanoseconds, making it a promising technology for deeper understanding of biomolecular processes.

Following the footsteps of Chao and Kou, we find the investigation that was carried by Gupta and Rawlings. The main goal in this research was to apply time continious Markov chains and Markov chain Monter Carlo techniques to "char-

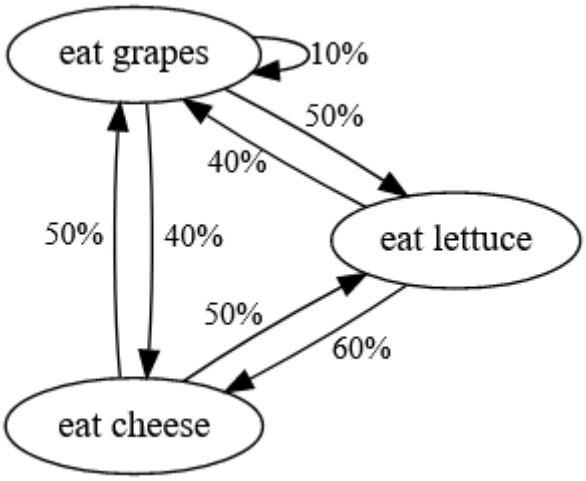


Figure 2.3: Markov chain. In this example we can see a Markov process represented by a Markov chain that corresponds to the dietary habits of mice. The states of this Markov chain are eat cheese, eat grapes and eat lettuce. The mice can only eat once a day, so if today the mice ate cheese tomorrow the probability of eating lettuce or grapes will be the same. [3]

acterize the kinetic behaviour of viral infection on baby hamster kidney cells" [14]. In order to do this, four phases corresponding to viral infection cycle were characterized; Activation, the phase in which the virus becomes active and ready to infect. Once the virus is active, it is able to penetrate the cell or host; Transcription, phase in which the virus blends with the DNA replication system in order to produce more viruses; Replication, exponential growth of the virus within the cell; and Translation, the final phase in which the virus is able to assemble into the active formation and the cell undergoes the process of lysis, releasing the newly produced viruses into the environment. The main conclusions that were withdrawn from this research were that Markov chains "required numerical integration in n_r dimensions, which is computationally expensive and that the uniformization technique required for the Markov chain Monte Carlo technique required enormous computation " [14]. This means that Markov chains have a limit in a biomolecular context due to the inherent complexity of the systems and new methods should be explored.

Finally, the research developed by Pratas *et at*, involving Markov chains has had a notable success. In this research Markov chains were utilized to sequence the chromosomes of chimpanzee and orangutans, which is an alignment technique that does not involve the edit distance algorithm. The use of Markov chains allowed the researchers to detect "large-scale and small-scale genomic rearrangements, including balanced translocations and inversions" [15]. This biological phenomenons can not be detected by fundamental laboratory techniques such as microscopic visualization, thus the use of Markov chains signified an extension to the tools used for genomic structure characterization.

Moving forward, the current tendency in bioinformatics has been to try to apply Chomsky's hierarchy in order to sequence and correctly predict protein folding. Chomsky's hierarchy, mainly context-free grammars, will be further ex-

plained in chapter four but now we are going to take a look at the first level of Chomsky's hierarchy, regular languanges (Figure 2.4).

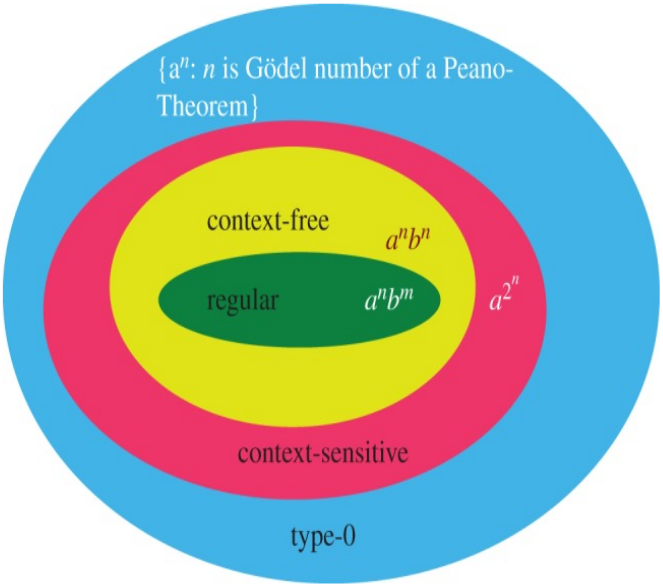


Figure 2.4: Chomsky's hierarchy. As we can see in the figure, the hierarchy is composed of four levels; The first level corresponds to regular grammars; the second level corresponds to context-free grammars; the third level corresponds to context-sensitive grammars; and the final corresponds to type-0 grammars . [4]

The genetic code can be viewed as a language, as DNA ,RNA and proteins are represented as strings of characters with a meaning. In this context we can express that the genetic code can also be defined as a regular language defined by a regular grammar. In this types of grammars, all rules take one of two forms:

$$\begin{aligned} A &\rightarrow a \\ A &\rightarrow aB \end{aligned}$$

where A and B are non-terminal symbols and a is a terminal symbol. In this definition "the non-terminals can be understood as category symbols and the arrow as "consists of" " [4]. Other interpretations regard that regular grammars are a representation of an automaton, where the non-terminals are the states of the automaton and the arrow is the transition to next the state, similar to the structure in Markov chains. In this type of automaton the start symbol usually is represented with the non-Terminal S and the rules without a non-terminal can be considered as a final state. Finally, as there is a finite number of non-terminals a regular grammar can be viewed as a finite state automaton. The power of this types of grammars resides in the fact that "it is possible to construct an algorithm (finite state automaton) that reads a string from left to right, and then outputs yes if the string belongs to a language or no otherwise" [4]. This means that each regular language corresponds to some finite state automaton, which is a algorithm that consumes one symbol and changes its state according to the symbol that has been used, if the last state visited is a final state the string will be accepted, otherwise the string can not be represented by this finite state automaton.

One of the most cited investigations regarding the application of automaton for the sequencing of proteins, is the investigation performed by Coste and Kerbellec. In this investigation the researchers used learning automaton for the "merging of ordered partial local multiple alignments" [5]. Local multiple alignments consist of conserved regions of amino acids that appear in all proteins that are from the same family or set. The purpose of this research was to find the best alignment of proteins pertaining to the same family, trying to center the alignment were the conserved regions appeared. The main motivation for this investigation was the fact that classical tools tend to have a bias towards the alignment of said proteins, due to the fact that all proteins tend to be used for the alignment. In order to eliminate this bias, the researches introduced a new term known as partial local multiple alignment which does not require the involvement of all the proteins of the set or family. By merging partial local multiple alignments, the researchers were able to "build automatons representing complex succession of local consensus" [5] (Figure 2.5).

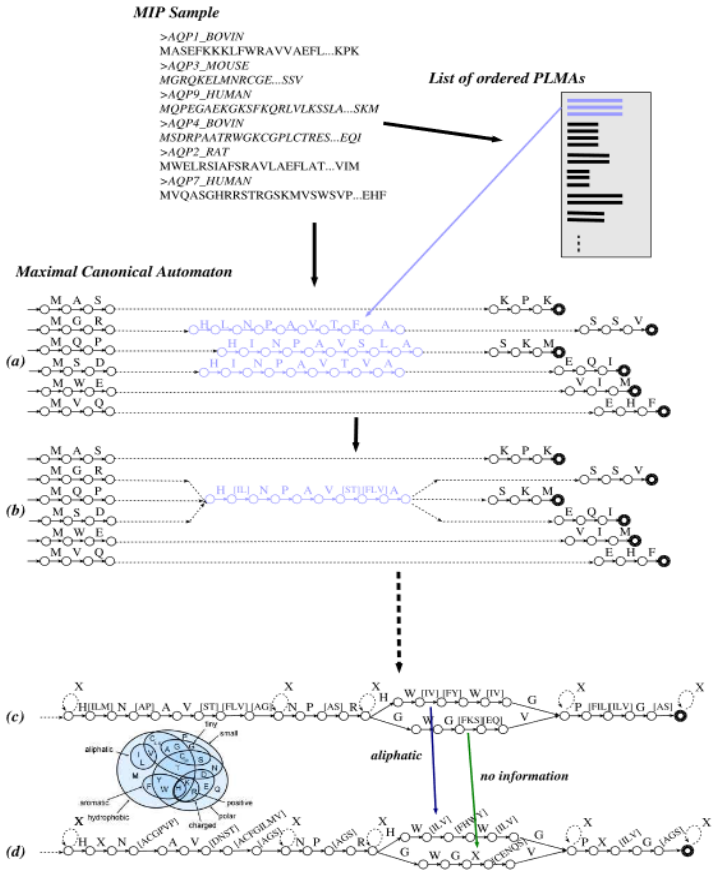


Figure 2.5: Partial local multiple alignment procedure. The first step in this process is to acquire a set of proteins that contain a conserved region $(H-(IL)-N-P-A-V)$. Next, automatons for each string or protein are modeled. a) The next step is to align the automaton, centered around the conserved region. b) Now that the conserved region is aligned, the automaton are merged in that exact position. c) Result of the merging of all the partial local multiple alignments with representative information and non-representative information. d) Final automaton returned that identifies the physico-chemical information of this set of proteins. [5]

Although the application of regular grammars or automatons have had a considerable success, they are limited in the identification of interactions within proteins (see Chapter 3). For this reason Dyrka has proposed the use of context-free grammars in order to determine possible interactions and other dependencies in the secondary structure of proteins. In this investigation, a framework was developed in order to establish descriptors that " allow the detection of protein regions that are involved in binding sites of proteins, but also provide insight in their structure" [6]. To do this, a series of grammars were developed by using genetic algorithms with the combination of the main properties of the common amino acids, and the application of a number of constraints. The results of this investigation demonstrated that the descriptors were able to "highlight meaningful biological characteristics and achieve a high accuracy in the annotation and detection of this characteristics" [6] (Figure 2.6). For this reason, this investigation was used as a basis for the development of this research.

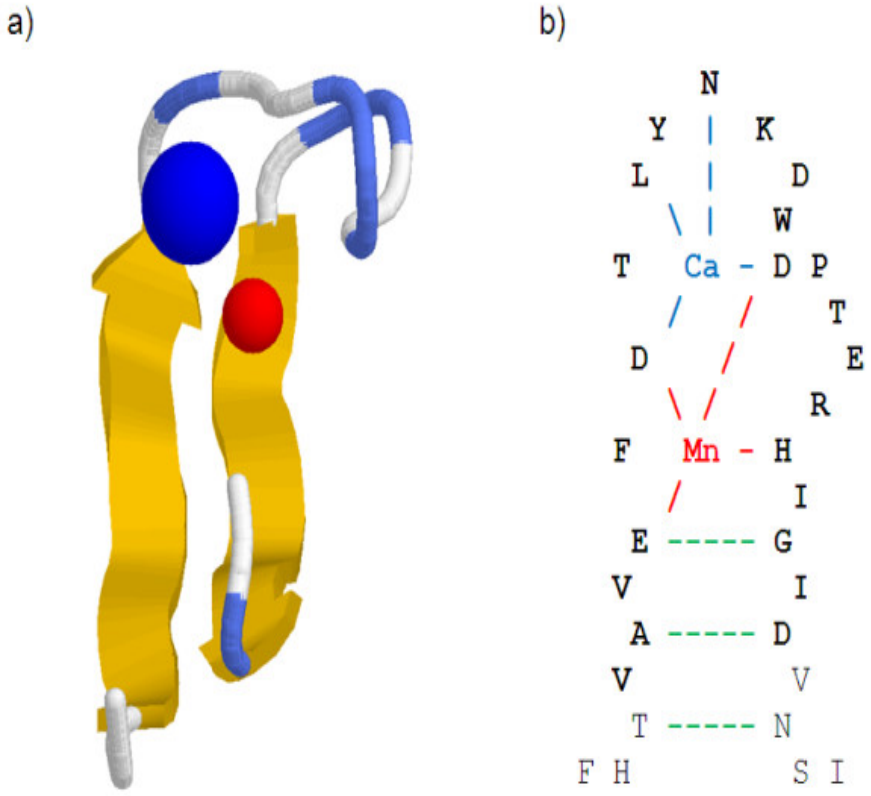


Figure 2.6: Example of applying context-free grammar for the modeling and characterization of proteins. a) 3D model of a protein belonging to the leguminous lectin family, the modeling was performed by the Protein Data Bank (PDB). b) Descriptor of the modeled protein, as we can see not only the interactions with the metal ions (calcium in blue and manganese in red) is represented, also the hydrogen bonding (dotted lines in green) between the beta chains (in yellow) is characterized. [6]

CHAPTER 3
Analysis of Protein Sequences

3.1 Biomolecular Context

In order to fully comprehend any research first a context has to be set. In this section we are going to introduce the reader to the biomolecular context upon which the investigation is founded. To do this, first we will inquire in what are amino acids and how do we classify them. Next, we will explain the main characteristics and nature of proteins. Finally, we will describe the main intra and inter actions from which different structures arise in the proteomic world.

3.1.1. Amino acids

Amino acids are the main building blocks of proteins. This means that each amino acid is joined to the subsequent amino acid, by a special type of bond (known as the peptide bond), to form the protein of interest. For this reason, "proteins can be deconstructed to their constituent amino acids (hydrolysis) by a variety of methods" [7] . Twenty different amino acids have a higher frequency of occurrence in proteins found in living organisms, known as common amino acids. These common amino acids have been assigned a three-letter code and one-letter characters, "which are used as a shortcut to indicate the composition and sequence of amino acids in a given protein" [7].

All twenty of the common amino acids have a carboxyl group (corresponds to the acidic part) and an amino group (corresponds to the amino part) bonded to the same atom, the alpha carbon (Figure 3.1). The main difference between these amino acids resides in the side chain, also known as the R group. "The R group is responsible for the main characteristics of each amino acid, such as size, electric charge and solubility of the amino acid in water."[7]

The understanding of the chemical properties of the common amino acids is a key factor for the comprehension of the biochemistry, which governs the proteomic interactions and therefore their folding capabilities. Upon this knowledge we can classify the different amino acids into five main classes based on the chemical nature of the R groups. Mainly we consider "their polarity or tendency to interact with water at a neutral pH (7.0) . The polarity of the R groups have a great

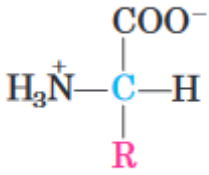


Figure 3.1: Structure of an amino acid. As we can see in the figure the carbon in blue corresponds to the alpha carbon.This carbon is bonded to an amino group (NH_3) and a carboxyl group (COO). We can also find the R group which varies from amino acid to amino acid. [7]

variability, from nonpolar and hydrophobic (water-insoluble) to highly polar and hydrophilic (water soluble)" [7]. (Figure 3.2).

Nonpolar amino acids. The main characteristic of this group of amino acids is a tendency to be non polar, meaning that they do not interact with water and tend to be allocated in the interior of proteins, and therefore are considered as hydrophobic. For this reason,"The R groups of alanine, valine, leucine and isoleucine tend to cluster together within proteins, stabilizing protein structure by hydrophobic interactions" [7] . Of all the amino acids, glycine has the simplest structure, although it is classified as a nonpolar amino acid, "its unusual small R group makes no real contribution to hydrophobic interactions" [7]. Methionine, one of the two sulfur containing amino acid tends to have a major contribution to hydrophobic interactions due to "the presence of a thioether group $(CH_2 - S - CH_3)$" [7]. Proline, "has a rigid conformation that reduces the structural flexibility of polypeptide regions and contributes to hydrophobic interactions" [7], due to the presence of said structural ring $(CH_2 - CH_2 - CH_3 - NH_3)$. Finally, both phenylalanine and tryptophan, with their aromatic side chains are "relatively nonpolar and therefore classified as hydrophobic". [7]

Polar amino acids. The main characteristic of these amino acids is their ability to be soluble in water, which is possible thanks to the functional groups present in their R groups. These functional groups are able to form hydrogen bonds,a special type of weak bond that we will later discuss, with water. In this class of amino acids, "we can find serine, cysteine, asparagine and glutamine" [7]. The polarity of serine and threonine is caused by the "presence of hydroxyl groups (OH)" [7]; the polarity of cysteine is caused by the "presence of a sulfhydryl group (SH)"[7]; and that of asparagine and glutamine by their amide groups (NH_2). Finally, we can also include in this group tyrosine due to the "presence of hydroxyl groups". [7]

Electrically charged amino acids. The amino acids that have the greatest capability of interaction with water are those in which their R group is either positively charged or negatively charged. The amino acids in which the R group is positively charged at neutral pH, are lysine which has an "amino group at the end of its R chain" [7]; arginine, which has a "positively charged guanidino group $(NH_2 - C - NH_2)$" [7]; and histidine which has an "imidazole group $(C - NH - CH - N - CH)$" [7]. On the other side we have two amino acids having R groups

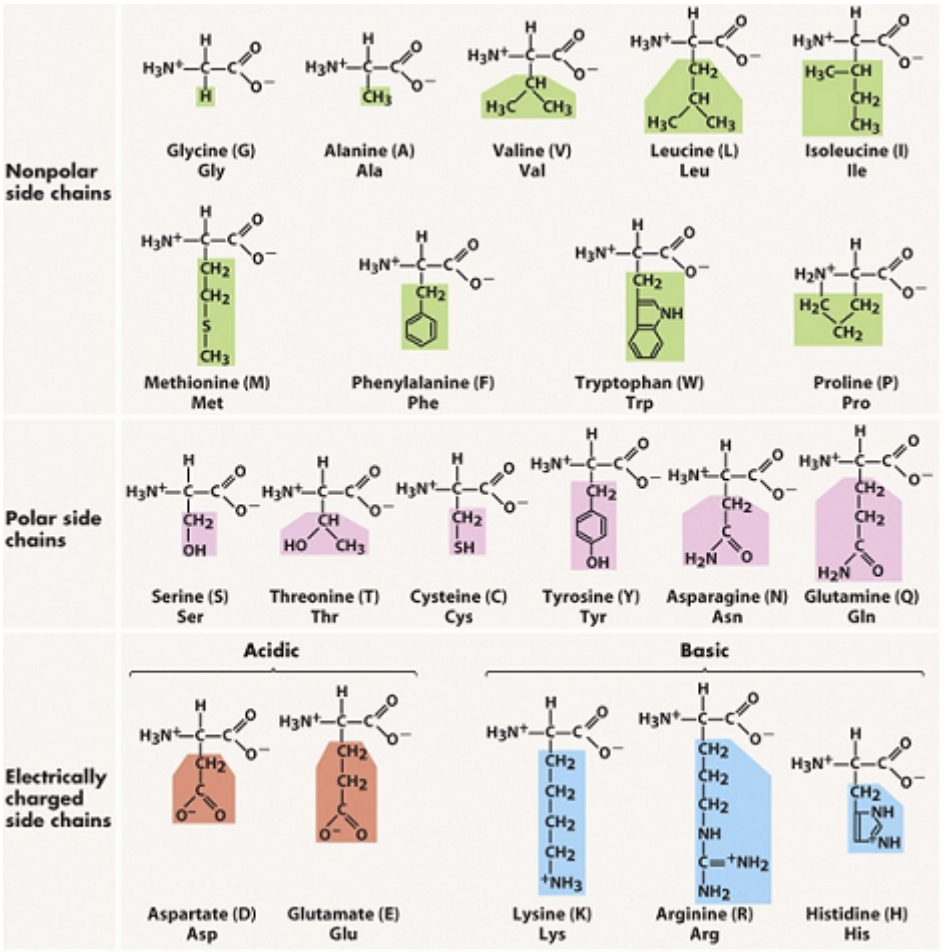


Figure 3.2: The structural formula of the common amino acids. The uncolored portions are those present in all the amino acids and the colored portions correspond to the R groups. [7]

with a net negative charge at neutral pH, aspartate and glutamate, each of which has "a second carboxyl group" [7].

Now that we have explained the main classification of amino acids, we can begin to understand that depending on the amino acids present in a protein, there will be a direct effect on the behavior of said protein in different environments. These characteristics will therefore have an impact not only in the structure of the protein but also in the main function that the protein will carry in a biological organism.

3.1.2. Proteins

Now are focus is going to shift towards the polymers of amino acids, peptides and proteins, and how do they bond with each other. "Biologically occurring proteins range in size from small to very large, consisting of three to thousands of linked amino acids chains" [7]. For this reason, first we are going to inquire on the chemical reactions that give rise to these polymers.

Linking amino acids to form peptides. Two amino acid molecules can be bonded through a specific linkage, known as peptide bond, to form a dipeptide.

"Such linkage is formed from the removal of the elements of water (dehydration) from the alpha carboxyl group of one amino acid and the alpha amino group of another amino acid" [7]. The peptide bond is a prime example of a condensation reaction, an usual reaction in all living organisms (Figure 3.3). Three amino acids can be bonded to give arise to a two-peptide bond and this will form a tripeptide, we can continue this process with any number of amino acids to obtain more complex peptides. Particularly,"When many amino acids bond through the peptide bond we obtain a polypeptide." [7]. The terms polypeptide and protein tend to be used at the same time, but is important to note that polypeptides are prone to have a molecular weight below 10,000 M (S.I. kg/kmol) while proteins have higher molecular weights, although in literature the smaller polypeptides are also referred as proteins.

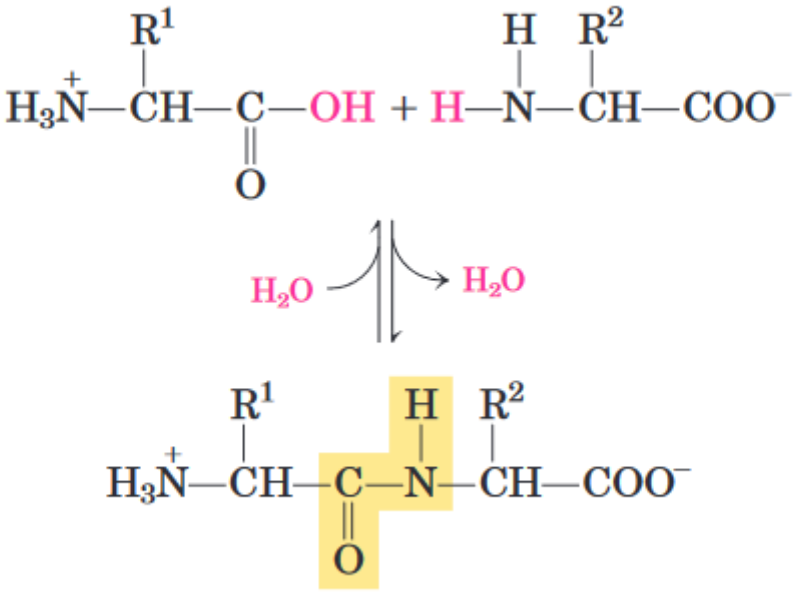


Figure 3.3: Condensation of the peptide bond. The amino group of one amino acid reacts with the carboxyl group of the other amino acid forming a peptide bond, colored in yellow. This process is reversible, when the peptide bond is formed a molecule of water is removed (dehydration), on the other hand when we want to break a peptide bond we will have to add a molecule of water to it (hydrolysis). [7]

Peptides can be characterized by their net charge. Proteins contain only one unbonded amino group and one unbonded carboxyl group, at the begin and end of their chain. These groups are charged depending on the environment, in an acidic environment the carboxyl group acquires a hydrogen and the net charge is positive, in a basic environment the amino group loses a hydrogen promoting a net negative charge. This process is known as "ionization and some R groups of the amino acids within the proteins can also succumb to ionization" [7]. Therefore, amino acids tend to contribute to the overall acid-base properties of the molecule. Thus, "the acid-base behavior of a peptide can be predicted form its unbonded amino and unbonded carboxyl groups, as well as the nature and number of its ionizable R groups" [7]. (Figure 3.4)

Proteins and polypeptides found in living organism occur in a vast range of sizes. No assumptions can be made about the molecular weights and size of

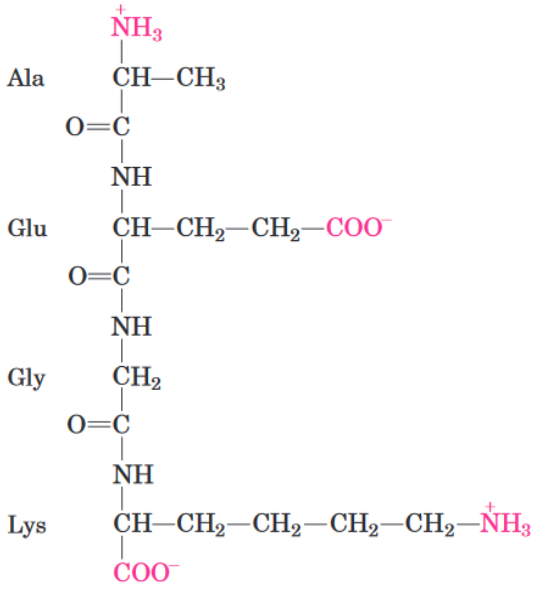


Figure 3.4: Alanylglutamylglycyllysine. As we can see, this tetrapeptide has one unbonded amino group and one unbonded carboxyl group at opposite ends of the chain. We can also find two ionizable R groups, corresponding to glutamate (Glu) and lysine (Lys). [7]

biologically active proteins in relation to the function that they acquire in an organism or their environment. Therefore, we have to consider all types and sizes of proteins,as even the smallest of proteins can have critical roles in biological contexts. One common example is oxytocin, "a nine amino acid in length peptide which is secreted by the posterior pituitary and stimulates uterine contractions" [7]. On the other hand, slightly larger peptides also have critical functions on organisms such as insulin, which contains two polypeptide chains, "one containing 30 amino acids and the other 21" [7].

Following the example of insulin, we can find proteins that have two or more polypeptide chains, this are refered as multisubunit proteins. The polypeptide chains in this type of proteins are able to interact with each other through weaker chemical interactions, which tend to increase the overall stability and give rise to the structure of the protein. The individual polypeptide chains in a multisubunit protein can be identical or different. "If at least two are identical the protein is known as oligomeric, and the identical units are referred to as protomers" [7].

In order to determine the approximate number of amino acids in a protein, "we can divide its molecular weight by 110" [7]. To obtain this number, we have to consider that "the average molecular weight of the 20 common amino acids is about 138 M" [7], although the smaller amino acids tend to appear more frequently in most proteins. If we take into consideration this tendency "the average molecular weight decreases to 128 M" [7]. Finally, as we noted before a molecule of water has to be removed to create a peptide bond (dehydration) (Figure 3.3) which has a molecular of weight of 18 M, this can be explained by the molecular weight of hydrogen which is 1 M while the molecular of oxygen is 16 M, as water has two hydrogen atoms and one oxygen atom we obtain a total molecular

weight of 18 M. Thus, if we subtract to the average molecular weight of amino acids (128 M) the molecular weight of water(18 M) we end with 110 M.

Proteins can be characterized through amino acid composition. The decomposition of polypeptides or proteins through the process of hydrolysis results in characteristic proportions of the different amino acids. "The common amino acids almost never appear in equal amounts in a protein" [7]. This means that some proteins may contain a large number of an amino acid in its sequence, while other amino acids appear with lower frequency or do not even appear.

The decomposition of proteins through hydrolysis alone may be not be enough for the analysis and determination of amino acid composition. This is mainly caused by the fact that some unwanted reactions can occur during the hydrolysis procedure. For example, the R groups of asparagine and glutamine can react to form aspartate and glutamate, giving rise to ambiguities in the protein sequencing. "When a precise amino acid composition is required, biochemists tend to use additional procedures to try to resolve ambiguities such as ELISA, Electrophoresis or High-Performance Liquid Chromatography (HPLC)" [7].

3.1.3. Structure and interactions

To comprehend and determine the structure of large molecules such as proteins, we have to define several levels of complexity. To do this, we have to arrange the different structures in the proteomic world in a conceptual hierarchy. For this reason, "Four levels of protein structure are defined" [7]. The first level recognized as the primary structure, corresponds to the joined amino acids through the peptide bond. Within this hierarchical level, "the most important element is the composition of amino acids" [7], that conform the protein. The next hierarchical level is the secondary structure, which refers to "the stable disposition of amino acids giving form to well-known structural conformations" [7]. This is possible thanks to weak interactions that arise between the amino acids of said secondary structure. Next, we find the tertiary structure "which describes all aspects of the three-dimensional folding of a protein" [7]. Finally, "when a protein has two or more polypeptide subunits, the arrangement in space is known as the quaternary structure" [7] (Figure 3.5).

Primary structure of proteins produce weak interactions of critical importance. "The spatial arrangement of atoms in a protein is called its conformation" [7]. All possible conformations of a given protein comprehend any structure that can be adopted without breaking the linkage between amino acids. A structure can change its conformation by rotation of the peptides bonds under certain biological conditions, such as a change in the pH of the environment or interactions with other proteins. "In the context of protein structure, the term stability can be defined as the tendency to maintain a native conformation" [7] . Native proteins are only partially stable, meaning that a protein theoretically can assume countless different conformations, and as a result "the unfolded state of the protein is characterized by a high degree of conformational entropy" [7], this means that in order for a native protein to unfold a certain amount of energy will be required, as the native formation tends to be more stable. This entropy and the hydrogen bonds that arise between the amino acids R groups are responsible

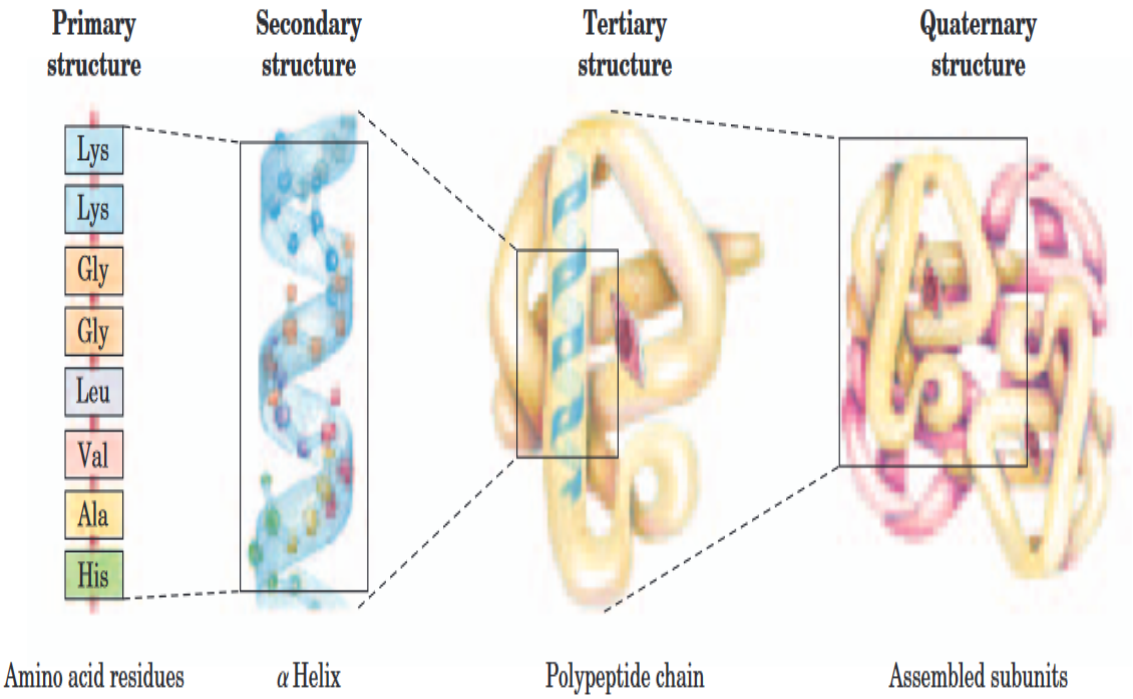


Figure 3.5: The four levels of hierarchy in proteins. The primary structure is composed by a sequence of amino acids joined together. The resulting polypeptide arranges itself into a secondary structure through amino acid interaction. This secondary structure is one of the many structures that appear in the tertiary structure of a protein. Finally, this tertiary structure can be one of the subunits that give rise to the quaternary structure, which turns out to be the protein hemoglobin. [7]

of the stability that prevents the unfolding of the protein (Figure 3.6). Going back to the common amino acids, the polar amino acids have the greatest capacity to form this type of bonds.

It is of critical importance to comprehend the role of these "weak interactions for us to understand how polypeptides chains fold into secondary and tertiary structures, and how they interact with other protein subunits, to form quaternary structures" [7]. We have to take into account that individual bonds (peptide bond) that contribute to the native conformations of proteins are much stronger, meaning that they require more energy for them to be broken, than weak interactions. Nevertheless, "the number of weak bonds or interactions are significantly higher and predominate as a stabilizing force in protein structure" [7]. The protein structure that gives rise to the conformation with the lowest entropy "is the one with maximum number of weak interactions" [7] and tends to be the native conformation for this reason.

Besides hydrogen bonding, hydrophobic interactions also play an important role in protein conformation; "the interior of a protein is generally a core of hydrophobic amino acids" [7]. Therefore, proteins will tend to protect the nonpolar amino acids in the interior while exposing the polar amino acids to interact with the environment. The combination of these two factors,hydrogen bonds and hydrophobic interactions, are key to the stabilization of protein folding.

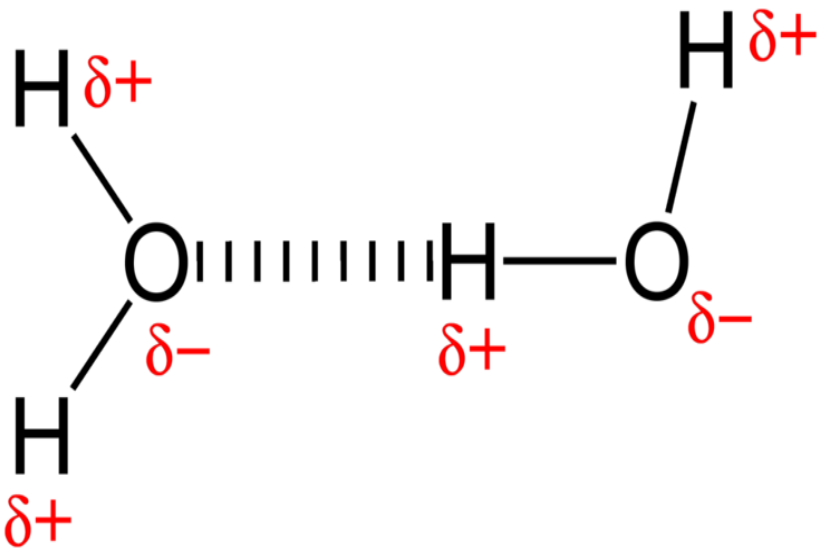


Figure 3.6: Hydrogen bond between two molecules of water. The hydrogen bond is one of the most important weak bonds in nature, as it stabilizes proteins and other macromolecules. As we can see in this example, two molecules of water form a hydrogen bond (represented by the lines parallel to each other) due to the fact that oxygen tends to be more electronegative than hydrogen, meaning that its affinity toward electrons is higher and thus electrons in the bond tend to be closer to oxygen. This creates what is known as a dipole moment (colored red) that polarizes the net charge of the molecule. Finally, as the oxygen now presents a negative dipole and the hydrogen presents a positive dipole, they are able to form a weaker bond known as hydrogen bond. [7]

Secondary structure of proteins tend to be alpha helix or beta strands. "The term secondary structure refers to the local conformation of some segments of the protein" [7]. There are a many secondary structures that can appear in nature and also be produced artificially, but we will focus only on the secondary structures that appear in nature and are the most stable. Of this proteins the most frequent and stable are the alpha helix and the beta strand conformations

The alpha helix conformation depends on the amino acid composition. "The simplest arrangement a polypeptide chain can assume is a helical structure known as the alpha helix" [7]. In this helical conformation the polypeptide chain travels around an imaginary axis drawn longitudinally and the R groups of the amino acids are expelled outward so that they end up exposed to the environment. "The repeating unit is a single turn of the helix, which extends around 5.4 angstroms (1 angstrom = 0.1 nanometer) along the long axis" [7] (Figure 3.7). This means that each turn that appears in this helical conformation includes an average of 3.6 amino acids. The stability of this turn is again a product of the hydrogen bonds that arise through the interaction of the nitrogen atoms present in the alpha amino group.

"The twist of an alpha helix ensures that critical interactions occur between amino acids" [7]. Not only the twist with the consequent formation of hydrogen bonds through the alpha amino group are responsible of the stability of the alpha helix, also the positioning or appearance of positively charged amino acids three amino acids away from negatively charge amino acids need to be taken

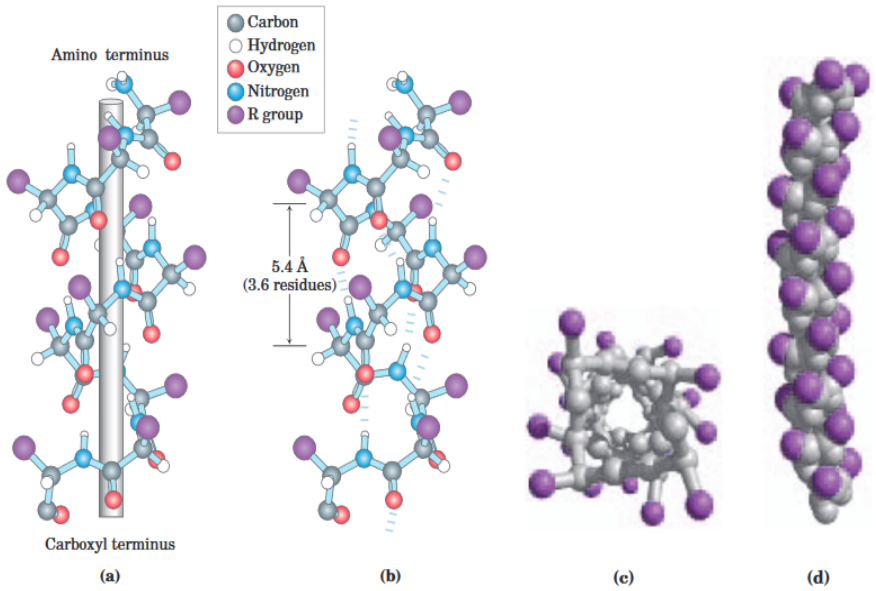


Figure 3.7: Models of alpha helices, representing different aspects of its structure. a) Model of alpha helix that travels around a longitudinal axis. b) Representation of the hydrogen bonds produced by the nitrogen in the alpha amino group(parallel lines to each other) that help stabilize this secondary structure. Also we can see, that each turn in the helical conformation comprehends an average of 3.6 amino acids. c) The alpha helix as viewed from the amino terminus. As we can see the R groups, colored purple, are protruded towards the outer side of the secondary structure. d) In this last model we can see how the atoms in the center of the alpha helix tend to form a cluster with few spaces in between. [7]

into consideration. This positioning results in the formation of opposite charged pairs, which in turn adds a new force that increases the overall stability of this secondary structure. Thus, the identity of the amino acid present near the ends of the turns in the alpha helix will favor and stabilize this type of structure. Another factor that needs to be taken into account is the size of the R groups. As we mentioned before around 3.6 amino acids are involved in a turn, but not all amino acids can be allocated in this turns as the amino acids with a large R group, such as tryptophan, will interact with the other R groups of the subsequent amino acids, thus acting as a destabilizing force. Mainly, three different type of constraints affect the conformation of an alpha helix, "first the electrostatic repulsion or attraction between successive amino acids R groups, second the size of adjacent R groups and third the interactions between R groups spaced three amino acids apart" [7].

The beta strand structure favors the interaction of peptide chains that result in structural sheets. This secondary structure is usually a larger conformation compared to the alpha helix, as a higher number of polypeptide chains are involved. Also the polypeptide chains tend to have a higher molecular weight and therefore have more amino acids in length (as we already noted, to obtain the average length of amino acids we have to divide the total molecular weight by 110). The main characteristic of the beta strand, is that "the backbone is arranged into a zig zag rather than a helical structure" [7]. This zig zag conformation aligns itself side by side to another zig zag conformation, to form a structure similar

to a series of folds known as a beta sheet. This arrangement favors the formation of hydrogen bonds between the alpha amino and alpha carboxyl groups of the adjacent amino acids which significantly increases the overall stability of this structure.

Another difference between the alpha helix and the beta sheet, is that the individual segments that form a beta sheet can appear nearby or far away in the linear sequence of amino acids, for this reason more polypeptides are involved in the formation of this structure. On the other hand, a similarity that both secondary structures share is the fact that the R groups of adjacent amino acids tend to protude from the polypeptide backbone, therefore increasing the interaction with the environment. Nevertheless, due to the zig zag nature of the beta strand the R groups appear in opposite directions, creating an alternating pattern.

"The adjacent polypeptide chains in a beta sheet can be either parallel or antiparallel, having the same or opposite amino-to-carbonyl orientations respectively" [7]. Both conformations of the beta sheet have more or less the same nature, although the main difference between them is that the distance between two amino acids that appear in the zig zag tends to be shorter in the parallel conformation, "6.5 angstroms for the parallel sheets and 7 angstroms for the antiparallel" [7]. This difference in the distance between amino acids affects directly to the hydrogen bonding pattern. In the antiparallel conformation the alpha amino group and the alpha carboxyl group face each other while in the parallel conformation both are faced in a 45 angle.(Figure 3.8).

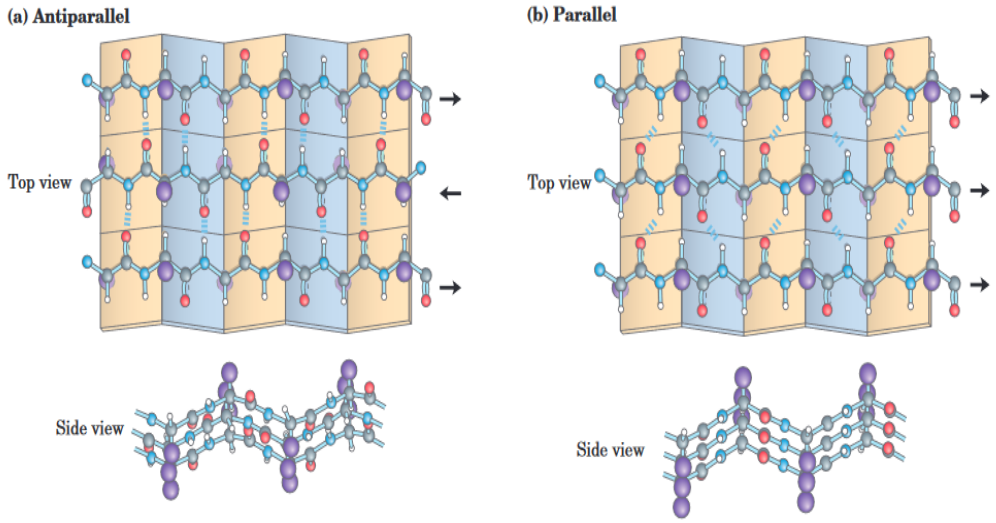


Figure 3.8: The beta sheet conformation. As we can see the the R groups protrude from the beta sheet and emphasize the zig zag formation. We can also see that between the beta strands a hydrogen bond is produced in order to stabilize this type of conformation. a) Antiparallel beta sheet, the black symbolize the direction from the last unbonded amino group to the last unbonded carboxyl group, we can also note that the hydrogen bond formed between adjacent amino acids is a straight line. b) Parallel beta sheet with the same amino-terminal to carboxyl-terminal orientation. If we take a look at the hydrogen orientation we can see that the it has a 45 angle tilt. [7]

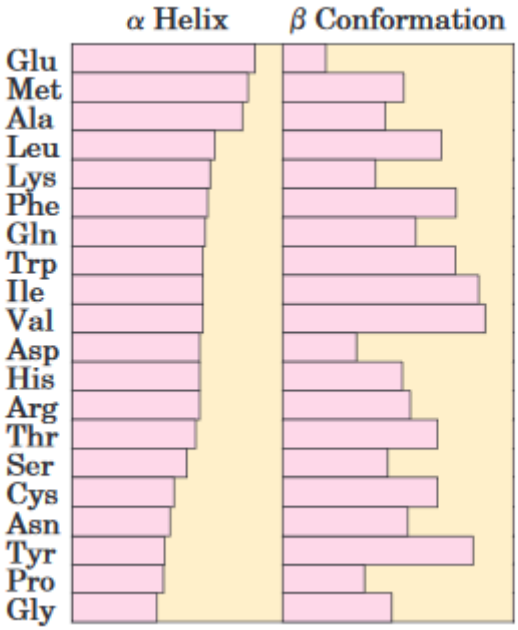


Figure 3.9: Relative probabilities that a given amino acid will occur in the two common types of secondary structure. [7]

Protein tertiary and quaternary structures. "The overall three-dimensional arrangement of all atoms in a protein is referred to as the protein's tertiary structure" [7]. In the secondary structure we were exploring how the relatively close positioning of the amino acids affects to the spatial configuration that they adopt based on their identity, whereas in the tertiary structure the interactions between the secondary structures is the topic at hand. This means that the range of interactions in the tertiary tends to be higher than in the secondary structure.

"Amino acids that are far apart in the polypeptide sequence and that reside in different types of secondary structure may interact within the completely folded structure of a protein through hydrogen bond or hydrophobic interactions" [7].This means that the amino acids in the different secondary structures will interact with each other and form weak interactions that will create the basis of the stability of the tertiary structures. It is important to note, that as we increase in the level of hierarchy the stability of the conformations tends to decreases as it largely depends on the formation of weak interactions. For this reason, a subtle change in the pH could result in the destabilization of this tertiary structures, a process known as protein denaturation.

As we already explained some proteins are formed by a series of polypeptide chains, or subunits. "The arrangement of these protein subunits in the three-dimensional complexes constitutes the quaternary structure" [7]. We can classify the proteins that have a quaternary structure into two classes: fibrous proteins, which tend to be displayed in sheets and composed in its majority by beta sheets, and globular proteins, which tend to appear in spherical shapes. One of the two main differences between the two classes is their structural behaviour. On one hand fibrous proteins are usually composed of a single type of secondary structure, the beta sheet. On the other hand, globular proteins are composed by a mixture of secondary structures. The second difference between the two classes

is the role that the play in living organisms. Fibrous proteins tend to "provide support, shape and external protection to organisms" [7]. Globular proteins tend to be involved in processes of hormonal regulation and metabolism, as most of the enzymes that partake in such activities are globular proteins.

Now that we have inquired in the biomolecular context, we can continue with the problem at hand, protein characterization and the understanding of the different patterns found in protein families.

3.2 Protein Characterization of Legume Lectins.

Now that we have explained the fundamental biochemical principles, we are going to elucidate the main characteristics and patterns found among the protein family that we are going to work with; Legume lectins. For this reason, in this section we will go into more depth of the physicochemical properties that make this set of proteins an interesting case of study.

Lectins are a class of proteins abundant in nature. Mainly these proteins can be found in animals, insects, plants and microorganisms. The main role of these proteins is the processing of sugars or carbohydrates in the main routes of metabolism and for this reason they have attracted interest over the last several decades. Also, "lectins are are excellent models for examining the molecular basis of specific reactions that occur between proteins and other types of molecules" [16].

Of all the lectins that have been analyzed and sequenced the legume lectins are the largest and best characterized family. Pertaining to this family, researchers have found highly conserved amino acid structures within taxonomically distant species of plants, making this set of proteins a homologous family. This allows for the demonstration that "these proteins have been conserved throughout evolution and a strong point can be made that they must have an important function (or functions) in nature" [16].

The family of legume lectins, including those from taxonomically distant species, share molecular characteristics in common (Figure 3.10) on which we are going to inquire in the following paragraphs. Among this characteristics we are going to rely on three fundamental properties that allow for the characterization of said proteins; beta sheet composition, metal ion presence and hydrophobic conserved areas.

Legume lectins are largely beta strand proteins, hence their quaternary and tertiary structure heavily rely on the formation of beta sheets. [17] The main structure of these proteins is generally a single or two polypeptide chain that share similar amino acid sequences with some variations in the length of the strands. This structure is characterized by what is known as the jelly roll motif, "present in many proteins and often related with carbohydrate-binding activity" [17]. This jelly roll is formed usually by two or three sets of antiparallel beta sheets, which are connected by several loops and single chain amino acids of varying lengths (Figure 3.11). As already noted in the previous section, beta strands are formed by only a select group of amino acids, due to their biochemichal properties (Figure 3.9). For this reason, in the beta strands present

Subunit	
Mol wt[a]	25–30 kDa
Number[b]	2 or 4
Combining sites	
Number[c]	1 per subunit
Specificity[d]	Usually identical
Metal ions	Ca^{2+}, Mn^{2+}
Amino acids	
Hydroxy- and carboxy	High
Sulfur containing	Low or absent
Sequence homology	Extensive
N-Glycosylation[e]	Common
3-Dimensional Structure	
α-helix	Low or absent
β-sheet	Main structural element

Figure 3.10: Main characteristics of legume lectins. The main properties used for this study were amino acid composition and beta-sheet formation, which is strictly related to the hydrophobic properties of the proteins. On the other hand, to classify proteins in this family researchers use the presence of metal ions that are directly related to the binding and interacting capacities of proteins and also the sequence homology that studies the evolutionary properties within a set of proteins.

in the legume lectins we can find highly conserved patterns of amino acids (homologous) that are expressed throughout the whole family. In this study, we will focus our efforts in this highly conserved patterns that give arise to the beta strands and therefore the beta sheets in order to detect and classify a set of proteins pertaining to the legume lectin family. The most prolific conserved pattern within this family corresponds to a beta strand that follows the pattern $[LIV] - [STAG] - V - [DEQV] - [FLI] - D - [ST]$ (i.e. the sequence V-A-V-E-F-D-T corresponds to this pattern) and for this reason it is used as a consensus for the classification of proteins in this family. Nevertheless, other patterns exist that are less conserved but still viable for classification purposes and therefore used in this study .

Legume lectins require the presence of metal ions in their structure in order to engage in metabolic reactions. [17] In the analysis and sequencing of this proteins, two metal ions are commonly found: Mn^{2+} and Ca^{2+}. This ions are responsible for the binding of carbohydrates which allow for the proper chemical reactions to ensure. Although this chemical property was not used for our study, its importance resides in the previous classification done by the researchers to build a corpus of proteins that contained this two metal ions and were part of the legume lectin family (Prosite: PS00307, available at https://prosite.expasy.org/). This corpus was the main source of proteins for this investigation.

Pairing of beta strands to form beta sheets allow for the appearance of conserved hydrophobic areas in legume lectins. As already stated in the previous section, the pairing of beta strands give rise to beta sheets and depending on the overall composition of amino acids in this structure we can find different physicochemical properties. In the case of the legume lectins, the amino acids that are

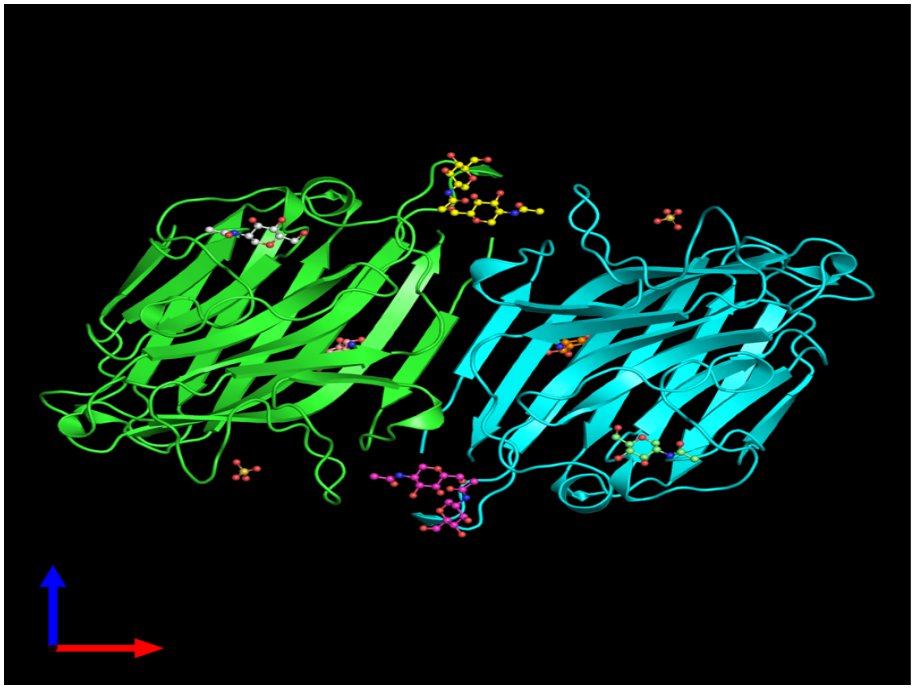


Figure 3.11: General structure of legume lectins. As we can see, there are two polypeptide chains present in this specific protein (green and blue) that are composed in their majority by beta strand structures (arrows) that interact with each other to form a beta sheet. These antiparallel beta strands are connected to each other through the presence of loops, which allow for the turning, and single chain amino acids that contribute to the overall length of the tertiary structure.

conserved throughout the family tend to be hydrophobic amino acids and therefore when paired to form beta sheets the resulting structure maintains this hydrophobicity. As this amino acids are well conserved, we can find repeated areas of hydrophobic interactions throughout the legume lectin family. This property allows us to detect and note the patterns that are replicated for the classification of proteins belonging to this family as we will see in the section 4.3.

CHAPTER 4
Protein Characterization based on Context Free Grammars

In this chapter we are going to set the formal language context that has been used to carry out the experimentation. First we will inquire in what are grammars and languages, followed by a specific type of these types of grammars, the context free grammars. Next, we will introduce a type of context of free grammars known as probabilistic context free grammars and finally we will explain the application of these types of grammars through their estimation.

4.1 Formal Language Context

4.1.1. Context Free Grammars

In this section we are going to introduce some definitions and concepts related to languages and grammars in the context of formal language theory. This notions will be required to relate both the genetic code and probabilistic context free grammars for the characterization of a given family of proteins.

In order to define a grammar first we have to understand what is an **alphabet**. An alphabet, $\sum$, is a finite set of symbols. These symbols are the "basic units or primitives that form the language and when they are group together they become strings or chains" [8], in a proteomic context these symbols would correspond to the amino acids and the string to the protein. The length of this string x is the number of symbols that it has and can be written as $|x|$. The empty string is the chain of symbols that has no elements and can be denoted as ϵ. We can now write the set of all the string which is higher or equal to zero and can be formed with the symbols of Σ as Σ^*. Likewise Σ^+ will denote the set of all strings with a length higher or equal to 1 that can be formed with element of Σ, thus $\Sigma^+ = \Sigma^* - \epsilon$.

Now we can define a **language**, L, on Σ as a subset of the set Σ^*. Thus a language can be understood as a "formal automaton that has a string accepting character or with a formal grammar that has a string generating character" [8]. For this particular investigation we are going to use formal grammars as a mechanism of specification of the formal language.

A **formal grammar** is defined as a tuple (N, Σ, S, P). In this tuple we find four elements; N is a finite set of symbols known as non terminals; Σ is a finite set of symbols known as terminals that meet the constraint $N \cap \Sigma = \emptyset$; P is a finite set of rules or productions, where each rule is a pair (α, β) and is represented as $\alpha \rightarrow \beta$ where $\alpha, \beta \in (N \cup \Sigma)^*$, in this context α is known as antecedent and β as consequent; finally $S \in N$ is the initial symbol or axiom of the grammar. As an example if we have a rule r = $(\alpha \rightarrow \beta)$ that pertains to P and $\mu, \omega \in (N \cup \Sigma)^*$, then there is a direct derivation that is expressed as $\mu\alpha\omega \Rightarrow^r \mu\beta\omega$ [8]. A derivation therefore exists when we can transform α_1 to α_2, where $\alpha_1, \alpha_2 \in (N \cup \Sigma)^*$, using a set of rules and steps that have the form $\alpha_1 = \mu_0, \mu_1, ..., \mu_m = \alpha_2$ with $\mu_0, \mu_1, ..., \mu_{m-1} \in (N \cup \Sigma)^*$ following the set of rules $(r_1, r_2, ..., r_m) \in P$ thus [8]:

$$\alpha_1 = \mu_0 \overset{r_1}{\Rightarrow} \mu_1 \overset{r_2}{\Rightarrow} ... \overset{r_m}{\Rightarrow} \mu_m = \alpha_2$$

The **generated language defined by a grammar** (G) can be written as $L(G) = \{x \in \Sigma^* | S \Rightarrow^* x\}$. Depending on the nature of the rules we can classify the grammars into four main classes [8] :

- The first class of grammars corresponds to regular grammars. The rules of regular grammars follow the form $A \rightarrow aB$ where $A, B \in N$ and $a \in \Sigma$.
- The second class of grammars are referred to as context free grammars where every rule follows the form $A \rightarrow \alpha$ where $A \in N$ and $\alpha \in (N \cup \Sigma)^*$.
- The third class of grammar correspond to context sensitive grammars where every rule follows the constraint $\alpha \rightarrow \beta$ such that $|\alpha| \leq |\beta|$.
- The fourth class of grammars are unrestricted grammars which do not use any type of constraints.

The hierarchy between these grammars is naturally extended to the formal languages. By means of this we say that a formal language is regular if it is generated by a regular grammar, it is context free if it is generated by a context free grammar and that is sensitive to context if it is generated by a sensitive grammar or unrestricted grammar.

The complexity of the problems that can be tackled by each class increases in accordance with the hierarchy. In this regard, regular grammars are used to solve simple problems, while unrestricted grammars are used for more complex problems. Parallel to this principle the algorithms that allow the manipulation of the grammars also grow with its expressive capacity. For this reason, "some problems that can be resolved by using regular grammars can not be approached by unrestricted grammars" [8] .

Context free grammars and languages suppose a reasonable compromise between the complexity of the problems that can be approached and the cost of the algorithms that allow for an adequate manipulation. In one hand, context free grammars have the sufficient expressive capacity to establish long term relations between the primitives of the language, making them a convenient tool

for the representation of complex problems. On the other hand, robust and efficient algorithms exist that allow for a suitable manipulation [8]. For this reason we decided to work with context free grammars (CFG) for the characterization of family proteins.

Now that we have established the framework of CFG, we are going to explain some of the properties related with these grammars. Given a context free language, this can be represented by more than one CFG. In this regard two CFG, G_1 and G_2, are equivalent if $L(G_1) = L(G_2)$. We also say that a CFG is in Chomsky Normal Form (CNF) if all the rules follow the form $A \rightarrow BC$ or $A \rightarrow a$, where $A, B, C \in N$ and $a \in \Sigma$. Other normal forms of CFG exist, although throughout this work we are going to work with CFG in CNF. This framework does not imply a loss in generality, as given a CFG G_1, there is a CFG G_2 in CNF such that $L(G_1) = L(G_2)$. This means that any context free grammar can be defined as a CFG in CNF [8].

A **leftmost derivation** of a string $x \in L(G)$, d_x, is a derivation such that $\mu_0 = S$, $\mu_m = x$ and r_i, $1 \leq i \leq m$, rewrites the non terminal which is leftmost to μ_{i-1}. In this manner the leftmost derivation is defined by the sequence of rules that has been used. Analogously we can also define the rightmost derivation, in which the element that is rewritten is the non terminal rightmost to μ_{i-1}.

A related concept to the derivation is the derivation or **analysis tree.** An ordered and labeled tree is a derivation tree if a CFG G [8]:

- Each node of the tree has a label, that is a symbol of $(N \cup \Sigma)$.
- The root of the tree has the label S.
- If a node that has an A label has a direct descendant that is different from itself, then $A \in N$.
- If the nodes $n_1, n_2, ..., n_m$ are direct descendants of the node n (which label is A) and the order is from left to right with labels $A_1, A_2, ..., A_m$ respectively, then $A \rightarrow A_1 A_2 ... A_m$ is a rule from P.

An analysis tree can be associated with one only derivation, by this manner the sequence of rules used is the one obtained doing a path in preorder of the tree and using the last characteristic of the previous definition (Figure 4.1). Given a string x and a CFG G such that $x \in L(G)$, it is possible that more than one tree of analysis that allows for the derivation of x starting from the initial symbol exists. In this regard the CFG defined in Figure 4.1 and the string 1010 can be generated with the derivation used in the figure or with the derivation $(S \rightarrow SS, S \rightarrow AS, A \rightarrow 1, S \rightarrow 0, S \rightarrow AS, A \rightarrow 1, S \rightarrow 0)$.

A CFG is said to be non ambiguous if for each $x \in L(G)$ exists a unique derivation that allows the generation of x; otherwise we say that the CFG is ambiguous. Given a string $x \in L(G)$ we will denote D_x as the set of all the different derivations that has the string x and $|D_x|$ the representation of its length [8].

Now that we have introduced the main properties and definitions of CFG, we will consider an essential issue: how can we identify that a string pertains to the language generated by a grammar. This problems consists on the evaluation of

the relation $x \in L(G)$, given a string x and a CFG G. The syntactic analysis of a string consists in the determination of this relation. The solution to this problem consists in the search of a sequence of derivations that allows the derivation of x from the initial symbol of G using the rules of the grammar [8].

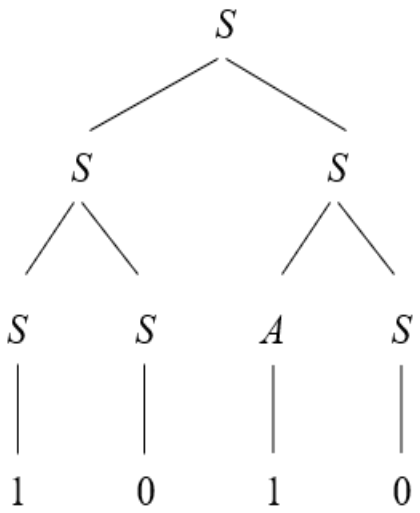


Figure 4.1: Derivation tree. Given the CFG $G = (\{S, A\}, \{0, 1\}, S, \{S \to SS, S \to AS, S \to 0, S \to 1, A \to 1\})$ and the derivation tree t, the associate derivation is $d = (S \to SS, S \to SS, S \to 1, S \to 0, S \to AS, A \to 1, S \to 0)$. [8]

The problem can be solved with a linear cost with the length of the string, but to do so we have to restrict severely the type of grammars and thus the expressive capacity of these grammars. Another solution is with a cubic cost with the length of the string using grammars without restrictions. These grammars, "allow the adequate representation of phenomenons of noise and variability, that are common in the problems we approach" [8].

An efficient solution to tackle the syntactic analysis consists in the use of tabular methods based on dynamic programming. These methods are are based on the construction of an analysis table, in which each cell represents the solution of a specific subproblem. The most known tabular method is the Cocke-Younger-Kasami algorithm, "that operates with a CFG in CNF, and the Earley algorithm, that allows the use of a CFG without any particular form" [8]. In their essence, both algorithms are similar, but for this work we are going to use the Cocke-Younger-Kasami algorithm.

The **Cocke-Younger-Kasami algorithm** is based on the construction of an analysis table V whose dimension is $|x| \times |x|$, such that if $A \in V_{i,j}$, then $A \Rightarrow_* x_i...x_j$. In this regard, $x \in L(G)$ if S pertains to $V_{1,|x|}$ (Figure 4.2). The algorithm operates analyzing parts of the string of bigger length each time until the whole string is analyzed. This type of analysis is known as bottom up analysis as it considers analysis subtrees from the leafs towards the root. "The time cost pf the Cocke-Younger -Kasami algorithm is $O(|x|^3|P|$ while the spatial cost is $O(|x|^2|N|)$" [8].

Algorithm 1: Cocke-Younger-Kasami

Data: CFG $G = (N, \Sigma, S, P)$ in CNF and a string x of length $n > 0$
Result: Analysis table V
for $i = 1$ *until* n **do**
  $V_{i,i} = \{A | A \to x_i \in P\}$
end
for $i = 1$ *until* $n - 1$ **do**
  for $j = i + 1$ *until* n **do**
    $V_{i,j} = \emptyset$
    for $k = i$ *until* $j - 1$ **do**
      $V_{i,j} = V_{i,j} \cup \{A | A \to BC, B \in V_{i,k} \text{ and } C \in V_{k+1,j}\}$
    end
  end
end

Figure 4.2: Cocke-Younger-Kasami algorithm for the computation of the analysis table [8].

4.1.2. Probabilistic Context Free Grammars

In the different applications of syntactic form recognition (SFR) there is an usual presence of noise and variability, together with certain phenomenons of uncertainty that require a more sophisticated treatment. This considerations introduce the necessity of a generalization of the already explained models (CFG) to adjust to the phenomenons mentioned. In the Theory of Formal Languages, "we can approach this problem by associating a measure of probability to some of the concepts already explained" [8]. Probabilistic context free grammars (PCFG) are appropriate models that introduce this notion to the formal languages, they are simple tools that have already existing efficient algorithms that allow their adequate use; they permit a compact and straightforward representation of noise and variability; and they have the advantage that robust algorithm exist for their machine learning. This characteristics make PCFG a potent tool to tackle complex problems such as protein characterization.

For this reason we are now going to introduce a series of definitions that will allow for a more in depth understanding of what are PCFG. In this manner, as we already explained CFG, we will follow a similar order than the already presented, but extending them in a stochastic framework.

A **probabilistic language** of an alphabet Σ can be defined as a pair (L, Φ), where L is a formal language and $\Phi : \Sigma^* \to \mathbb{R}$ is a real function of the strings from Σ^*. The probability function Φ satisfies the following conditions [8]:

- $x \notin L \Rightarrow \Phi(x) = 0$ for all $x \in \Sigma^*$.
- $x \in L \Rightarrow 0 < \Phi(x) \leq 1$ for all $x \in \Sigma^*$.
- $\sum_{x \in L} \Phi(x) = 1$.

Previously we have seen how formal grammars are adequate tools for the definition of formal languages. In the same regard we can extend the mentioned concepts to introduce new methods that allow us to treat probabilistic grammars. In this work we are mainly focused on context free models, so the definitions that we are going to explain can be applied to the other classes of grammars.

A **probabilistic context free grammar** (PCFG), G_p is defined as a pair (G, p) such that G is a CFG, denoted as a characteristic grammar and p is a function $p : P \rightarrow]0, 1]$, that presents the property [8]:

$$\forall A \in N, \sum_{(A \rightarrow \alpha) \in \Gamma_A} p(A \rightarrow \alpha) = 1. \tag{4.1}$$

Where Γ_A represents the set of rules of the grammar whose antecedent is A.

We define the **derivation probability** d_x of the string x as: [8]:

$$Pr(x, d_x|G_p) = \prod_{(A \rightarrow \alpha) \in P_A} p(A \rightarrow \alpha)^{N(A \rightarrow \alpha, d_x)}. \tag{4.2}$$

Where $N(A \rightarrow \alpha, d_x)$ represents the number of times that the rule $A \rightarrow \alpha$ has appeared in the derivation d_x.

In this manner we define the **probability of a string** x as: [8]

$$Pr(x|G_p) = \sum_{d_x \in D_x} Pr(x, d_X|G_p). \tag{4.3}$$

Where D_x denotes the set of all the different derivations of the string x.

Given a PCFG $G_p = (G, p)$, we can define a PCFG $G_p^{'}$ whose characteristic grammar is in CNF and for all $x \in L(G)$ and $Pr(x|G_p) = Pr(x|G_p^{'})$ is fulfilled [8].

We will denote the **probability of the best derivation of the string** x as [8]:

$$\widehat{Pr}(x|G_p) = max_{d_x \in D_x} Pr(x, d_x|G_p). \tag{4.4}$$

And the **most probable derivation** or the best derivation as [8]:

$$\hat{d}_x = argmax_{d_x \in D_x} Pr(x, d_x|G_p). \tag{4.5}$$

In this regard we can also express the best best possible derivation as $Pr(x, \hat{d}_x|G_p)$. The definitions 4.3 and 4.4 can be extended to an arbitrary number of derivations following the next definition.

Given the string x and a set of derivations of this string $\Delta_x \subseteq D_x$, we define the **probability of the string** as: [8]

$$Pr(x, \Delta_x|G_p) = \sum_{d_x \in \Delta_x} Pr(x, d_x|G_p). \tag{4.6}$$

We can observe that this expression is equal to the expression 4.3 when the set of derivations is equal to the maximum. This expression is also equal to the equation 4.4 when the derivation is the one with maximum probability of all the possible derivations.

We define the **language generated by PCFG** G_p as, $L(G_p) = \{x \in L(G) | Pr(x|G_p) > 0\}$. [8]

Given a probabilistic language (L, Φ), where L is the context free grammar, it is reasonable to think that is possible to find a CFG $G_p = (G, p)$ such that $L = L(G)$ and Φ is computed in terms of the expression 4.3. Nevertheless, no all probabilistic languages in which L is a context free grammar can be represented by a PCFG as the following theorem establishes: [8]

Given a context free language $L = \{a^n b^n | n \geq 0\}$ *and the function* $\Phi(a^n b^n) = \frac{1}{en!}$ *and* $\Phi(x) = 0$ *if* $x \notin L$*, then there is no PCFG* G_p *that represents the probabilistic language* (L, Φ).

We can observe that (L, Φ) meets the conditions of the definition of a probabilistic language. We can demonstrate this theorem in a simple manner, if we examine the expression 4.3 we can observe that it grows inversely to a polynomial function depending on the length of the string. While the function Φ of the last theorem grows inversely to a exponential function depending on the length of the string. As no polynomial can approximate a function that grows exponentially, Φ can not be computed by any PCFG.

A PCFG is consistent if and only if:

$$\sum_{x \in L(G)} Pr(x|G_p) = 1 \tag{4.7}$$

In any other case the grammar is not consistent.

Given a consistent PCFG G_p, the pair $(L(G), \mathcal{P})$ is a probabilistic context free language, where $\mathcal{P}$ is a function of probability computed in terms of the expression 4.3. [8]

The problem of consistency in a PCFG is of vital importance in order to use this formalism as a method of representing context free probabilistic languages.

For this reason, we are going to inquire in a series of concepts that will allow us to study the problem of consistency in a PCFG. When a PCFG is consistent, we can acquire a series of important characteristics of the language that it generates. The first of these properties is: [8]

Given a PCFG G_p, we define the **expected values of non terminals matrix** $E = (e_{i,j}, 1 \leq i, j \leq |N|$ as:

$$e_{i,j} = \sum_{(A \to \alpha) \in \Gamma_{A_i}} p(A \to \alpha) N(A_j, \alpha). \tag{4.8}$$

Where the value of $N(A_j, \alpha)$ represents the number of times that the non terminal A_j appears in the consequent α. In this expression the value of $e_{i,j}$ is the expected number of non terminals that A_j can generate directly from A_i.

In this regard, we define the **expected values of terminals matrix** $Z = (z_{i,j}), 1 \leq u \leq |N|, i \leq j \leq |\Sigma|$ as: [8]

$$z_{i,j} = \sum_{(A \to \alpha) \in \Gamma_{A_i}} p(A \to \alpha) N(a_j, \alpha) \tag{4.9}$$

Where the value $N(a_j, \alpha)$ represents the number of times that the terminal a_j appears in the consequent α. As the previous definition, the value $z_{i,j}$ represents the expected value of terminals a_j that can be generated directly from A_i.

We can now establish a new theorem based on the previous definitions to determine which conditions have to be met for a PCFG to be consistent: [8]

A PCFG is consistent if $\rho(E) < 1$, where $\rho(E)$ or spectral radius is the absolute value of the greatest eigenvalue of the matrix E.

This theorem provides a simple and direct way to prove the consistency of a PCFG by only studying the characteristics of E. There are different methods to calculate the eigenvalues of a matrix. Nevertheless, we are only interested in the greatest absolute value of them all. The next theorem allows the resolution of this problem.

For any squared matrix M of dimension m, $\rho < 1$ if and only if there exists a $n \geq 1$ for every $i, 1 \leq i \leq m$: [8]

$$\sum_{j=1}^{m} |(M^n)_{ij}| < 1 \tag{4.10}$$

One method to determine if the spectral value of the matrix is less than one consists in the evaluation of the expression 4.10. If the result of the operation is not less than one, we multiply the matrix by itself and we apply the expression again. This process is repeated until we can check that the initial condition is true or the process is carried out a sufficient number of times [8].

The syntactic probabilistic analysis of PCFG consists in the determination of $P_r(x|G_p) > 0$. To solve this problem we have to find at least one derivation whose probability is greater than zero and allows the derivation of the string from the initial symbol of the grammar.

To solve this problem we can use three different algorithms. The first algorithm, known as Inside, calculates the probability of the string from all the possible derivations. The Inside algorithm is based on a dynamic programming scheme analogous to the Cocke-Younger-Kasami algorithm (Figure 4.2). This algorithm is based on the definition $e(A < i, j >= P_r(A \overset{*}{\Rightarrow} x_i ... x_j | G_P)$ as the probability that the substring $x_i ... x_j$ is generated from A. This probability can be evaluated efficiently, for all $A \in N$, as [8] :

$$e(A < i, i >) = p(A \to x_i) \quad 1 \leq i \leq |x|, \tag{4.11}$$

$$e(A < i, j >) = \sum_{B,C \in N} p(A \to BC) \sum_{k=i}^{j-1} e(B < i, k >) e(C < k+1, j >) \quad 1 \leq i \leq j \leq |x|. \tag{4.12}$$

In this manner, $P_r(x|G_p) = e(S < 1, |x|)$. The time cost of this algorithm is $O(|x|^3|P|)$ and the space cost is $O(|x|^2|N|)$.

Another solution to the syntactic analysis is the Outside algorithm. This algorithm is analogous to the Inside algorithm as it allows to determine if a string x can be generated by a PCFG, by calculating the probability of said string from all possible derivations. In the Outside algorithm we define $f(A(< i, j >= P_r(S \overset{*}{\Rightarrow} x_1...x_{i-1}Ax_{j+1}...x_{|x|}|G_p)$ as the probability of generating the substring $x_1...x_{i-1}$ from the initial axiom, then the generation of the non terminal A and then the generation of the substring $x_{j+1}...x_{|x|}$. In this regard, the non terminal A is in charge of generating the substring $x_i...x_j$. This expression can be calculated following the next scheme. For all $A \in N$ [8]:

$$f(A < 1, |x| >) = \begin{cases} 1 & \text{if } A = S \\ 0 & \text{if } A \neq S \end{cases} \tag{4.13}$$

And, for $1 \leq i \leq j \leq |x|$,

$$\begin{aligned} f(A < i, j >) \;=\; & \sum_{B,C \in N} \Bigg(p(B \to CA) \sum_{k=1}^{i-1} f(B < k, j >)\, e(C < k, i-1 >) \\ & + \; p(B \to AC) \sum_{k=j+1}^{|x|} f(B < i, k >)\, e(C < j+1, k >) \Bigg) \end{aligned} \tag{4.14}$$

In this manner $P_r(x|G_p) = \sum_{A \in N} f(A < i, i >) p(A \to x_i), 1 \leq i \leq |x|$. As the Outside algorithm has a similar behaviour to the Inside algorithm, the time cost and space cost is analogous and therefore $O(|x|^3|P|)$ and $O(|x|^2|N|)$ respectively.

The last solution is the Viterbi algorithm. With this solution we are able to determine if $Pr(x|G_p) > 0$ by finding at least one derivation whose probability is bigger than zero. With the Viterbi algorithm we are able to calculate the derivation of the string whose probability is the maximum. The basis of this calculation is based upon the definition $\widehat{e}(A < i, j >) = \widehat{Pr}(A \overset{*}{\Rightarrow} x_i...x_j|G_p)$ as the probability of the best derivation that generates the substring $x_i...x_j$ starting from A. For all $A \in N$ [8]:

$$\widehat{e}(A < i, i >) = p(A \to x_i), \tag{4.15}$$

$$\widehat{e}(A < i, j >) = \max_{B,C \in N} p(A \to BC) \max_{k=i,...,j-1} \widehat{e}(B < i, k >) \widehat{e}(C < k+1, j >) \tag{4.16}$$

$$1 \leq i < j \leq |x|. \tag{4.17}$$

Therefore, $\widehat{Pr}(x|G_p) = \widehat{e}(S < 1, |x| >)$.

Similar to the algorithms already presented the time cost is $O(|x|^3|P|$, while the space cost is $O(|x|^2|P|$.

The best derivation of the string $x, \widehat{d_x}$, can be obtained easily from the Viterbi algorithm if we store the arguments that maximize each of the subproblems. This information is of vital importance if we want to establish the most probable relations between the different parts of a given string.

4.2 Estimation of Probabilistic Context Free Grammars

4.2.1. Discriminative Estimation

Discriminative training is a method that is applied for the improvement of recognition accuracy in the field of Natural Language Processing (NLP) problems. This method of recognition used for the estimation of parameters relies on three main requirements; "A set of features obtained from the structure that represents the object of study; an objective function that needs to be optimized such as Maximum Likelihood Estimation (MLE), Conditional Maximum Likelihood Estimation (CMLE) or Maximum Mutual Information (MMI); and an established optimization method" [18].

The first requirement is satisfied by the labeling of samples through parenting of known characteristics of a set of proteins (Legume lectins). This parenting will be discussed in the section 4.3.

For the satisfaction of the second requirement, we know that discriminative training methods can be used as estimation models for PCFG by using what is known as the H-criteria, "a common framework for the representation of MLE, MMI and CMLE learning criteria" [18]. In this study, we propose a discriminative method for training parsers based in the generalization of the H-criteria. With this new framework we can consider multiple reference trees simultaneously and therefore we can acquire a compact PCFG obtained from several generative models.

Given a PCFG G_s in CNF, a training sample Ω and a set of derivations Δ_x for each $x \in \Omega$. The estimation of G_s is obtained by maximizing the following objective function [18]:

$$\tilde{F}_h(\Theta) = \frac{1}{|\Omega|} \sum_{x \in \Omega} log \frac{P_{G_s}(x, d_x)}{(\sum_{d_x \in \Delta_x} P_{G_s}(x, d_x))^h} \tag{4.18}$$

This function can be simplified as follows:

$$\tilde{F}_h(\Theta) = \frac{1}{|\Omega|} \prod_{x \in \Omega} log \frac{P_{G_s}(x, d_x)}{(P_{G_s}, \Delta_x(x))^{h'}} \tag{4.19}$$

Where $0 \leq h \leq 1$, and d_x is the derivation of a correct parsing. The sum in the denominator of equation 4.18 is the probability of x in regards of Δ_x, where

Δ_x defines a set of competing derivations. Finally, the h parameter establishes the degree that the competing derivations (denominator) discriminate against the derivation of reference (numerator). In this manner, an optimization of the H-criteria tries to simultaneously maximize the numerator $P_{G_s}(x, d_x)$ and minimize the denominator $P_{G_s}(x, d_x))^h$ for each string x in the training sample.

It is important to point out that in the equation 4.19 we consider that each sample string has only one possible reference parsing (numerator), although it is possible that some strings have more than one reference parsing. For this reason we modify the H-criterion to consider all the possibilities as follows [18]:

$$F_h(\Theta) = \prod_{x \in \Omega} \frac{P^{\eta}_{G_s}, \Delta^r_x(x)}{P^{\eta}_{G_s}, \Delta^c_x(x)^h} \tag{4.20}$$

This formula can be rewritten as:

$$F_h(\Theta) = \frac{\prod_{x \in \Omega} P^{\eta}_{G_s}, \Delta^r_x(x)}{\prod_{x \in \Omega} P^{\eta}_{G_s}, \Delta^c_x(x)^{h'}} \tag{4.21}$$

Where $0 < \eta$, $0 \leq h \leq 1$ and $\Delta^r_x \subseteq \Delta^c_x$. The set Δ^r_x must contain only derivations of the correct parsing of the sentence x, while the set Δ^c_x contains competing derivations of any parsing of the string x. On the other hand, the values chosen for the parameters η and h determine which criteria is represented. If the parameters are $\eta = 1$ and $h = 0$ then the MLE criteria is obtained, and if the parameters have a fixed value of $\eta = h = 1$ then the CMLE criteria is obtained. For this reason, if $h > 0$ the H-criteria can be viewed as a discriminative training method.

Now that we have explained the objective function, we are going to inquire in the method of optimization. In this study, the growth transformation was used as a method for the optimization of H-criteria. The growth transformation framework is a maximization framework where a set of parameters Θ is iteratively transformed into a new set of paremeters Θ' such that $F_h(\Theta') > F_h(\Theta)$ [18]. "This process is carried in two steps on the initial SCFG until a local maximum is achieved" [19]. For each iteration, the set Δ_x is computed for each $x \in \Omega$, in regards to the selected criterion and then the transformation 4.22 is applied and a new SCFG is obtained. In order to asses the adequate election of Δ_x we need a merit function that has to increase from iteration to iteration. The transformation method and merit function are presented below [19]:

Transformation method

$$\tilde{p}(A \to \alpha) = \frac{\sum_{x \in \Omega} \frac{1}{Pr_{G_s}(x, \Delta x)} \sum_{\forall d_x \in \Delta_x} N(A \to \alpha, d_x) Pr_{G_s}(x, d_x)}{\sum_{x \in \Omega} \frac{1}{Pr_{G_s}(x, \Delta x)} \sum_{\forall d_x \in \Delta_x} N(A, d_x) Pr_{G_s}(x, d_x)} \tag{4.22}$$

Merit function

$$Pr_{G_s}(\Omega, \Delta_\Omega) = \prod_{x \in \Omega} Pr_{G_s}(x, \Delta_x) \tag{4.23}$$

It is important to note that the merit function defines a family of functions that depend on the Δ_x. "This expression coincides with the likelihood of the best parse of the sample when Δ_x has only the best derivation of each string" [19]. Referring to the growth transformation, as this method is a gradient-ascent technique, the initial probabilities greatly influence the maximum that is achieved. For this reason, several strategies for obtaining initial grammars have been proposed but for this study we will use a "heuristic initialization based on an exhaustive ergodic model with probabilites randomly generated" [19].

Finally, this type of transformation allows for the use of different estimation algorithms depending on the set of Δ_x. In this regard, the transformation coincides with IO algorithm when Δ_x has all the derivations of each x, while it coincides with VS (Viterbi Score) algorithm when Δ_x has only the best derivation over all possible derivations [19].

4.3 Novel Method Proposal For Protein Characterization

In this section, we are going to present the methodology that was carried out in order to characterize the family of legume lectins. To do so, we will present the patterns that tend to appear in this family and that were chosen for said characterization. Also, we will establish the parenting method that allows for the recognition of compostion and structural information.

Protein homology, is the understanding that sequence similarity between two given proteins is a strong evidence that the sequences have a shared evolutionary tree. In this regard, proteins tend to have structures that are highly conserved and shared in a protein family.

In the case of legume lectins, we already established in section 3.2, that the proteins pertaining to this family tend to have a higher concentration of beta strands that interact to form beta sheets. In this manner, we found that this beta strands were highly conserved throughout the family, but also that some of this beta strands formed pairs with each other throughout the set. For this reason, it was of crucial importance identifying this conserved beta strands, as we could obtain relevant structural information and produce a model that was useful for a correct classification of this proteins. The patterns that represent this beta strands (using regex) are the following:

- $[LIV] - [STAG] - V - [DEQV] - [FLI] - D - [ST]$
- $[FL] - I - L - Q - [SG]$
- $L - [QE] - L - T$
- $G - R - A - L - [FY] - [YASP]$

- $K - V - G - T - A - H - I - [IS] - Y - N$
- $[PRQ] - H - I - G - I - [DN] - [IV] - [NK] - [ST] - [VIL] - [KIR]$
- $[DS] - S - A - T - V - S - Y - D$
- $G - [LI] - [ATV] - [FW] - F - [FIA] - [LAS]$
- $R - L - S - A - [VI] - V - S - Y$
- $[DTN] - [VI] - L - S - W - S - F - [FTESAD] - [AS] - [SNK] - [NLFP] - [FKDSIPN]$

In these patterns, each of the letters that appear represents one amino acid. In the case of the amino acids that appear between the square brackets, only one of them can be chosen at a time. On the other hand, the amino acids that appear without square brackets were found to always be in that given position. An example chain that could match with one of the suggested patterns is the following:

$$[FL] - I - L - Q - [SG] \Rightarrow FILQS \tag{4.24}$$

Of all the patterns that were used, there is one that is present in all of the proteins that are classified in the legume lectin family, the pattern $[LIV] - [STAG] - V - [DEQV] - [FLI] - D - [ST]$. For this reason, this pattern is used by sites like Prosite (https://prosite.expasy.org/), as a consesus pattern which determines if a given protein is classified as a legume lectin. This type of approach presents a severe problematic, as a mutation or change in any of the amino acids that appear in the pattern will prevent a protein from been classified correctly. For this reason, we decided to use not only the consensus pattern, but also the rest of the patterns that have been presented, in order to estimate the probability for a given protein to belong to the legume lectin family based on the appearance of these patterns. These patterns were introduced in the initial grammar, as can be seen in the Appendix A.

Once the patterns were determined, the next step was to label the proteins pertaining to the legume lectin family. In this manner, each of the proteins was investigated and labelled using a parenting method. With this method, we established not only the presence of the beta strands that are represented in the pattern, but also the interaction that this beta strands had between them (Figure 4.3). Following the example of Figure 4.3 the parenting of the this protein would be as follows:

$$[\ [\ V\ A\ V\ V\ F\ D\ T\ ]...[\ G\ L\ A\ F\ F\ A\ L\ ]\ ] \tag{4.25}$$

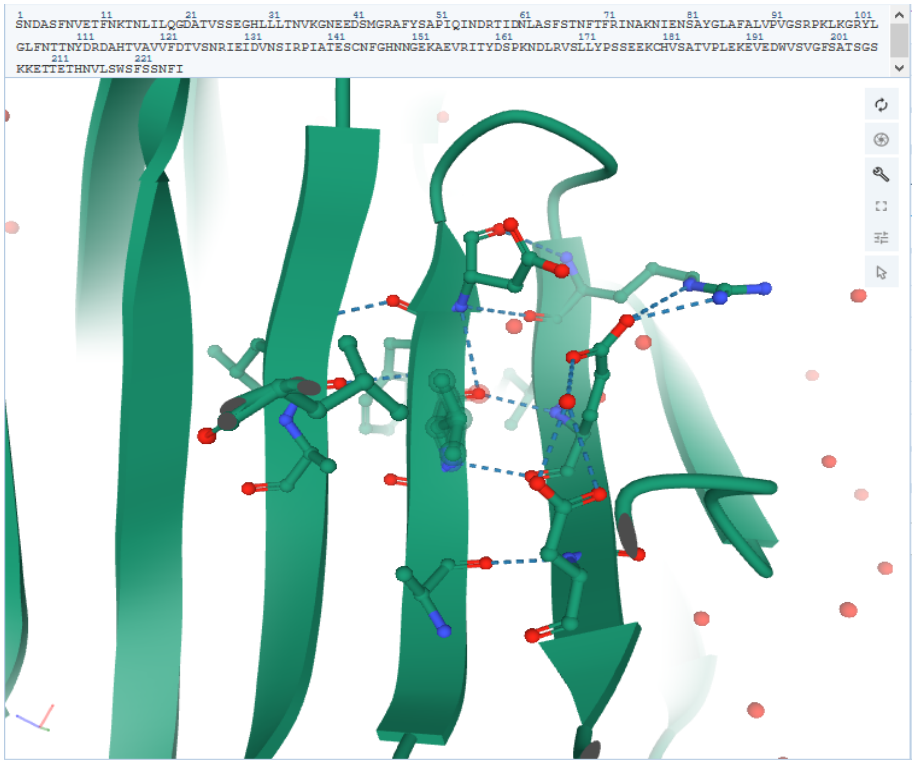


Figure 4.3: Interaction between two beta strands. As we can see by the blue dotted lines, there is an interaction between these two strands. The left strand is composed by the following chain $VAVVFDT$, that matches with the pattern $[LIV] - [STAG] - V - [DEQV] - [FLI] - D - [ST]$. The right strand is composed by the chain $GLAFFAL$, that matches with the pattern $G - [LI] - [ATV] - [FW] - F - [FIA] - [LAS]$.

In this manner, the appearance of a beta strand that matches the pattern is represented by the use of square brackets ($[\ V\ A\ V\ V\ F\ D\ T\ ]$). Meanwhile, the second set of square brackets represents that there is an interaction between these two beta strands. Thanks to this method of parenting we obtain the composition and structural information of the protein, that we can then use to classify it.

CHAPTER 5
Experimental Evaluation

In this chapter we will give a major insight of the corpus composition. Also, we will present the metrics used to evaluate the classification process and the main processes of operation that were used in the corpus.

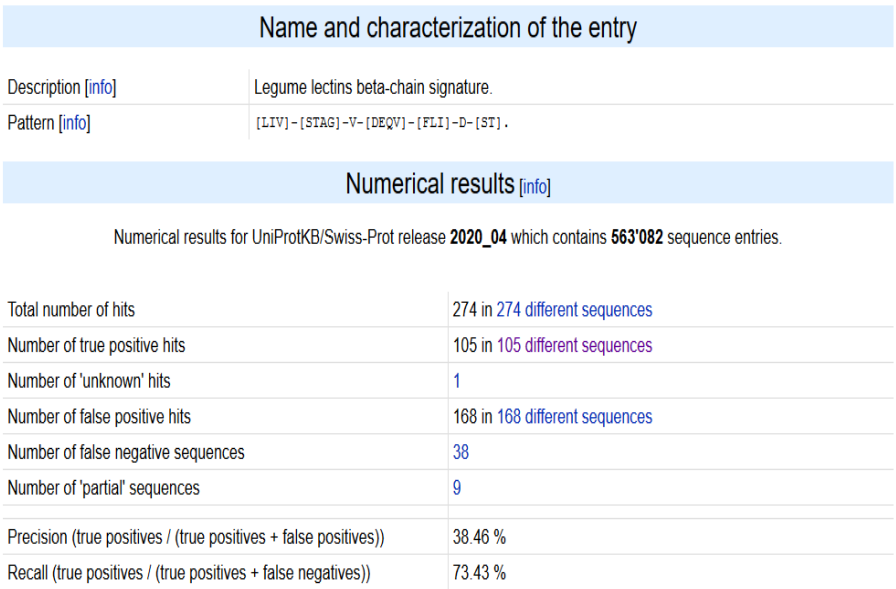
5.1 The PS00307 Corpus

The PS00307 corpus (legume lectins) is a collection of proteins that have been classified by different groups of researchers and published by Prosite (https://prosite.expasy.org/PS00307#TP). It is composed by a total of 312 proteins, that pertain to one of the following three groups:

- True positives : This group contains 105 entries of proteins. This set represents those proteins that contain the pattern $[LIV]-[STAG]-V-[DEQV]-[FLI]-D-[ST]$ and have been correctly classified as legume lectins.
- False negatives: This group contains 38 entries of proteins. This set represents those proteins that do not contain the pattern $[LIV]-[STAG]-V-[DEQV]-[FLI]-D-[ST]$, but are considered to be part of the legume lectins family.
- False positives: This group contains 168 entries of proteins. This set represents those proteins that contain the pattern $[LIV]-[STAG]-V-[DEQV]-[FLI]-D-[ST]$ and are incorrectly classified as legume lectins.

As already mentioned in section 4.3, the use of a consensus pattern gives rise to a problem of classification, as a simple mutation or change in the amino acid pattern produces an incorrect labelling of the proteins. Nevertheless, Prosite also presents different metrics that allow for a more clear comprehension of the effectiveness of their classification (Figure 5.1).

In this study, we only used the true positives and false negatives as these two groups represent the positive samples required for the training process. In Chapter 6, we will discuss the possibility of extending this line of investigation to also include the use of false positive, that will represent the negative samples.



Name and characterization of the entry

Description [info]	Legume lectins beta-chain signature.
Pattern [info]	[LIV]-[STAG]-V-[DEQV]-[FLI]-D-[ST].

Numerical results [info]

Numerical results for UniProtKB/Swiss-Prot release **2020_04** which contains **563'082** sequence entries.

Total number of hits	274 in 274 different sequences
Number of true positive hits	105 in 105 different sequences
Number of 'unknown' hits	1
Number of false positive hits	168 in 168 different sequences
Number of false negative sequences	38
Number of 'partial' sequences	9
Precision (true positives / (true positives + false positives))	38.46 %
Recall (true positives / (true positives + false negatives))	73.43 %

Figure 5.1: Metrics used in Prosite. As we can see the Precision is substantially low, which makes the finding of a new method of classification an interesting case of study. On the other hand, the Recall value is significantly high, although is partially due to the fact that the number of proteins that pertain to the false negative group is considerably low.

5.2 Evaluation Metrics

The main metric that was used for the evaluation of the classification process was the Classification Accuracy. "This metric is a straightforward paradigm that represents the ratios of samples that a classifier correctly recognizes" [20]. The formula for the calculation of the Classifier Accuracy is quite simple [20]:

$$CR = \frac{C}{A} \times 100 \tag{5.1}$$

Where CR represents the correct rate, C is the number of samples recognized correctly and A is the number of samples.

Although, we could also had calculated the Recall obtained after classifying the proteins, we will discuss the main problematic that did not allow for this calculation in the following section.

5.3 Experimental Results with the PS00307 Corpus

The corpus used for this study was composed exclusively of positive samples from the Prosite web page.

In this manner, the corpus was formed by 135 proteins that were subsequently distributed into nine different partitions. The main reason behind this approach was that we decide to train the models with the use of the k-fold cross-validation. This technique uses different subsets of limited data to estimate the skill of a machine learning model on unseen data.

In this manner, the nine partitions were created by the random shuffle of all the positive samples and then distributed evenly.

Once the random partition was carried out, the next step was the training of the SCFG. In order to do so we carried out the following steps:

- Select a fraction of the partition as a test set.
- Use the remaining proteins of the partition as a training set.
- Fit the SCFG model on the training set and evaluate it on the test set.

For each of the partitions, a total of 15 iterations were carried out in the PRHLT cluster (Intel(R) Core(TM) i7-7800X CPU @ 3.50GHz) and evaluated by using the classifier accuracy. The results can be seen here:

Table 5.1: Evaluation based on the classification accuracy (%) of the partitions.

it.	P1	P2	P3	P4	P5	P6	P7	P8	P9	Average
1	53.3	66.7	53.3	33.3	53.3	53.3	53.3	53.3	40	51.1
2	53.3	66.7	53.3	33.3	53.3	53.3	53.3	53.3	40	51.1
3	46.67	66.7	53.3	33.3	53.3	53.3	53.3	53.3	40	50.4
4	46.67	66.7	53.3	33.3	53.3	53.3	53.3	53.3	40	50.4
4	46.67	66.7	53.3	33.3	53.3	53.3	53.3	53.3	40	50.4
5	46.67	66.7	53.3	33.3	53.3	53.3	53.3	53.3	40	50.4
6	46.67	66.7	53.3	33.3	53.3	53.3	53.3	53.3	40	50.4
7	46.67	66.7	53.3	33.3	53.3	53.3	53.3	53.3	40	50.4
8	46.67	66.7	53.3	33.3	53.3	53.3	53.3	53.3	40	50.4
9	46.67	66.7	53.3	33.3	60	53.3	53.3	53.3	40	51.1
10	46.67	66.7	53.3	33.3	60	53.3	53.3	53.3	40	51.1
11	46.67	66.7	53.3	33.3	60	53.3	53.3	53.3	40	51.1
12	46.67	66.7	53.3	33.3	60	53.3	53.3	53.3	40	51.1
13	46.67	66.7	53.3	33.3	60	53.3	53.3	53.3	40	51.1
14	46.67	66.7	53.3	33.3	60	53.3	53.3	53.3	40	51.1
15	46.67	66.7	53.3	33.3	60	53.3	53.3	53.3	40	51.1

As we can see by the results, the number of iterations is not significant to the classification accuracy or more iterations should be carried out in order to see an increase in the number of proteins correctly classified. If we compare the best result of the best average classification accuracy (51.1 %) to the the accuracy of Prosite (67.0 %) [6], we can see that there is room for improvement. Also, we can calculate the best classification for each of the partitions and we obtain an average classifier accuracy of 51.8 %, but still falls short when compared to the Prosite results. Nevertheless, it is important to note that although the accuracy does not increase in each iteration, the value of the objective function (equation 4.18 through 4.21) is optimized in each iteration, as demonstrated by the fluctuations in the accuracy.

For this reason, in the next chapter we will propose a series of experiments that should be performed to see if we can increase the capability of classification and acquire more valuable results.

CHAPTER 6
Conclusion and Future Work

6.1 Conclusion

In this this study, we have introduced a new method for the analysis and characterization of proteins. This was possible through the understanding of Chomsky's hierarchy and comprehending that SCFG are able to detect the composition and structural information of a set of proteins.

In order to acquire this information, a biological context where the interactions between the different structures of proteins are explained had to be introduced, to understand the nuances that govern the proteomic world. Again, through the biological landscape that was introduced, we were able to select a group of proteins, the legume lectins, by establishing their unique physicochemical characteristics that make them an interesting case of study.

Once the object of study was chosen, a corpus of 135 proteins representing the structural information was constructed to perform a series of discriminative estimation processes and obtain different SCFG models. With the estimated models, we calculated the classifier accuracy and discovered that new experiments, that could not be carried out due to a lack of time caused by initial problems with the software, had to be performed in order to increase the classification score.

6.2 Future Work

As already stated, the results of the classification were underwhelming. For this reason the line of work should be expanded by adding new experiments such as:

- Experiments changing the learning criteria, by the modification of the h parameter.
- Experimentation including negative samples.
- Comparison with the classifiers of W. Dyrka and Prosite [6].
- Increase in the number of iterations in the training process.

Finally, we should perform a modification centered in the patterns that were used. The main problematic with the patterns is that some of them were too strict and more variability should be introduced . In this regard, we present a new set of patterns that increase the variability:

- $[FLI] - [IKTL] - [LFT] - [QF] - [SGRKE]$
- $L - [QETVLRAK] - L - [TN]$
- $[GKS] - R - [AVTY] - [LTF] - [FYS] - [YASRTVQP] - [AKTPLSMYDQN]$
- $K - [IVT] - [AG] - T - [VA] - H - I[ISN] - Y - N$
- $H - [IVML] - G - [FVI] - [DN] - [IVTALEN] - [NKDS] - [STGRCGN] - [VILPAM]$
- $[DS] - S - [AT] - T - V - S - Y - D$
- $[GEA] - [LIFM] - [ATIVLC] - [FWRA][VFAYITL][FIALVM][LASPVFGC]$
- $R - L - [TS] - [AV] - [VI] - V - S - Y$
- $[QEDPK] - [WRKSYD] - V - [RISD] - [VIFP]G[FL][TS][TSAGP]$
- $[DTNEGLRYFK] - [VIL] - [FLEYHQRM] - [ASYNDG] - W - [STYHN] - F - [FTESADHLKQRNG] - [ASTKILRNM] - [SNKTEVG] - [NLFPISMREGFRK]$

Acknowledgements

I would like to give special thanks to my tutors José Miguel Benedí Ruiz and Joan Andreu Sánchez Peiró for their continous implication, insight, motivation and patience towards the completion of this work.

Also, I would like to thank the constant support, empathy and care of my friends and family, without them this journey would had been extremely arduous.

Bibliography

[1] J. Slonczeswick, *Genetics and Development*. Kenyon college, 2006, http://biology.kenyon.edu/courses/biol114/biol114_fall_sec0.html.

[2] H. ThienLuan, O. Seung-Rohk, and K. HyunJin, "A parallel approximate string matching under levenshtein distance on graphics processing units using warp-shuffle operations," *Plos One*, vol. 12, no. 10, 2017, https://doi.org/10.1371/journal.pone.0186251.

[3] S. Karlin and H. M. Taylor, *A First Course on Stochastic Processes*. Mathematical association of America, 1966, doi: 10.2307/2314395.

[4] G. Jäger and J. Rogers, "Formal language theory: refining the chomsky hierarchy," *Philos Trans R Soc Lond B Biol Sci.*, vol. 367, no. 1598, pp. 1956–1970, 2012, doi:10.1098/rstb.2012.0077.

[5] F. Coste and G. Kerbellec, "Learning automata on protein sequences," *7th Journées Ouvertes Biologie Informatique Mathématiques*, pp. 199–210, 2006, doi:10.1098/rstb.2012.0077.

[6] W. Dyrka and J.-C. Nebel, "Stochastic context free grammar based framework for analysis of protein sequences," *BMC Bioinformatics*, vol. 10, no. 323, 2009, https://doi.org/10.1186/1471-2105-10-323.

[7] A. Lehninger, D. Nelson, and M. Cox, *Lehninger Principles of Biochemistry*, 5th ed. Wh Freeman, 2008, https://doi.org/10.1007/978-3-662-08289-8.

[8] J. A. Sánchez, "Estimación de gramáticas incontextuales probabilísticas y su aplicación en modelización del lenguaje," *PhD Thesis for the Universidad Politécnica de Valencia*, 1999.

[9] A. Lesk, "Bioinformatics," *Encyclopaedia Britannica*, 2019, https://www.britannica.com/science/bioinformatics.

[10] P. Hogeweg, "The roots of bioinformatics in theoretical biology," vol. 7, no. 3, 2011, https://doi.org/10.1371/journal.pcbi.10020213.

[11] G. Navarro, "A guided tour to approximate string matching," *ACM Computing Surveys*, vol. 33, no. 1, pp. 31–88, 2001, https://doi.org/10.1145/375360.375365.

[12] S. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *J Theor Biol*, vol. 22, no. 3, p. 437-467, 1969, doi:10.1016/0022-5193(69)90015-0.

[13] D. Chao and S. Kou, "Correlation analysis of enzymatic reaction of a single protein molecule," *Ann Appl Stat.*, vol. 6, no. 3, pp. 950–976, 2012, doi: 10.1214/12-AOAS541.

[14] A. Gupta and J. Rawlings, "Comparison of parameter estimation methods in stochastic chemical kinetic models: examples in systems biology," *Ann Appl Stat.*, vol. 60, no. 4, pp. 1253–1268, 2014, doi: 10.1002/aic.14409.

[15] D. Pratas, R. Silva, P. Armando, and P. Ferreira, "An alignment-free method to find and visualise rearrangements between pairs of dna sequences," *Scientific Reports*, vol. 5, no. 10203, 2015, doi: 10.1038/srep10203.

[16] N. Sharon and H. Lis, "Legume lectins - a large family of homologous proteins," *FASEB J.*, vol. 4, no. 14, pp. 3198–3208, 1990, doi:10.1096/fasebj.4.14.2227211.

[17] K. V. e. a. Brinda, "Determinants of quaternary association in legume lectins," *Protein science : a publication of the Protein Society*, vol. 13, no. 7, pp. 1735–1749, 2004, doi:10.1110/ps.04651004.

[18] J. M. Benedí, J. A. Sánchez, and M. Maca, "Discriminative training for probabilistic context-free grammars," *Departamento de Sistemas Informáticos y Computación, Universitat Politècnica de València*, 2020.

[19] J. M. Benedí and J. A. Sánchez, "Estimation of stochastic context-free grammars and their use as language models," *Computer Speech and Language*, vol. 19, pp. 249–274, 2005, doi:10.1016/j.csl.2004.09.001.

[20] D. Wong, "Optimization of bagging classifiers based on sbcb algorithm," *Proceedings of the Ninth International Conference on Machine Learning and Cybernetics*, 2010, doi: 10.1109/ICMLC.2010.5581054.

APPENDIX A
Initial Grammar

A.1 Initial grammar

- P0 is the axiom and defines the start of the molecule, P1 defines the end, P2 and P3 allow for some recursivity.
- N0-N1 and M0-M4 are non terminals and are used to describe the patterns, beta strands and variable segments.
- F0-F2 define the end of the molecule.
- X0-X1 define variable segments
- B0-B1 define the beta strands
- R0-R9 define the patterns that were found in the samples.

```
# PCFG handmade from positive samples
#
NonTerminals 130
P0
P1
P2
P3
N0
N1
M0
M1
M2
M3
M4
F0
F1
F2
X0
X1
B0
B1
```

R0
R0A
R0B
R0C
R0D
R0E
R1
R1A
R1B
R1C
R2
R2A
R2B
R3
R3A
R3B
R3C
R3D
R4
R4A
R4B
R4C
R4D
R4E
R4F
R4G
R4H
R5
R5A
R5B
R5C
R5D
R5E
R5F
R5G
R5H
R5I
R6
R6A
R6B
R6C
R6D
R6E
R6F
R7
R7A
R7B
R7C
R7D
R7E
R8
R8A

R8B
R8C
R8D
R8E
R8F
R9
R9A
R9B
R9C
R9D
R9E
R9F
R9G
R9H
R9I
R9J
RA
RD
RF
RG
RH
RI
RK
RL
RN
RQ
RR
RS
RT
RV
RW
RY
RAS
RDN
RDS
RFL
RFW
RFY
RIS
RIV
RLI
RNK
RQE
RSG
RST
RATV
RDTN
RFIA
RFLI
RKIR
RLAS
RLIV

```
RPRQ
RSNK
RDEQV
RNLFP
RSTAG
RYASP
RFTE6
RFKD7

Terminals 20
A
C
D
E
F
G
H
I
K
L
M
N
P
Q
R
S
T
V
W
Y

Rules
0.8000 P0 --> X0 P1
0.2000 P0 --> X0 P2
0.5000 P2 --> B0 P3
0.4991 P2 --> R3 P3
0.0001 P2 --> R0 P3
0.0001 P2 --> R1 P3
0.0001 P2 --> R2 P3
0.0001 P2 --> R4 P3
0.0001 P2 --> R5 P3
0.0001 P2 --> R6 P3
0.0001 P2 --> R7 P3
0.0001 P2 --> R8 P3
0.0001 P2 --> R9 P3
0.6000 P3 --> X0 P1
0.4000 P3 --> X0 P2
0.4000 P1 --> N0 F0
0.6000 P1 --> N0 X0
#-----------------------
0.4000 N0 --> B0 N1
0.3000 N0 --> M0 N1
```

```
0.0300 N0 --> R0 N1
0.0300 N0 --> R1 N1
0.0300 N0 --> R2 N1
0.0300 N0 --> R3 N1
0.0300 N0 --> R4 N1
0.0300 N0 --> R5 N1
0.0300 N0 --> R6 N1
0.0300 N0 --> R7 N1
0.0300 N0 --> R8 N1
0.0300 N0 --> R9 N1
0.3992 N1 --> X0 N0
0.1200 N1 --> X0 B0
0.1200 N1 --> X0 M0
0.1500 N1 --> X0 R7
0.2100 N1 --> X0 R9
0.0001 N1 --> X0 R0
0.0001 N1 --> X0 R1
0.0001 N1 --> X0 R2
0.0001 N1 --> X0 R3
0.0001 N1 --> X0 R4
0.0001 N1 --> X0 R5
0.0001 N1 --> X0 R6
0.0001 N1 --> X0 R8
#-----------------------
0.5101 M0 --> B0 M1
0.0204 M0 --> R0 M1
0.0612 M0 --> R1 M1
0.0204 M0 --> R2 M1
0.0001 M0 --> R3 M1
0.0001 M0 --> R4 M1
0.1632 M0 --> R5 M1
0.0001 M0 --> R6 M1
0.0816 M0 --> R7 M1
0.0816 M0 --> R8 M1
0.0612 M0 --> R9 M1
0.4000 M1 --> X0 M2
0.3930 M1 --> X0 B0
0.1003 M1 --> X0 R0
0.0001 M1 --> X0 R1
0.0020 M1 --> X0 R2
0.0020 M1 --> X0 R3
0.0001 M1 --> X0 R4
0.0001 M1 --> X0 R5
0.1002 M1 --> X0 R6
0.0020 M1 --> X0 R7
0.0001 M1 --> X0 R8
0.0001 M1 --> X0 R9
0.5829 M2 --> B0 M3
0.0001 M2 --> R0 M3
0.1665 M2 --> R1 M3
0.2498 M2 --> R2 M3
0.0001 M2 --> R3 M3
```

```
0.0001 M2 --> R4 M3
0.0001 M2 --> R5 M3
0.0001 M2 --> R6 M3
0.0001 M2 --> R7 M3
0.0001 M2 --> R8 M3
0.0001 M2 --> R9 M3
0.3000 M3 --> X0 M4
0.2000 M3 --> X0 B0
0.0500 M3 --> X0 R0
0.0500 M3 --> X0 R1
0.0500 M3 --> X0 R2
0.0500 M3 --> X0 R3
0.0500 M3 --> X0 R4
0.0500 M3 --> X0 R5
0.0500 M3 --> X0 R6
0.0500 M3 --> X0 R7
0.0500 M3 --> X0 R8
0.0500 M3 --> X0 R9
0.1000 M4 --> B0 M3
0.0900 M4 --> R0 M3
0.0900 M4 --> R1 M3
0.0900 M4 --> R2 M3
0.0900 M4 --> R3 M3
0.0900 M4 --> R4 M3
0.0900 M4 --> R5 M3
0.0900 M4 --> R6 M3
0.0900 M4 --> R7 M3
0.0900 M4 --> R8 M3
0.0900 M4 --> R9 M3
#----------------------
1.0000 F0 --> X0 F1
0.1000 F1 --> B0 F2
0.2500 F1 --> B0 X0
0.1984 F1 --> R9 F2
0.4500 F1 --> R9 X0
0.0001 F1 --> R0 F2
0.0001 F1 --> R1 F2
0.0001 F1 --> R2 F2
0.0001 F1 --> R3 F2
0.0001 F1 --> R4 F2
0.0001 F1 --> R5 F2
0.0001 F1 --> R6 F2
0.0001 F1 --> R7 F2
0.0001 F1 --> R8 F2
0.0001 F1 --> R0 X0
0.0001 F1 --> R1 X0
0.0001 F1 --> R2 X0
0.0001 F1 --> R3 X0
0.0001 F1 --> R4 X0
0.0001 F1 --> R5 X0
0.0001 F1 --> R6 X0
0.0001 F1 --> R7 X0
```

```
0.0001 F1 --> R8 X0
1.0000 F2 --> X0 F1
#----------------------
0.6000 X0 --> X0 X1
0.0200 X0 --> A
0.0200 X0 --> C
0.0200 X0 --> D
0.0200 X0 --> E
0.0200 X0 --> F
0.0200 X0 --> G
0.0200 X0 --> H
0.0200 X0 --> I
0.0200 X0 --> K
0.0200 X0 --> L
0.0200 X0 --> M
0.0200 X0 --> N
0.0200 X0 --> P
0.0200 X0 --> Q
0.0200 X0 --> R
0.0200 X0 --> S
0.0200 X0 --> T
0.0200 X0 --> V
0.0200 X0 --> W
0.0200 X0 --> Y
0.0500 X1 --> A
0.0500 X1 --> C
0.0500 X1 --> D
0.0500 X1 --> E
0.0500 X1 --> F
0.0500 X1 --> G
0.0500 X1 --> H
0.0500 X1 --> I
0.0500 X1 --> K
0.0500 X1 --> L
0.0500 X1 --> M
0.0500 X1 --> N
0.0500 X1 --> P
0.0500 X1 --> Q
0.0500 X1 --> R
0.0500 X1 --> S
0.0500 X1 --> T
0.0500 X1 --> V
0.0500 X1 --> W
0.0500 X1 --> Y
#----------------------
0.6000 B0 --> B0 B1
0.0200 B0 --> A
0.0200 B0 --> C
0.0200 B0 --> D
0.0200 B0 --> E
0.0200 B0 --> F
0.0200 B0 --> G
```

```
0.0200 B0 --> H
0.0200 B0 --> I
0.0200 B0 --> K
0.0200 B0 --> L
0.0200 B0 --> M
0.0200 B0 --> N
0.0200 B0 --> P
0.0200 B0 --> Q
0.0200 B0 --> R
0.0200 B0 --> S
0.0200 B0 --> T
0.0200 B0 --> V
0.0200 B0 --> W
0.0200 B0 --> Y
0.0500 B1 --> A
0.0500 B1 --> C
0.0500 B1 --> D
0.0500 B1 --> E
0.0500 B1 --> F
0.0500 B1 --> G
0.0500 B1 --> H
0.0500 B1 --> I
0.0500 B1 --> K
0.0500 B1 --> L
0.0500 B1 --> M
0.0500 B1 --> N
0.0500 B1 --> P
0.0500 B1 --> Q
0.0500 B1 --> R
0.0500 B1 --> S
0.0500 B1 --> T
0.0500 B1 --> V
0.0500 B1 --> W
0.0500 B1 --> Y
#
# EXPRESIONES REGULARES DE PATRONES
#----------------------------------
# R0 = [LIV][STAG]V[DEQV][FLI]D[ST]
1.0000 R0  --> RLIV  R0A
1.0000 R0A --> RSTAG R0B
1.0000 R0B --> RV    R0C
1.0000 R0C --> RDEQV R0D
1.0000 R0D --> RFLI  R0E
1.0000 R0E --> RD    RST
#---------------------------
# R1 = [FL]ILQ[SG]
1.0000 R1  --> RFL   R1A
1.0000 R1A --> RI    R1B
1.0000 R1B --> RL    R1C
1.0000 R1C --> RQ    RSG
#---------------------------
# R2 = L[QE]LT
```

```
1.0000 R2  --> RL    R2A
1.0000 R2A --> RQE   R2B
1.0000 R2B --> RL    RT
#--------------------------
# R3 = GRAL[FY][YASP]
1.0000 R3  --> RG    R3A
1.0000 R3A --> RR    R3B
1.0000 R3B --> RA    R3C
1.0000 R3C --> RL    R3D
1.0000 R3D --> RFY   RYASP
#--------------------------
# R4 = KVGTAHI[IS]YN
1.0000 R4  --> RK    R4A
1.0000 R4A --> RV    R4B
1.0000 R4B --> RG    R4C
1.0000 R4C --> RT    R4D
1.0000 R4D --> RA    R4E
1.0000 R4E --> RH    R4F
1.0000 R4F --> RI    R4G
1.0000 R4G --> RIS   R4H
1.0000 R4H --> RY    RN
#--------------------------
# R5 = [PRQ]HIGI[DN][IV][NK][ST][VIL][KIR]
1.0000 R5  --> RPRQ  R5A
1.0000 R5A --> RH    R5B
1.0000 R5B --> RI    R5C
1.0000 R5C --> RG    R5D
1.0000 R5D --> RI    R5E
1.0000 R5E --> RDN   R5F
1.0000 R5F --> RIV   R5G
1.0000 R5G --> RNK   R5H
1.0000 R5H --> RST   R5I
1.0000 R5I --> RLIV  RKIR
#--------------------------
# R6 = [DS]SATVSYD
1.0000 R6  --> RDS   R6A
1.0000 R6A --> RS    R6B
1.0000 R6B --> RA    R6C
1.0000 R6C --> RT    R6D
1.0000 R6D --> RV    R6E
1.0000 R6E --> RS    R6F
1.0000 R6F --> RY    RD
#--------------------------
# R7 = G[LI][ATV][FW]F[FIA][LAS]
1.0000 R7  --> RG    R7A
1.0000 R7A --> RLI   R7B
1.0000 R7B --> RATV  R7C
1.0000 R7C --> RFW   R7D
1.0000 R7D --> RF    R7E
1.0000 R7E --> RFIA  RLAS
#--------------------------
# R8 = RLSA[VI]VSY
```

```
1.0000 R8  --> RR    R8A
1.0000 R8A --> RL    R8B
1.0000 R8B --> RS    R8C
1.0000 R8C --> RA    R8D
1.0000 R8D --> RIV   R8E
1.0000 R8E --> RV    R5F
1.0000 R8F --> RS    RY
#---------------------------
# R9 = [DTN][VI]LSWSF[FTESAD][AS][SNK][NLFP][FKDSIPN]
1.0000 R9  --> RDTN  R9A
1.0000 R9A --> RIV   R9B
1.0000 R9B --> RL    R9C
1.0000 R9C --> RS    R9D
1.0000 R9D --> RW    R9E
1.0000 R9E --> RS    R9F
1.0000 R9F --> RF    R9G
1.0000 R9G --> RFTE6 R9H
1.0000 R9H --> RAS   R9I
1.0000 R9I --> RSNK  R9J
1.0000 R9J --> RNLFP RFKD7
#-------------------------------------------------
# RA   A               1
0.9981 RA  --> A
0.0001 RA  --> C
0.0001 RA  --> D
0.0001 RA  --> E
0.0001 RA  --> F
0.0001 RA  --> G
0.0001 RA  --> H
0.0001 RA  --> I
0.0001 RA  --> K
0.0001 RA  --> L
0.0001 RA  --> M
0.0001 RA  --> N
0.0001 RA  --> P
0.0001 RA  --> Q
0.0001 RA  --> R
0.0001 RA  --> S
0.0001 RA  --> T
0.0001 RA  --> V
0.0001 RA  --> W
0.0001 RA  --> Y
#-----------------------
# RD   D               1
0.0001 RD  --> A
0.0001 RD  --> C
0.9981 RD  --> D
0.0001 RD  --> E
0.0001 RD  --> F
0.0001 RD  --> G
0.0001 RD  --> H
0.0001 RD  --> I
```

```
0.0001 RD  --> K
0.0001 RD  --> L
0.0001 RD  --> M
0.0001 RD  --> N
0.0001 RD  --> P
0.0001 RD  --> Q
0.0001 RD  --> R
0.0001 RD  --> S
0.0001 RD  --> T
0.0001 RD  --> V
0.0001 RD  --> W
0.0001 RD  --> Y
#-----------------------
# RF   F               1
0.0001 RF  --> A
0.0001 RF  --> C
0.0001 RF  --> D
0.0001 RF  --> E
0.9981 RF  --> F
0.0001 RF  --> G
0.0001 RF  --> H
0.0001 RF  --> I
0.0001 RF  --> K
0.0001 RF  --> L
0.0001 RF  --> M
0.0001 RF  --> N
0.0001 RF  --> P
0.0001 RF  --> Q
0.0001 RF  --> R
0.0001 RF  --> S
0.0001 RF  --> T
0.0001 RF  --> V
0.0001 RF  --> W
0.0001 RF  --> Y
#-----------------------
# RG   G               1
0.0001 RG  --> A
0.0001 RG  --> C
0.0001 RG  --> D
0.0001 RG  --> E
0.0001 RG  --> F
0.9981 RG  --> G
0.0001 RG  --> H
0.0001 RG  --> I
0.0001 RG  --> K
0.0001 RG  --> L
0.0001 RG  --> M
0.0001 RG  --> N
0.0001 RG  --> P
0.0001 RG  --> Q
0.0001 RG  --> R
0.0001 RG  --> S
```

```
0.0001 RG  --> T
0.0001 RG  --> V
0.0001 RG  --> W
0.0001 RG  --> Y
#-----------------------
# RH   H               1
0.0001 RH  --> A
0.0001 RH  --> C
0.0001 RH  --> D
0.0001 RH  --> E
0.0001 RH  --> F
0.0001 RH  --> G
0.9981 RH  --> H
0.0001 RH  --> I
0.0001 RH  --> K
0.0001 RH  --> L
0.0001 RH  --> M
0.0001 RH  --> N
0.0001 RH  --> P
0.0001 RH  --> Q
0.0001 RH  --> R
0.0001 RH  --> S
0.0001 RH  --> T
0.0001 RH  --> V
0.0001 RH  --> W
0.0001 RH  --> Y
#-----------------------
# RI   I               1
0.0001 RI  --> A
0.0001 RI  --> C
0.0001 RI  --> D
0.0001 RI  --> E
0.0001 RI  --> F
0.0001 RI  --> G
0.0001 RI  --> H
0.9981 RI  --> I
0.0001 RI  --> K
0.0001 RI  --> L
0.0001 RI  --> M
0.0001 RI  --> N
0.0001 RI  --> P
0.0001 RI  --> Q
0.0001 RI  --> R
0.0001 RI  --> S
0.0001 RI  --> T
0.0001 RI  --> V
0.0001 RI  --> W
0.0001 RI  --> Y
#-----------------------
# RK   K               1
0.0001 RK  --> A
0.0001 RK  --> C
```

```
0.0001 RK  --> D
0.0001 RK  --> E
0.0001 RK  --> F
0.0001 RK  --> G
0.0001 RK  --> H
0.0001 RK  --> I
0.9981 RK  --> K
0.0001 RK  --> L
0.0001 RK  --> M
0.0001 RK  --> N
0.0001 RK  --> P
0.0001 RK  --> Q
0.0001 RK  --> R
0.0001 RK  --> S
0.0001 RK  --> T
0.0001 RK  --> V
0.0001 RK  --> W
0.0001 RK  --> Y
#-----------------------
# RL   L               1
0.0001 RL  --> A
0.0001 RL  --> C
0.0001 RL  --> D
0.0001 RL  --> E
0.0001 RL  --> F
0.0001 RL  --> G
0.0001 RL  --> H
0.0001 RL  --> I
0.0001 RL  --> K
0.9981 RL  --> L
0.0001 RL  --> M
0.0001 RL  --> N
0.0001 RL  --> P
0.0001 RL  --> Q
0.0001 RL  --> R
0.0001 RL  --> S
0.0001 RL  --> T
0.0001 RL  --> V
0.0001 RL  --> W
0.0001 RL  --> Y
#-----------------------
# RN   N               1
0.0001 RN  --> A
0.0001 RN  --> C
0.0001 RN  --> D
0.0001 RN  --> E
0.0001 RN  --> F
0.0001 RN  --> G
0.0001 RN  --> H
0.0001 RN  --> I
0.0001 RN  --> K
0.0001 RN  --> L
```

```
0.0001 RN  --> M
0.9981 RN  --> N
0.0001 RN  --> P
0.0001 RN  --> Q
0.0001 RN  --> R
0.0001 RN  --> S
0.0001 RN  --> T
0.0001 RN  --> V
0.0001 RN  --> W
0.0001 RN  --> Y
#-----------------------
# RQ   Q               1
0.0001 RQ  --> A
0.0001 RQ  --> C
0.0001 RQ  --> D
0.0001 RQ  --> E
0.0001 RQ  --> F
0.0001 RQ  --> G
0.0001 RQ  --> H
0.0001 RQ  --> I
0.0001 RQ  --> K
0.0001 RQ  --> L
0.0001 RQ  --> M
0.0001 RQ  --> N
0.0001 RQ  --> P
0.9981 RQ  --> Q
0.0001 RQ  --> R
0.0001 RQ  --> S
0.0001 RQ  --> T
0.0001 RQ  --> V
0.0001 RQ  --> W
0.0001 RQ  --> Y
#-----------------------
# RR   R               1
0.0001 RR  --> A
0.0001 RR  --> C
0.0001 RR  --> D
0.0001 RR  --> E
0.0001 RR  --> F
0.0001 RR  --> G
0.0001 RR  --> H
0.0001 RR  --> I
0.0001 RR  --> K
0.0001 RR  --> L
0.0001 RR  --> M
0.0001 RR  --> N
0.0001 RR  --> P
0.0001 RR  --> Q
0.9981 RR  --> R
0.0001 RR  --> S
0.0001 RR  --> T
0.0001 RR  --> V
```

```
0.0001 RR  --> W
0.0001 RR  --> Y
#-----------------------
# RS   S               1
0.0001 RS  --> A
0.0001 RS  --> C
0.0001 RS  --> D
0.0001 RS  --> E
0.0001 RS  --> F
0.0001 RS  --> G
0.0001 RS  --> H
0.0001 RS  --> I
0.0001 RS  --> K
0.0001 RS  --> L
0.0001 RS  --> M
0.0001 RS  --> N
0.0001 RS  --> P
0.0001 RS  --> Q
0.0001 RS  --> R
0.9981 RS  --> S
0.0001 RS  --> T
0.0001 RS  --> V
0.0001 RS  --> W
0.0001 RS  --> Y
#-----------------------
# RT   T               1
0.0001 RT  --> A
0.0001 RT  --> C
0.0001 RT  --> D
0.0001 RT  --> E
0.0001 RT  --> F
0.0001 RT  --> G
0.0001 RT  --> H
0.0001 RT  --> I
0.0001 RT  --> K
0.0001 RT  --> L
0.0001 RT  --> M
0.0001 RT  --> N
0.0001 RT  --> P
0.0001 RT  --> Q
0.0001 RT  --> R
0.0001 RT  --> S
0.9981 RT  --> T
0.0001 RT  --> V
0.0001 RT  --> W
0.0001 RT  --> Y
#-----------------------
# RV   V               1
0.0001 RV  --> A
0.0001 RV  --> C
0.0001 RV  --> D
0.0001 RV  --> E
```

```
0.0001 RV  --> F
0.0001 RV  --> G
0.0001 RV  --> H
0.0001 RV  --> I
0.0001 RV  --> K
0.0001 RV  --> L
0.0001 RV  --> M
0.0001 RV  --> N
0.0001 RV  --> P
0.0001 RV  --> Q
0.0001 RV  --> R
0.0001 RV  --> S
0.0001 RV  --> T
0.9981 RV  --> V
0.0001 RV  --> W
0.0001 RV  --> Y
#-----------------------
# RW   W               1
0.0001 RW  --> A
0.0001 RW  --> C
0.0001 RW  --> D
0.0001 RW  --> E
0.0001 RW  --> F
0.0001 RW  --> G
0.0001 RW  --> H
0.0001 RW  --> I
0.0001 RW  --> K
0.0001 RW  --> L
0.0001 RW  --> M
0.0001 RW  --> N
0.0001 RW  --> P
0.0001 RW  --> Q
0.0001 RW  --> R
0.0001 RW  --> S
0.0001 RW  --> T
0.0001 RW  --> V
0.9981 RW  --> W
0.0001 RW  --> Y
#-----------------------
# RY   Y               1
0.0001 RY  --> A
0.0001 RY  --> C
0.0001 RY  --> D
0.0001 RY  --> E
0.0001 RY  --> F
0.0001 RY  --> G
0.0001 RY  --> H
0.0001 RY  --> I
0.0001 RY  --> K
0.0001 RY  --> L
0.0001 RY  --> M
0.0001 RY  --> N
```

```
0.0001 RY  --> P
0.0001 RY  --> Q
0.0001 RY  --> R
0.0001 RY  --> S
0.0001 RY  --> T
0.0001 RY  --> V
0.0001 RY  --> W
0.9981 RY  --> Y
#-------------------------------------------------
# RAS  [AS]            2
0.4491 RAS --> A
0.0001 RAS --> C
0.0001 RAS --> D
0.0001 RAS --> E
0.0001 RAS --> F
0.0001 RAS --> G
0.0001 RAS --> H
0.0001 RAS --> I
0.0001 RAS --> K
0.0001 RAS --> L
0.0001 RAS --> M
0.0001 RAS --> N
0.0001 RAS --> P
0.0001 RAS --> Q
0.0001 RAS --> R
0.4491 RAS --> S
0.0001 RAS --> T
0.0001 RAS --> V
0.0001 RAS --> W
0.0001 RAS --> Y
#-----------------------
# RDN  [DN]            2
0.0001 RDN --> A
0.0001 RDN --> C
0.4491 RDN --> D
0.0001 RDN --> E
0.0001 RDN --> F
0.0001 RDN --> G
0.0001 RDN --> H
0.0001 RDN --> I
0.0001 RDN --> K
0.0001 RDN --> L
0.0001 RDN --> M
0.4491 RDN --> N
0.0001 RDN --> P
0.0001 RDN --> Q
0.0001 RDN --> R
0.0001 RDN --> S
0.0001 RDN --> T
0.0001 RDN --> V
0.0001 RDN --> W
0.0001 RDN --> Y
```

```
#-----------------------
# RDS  [DS]            2
0.0001 RDS --> A
0.0001 RDS --> C
0.4491 RDS --> D
0.0001 RDS --> E
0.0001 RDS --> F
0.0001 RDS --> G
0.0001 RDS --> H
0.0001 RDS --> I
0.0001 RDS --> K
0.0001 RDS --> L
0.0001 RDS --> M
0.0001 RDS --> N
0.0001 RDS --> P
0.0001 RDS --> Q
0.0001 RDS --> R
0.4491 RDS --> S
0.0001 RDS --> T
0.0001 RDS --> V
0.0001 RDS --> W
0.0001 RDS --> Y
#-----------------------
# RFL  [FL]            2
0.0001 RFL --> A
0.0001 RFL --> C
0.0001 RFL --> D
0.0001 RFL --> E
0.4491 RFL --> F
0.0001 RFL --> G
0.0001 RFL --> H
0.0001 RFL --> I
0.0001 RFL --> K
0.4491 RFL --> L
0.0001 RFL --> M
0.0001 RFL --> N
0.0001 RFL --> P
0.0001 RFL --> Q
0.0001 RFL --> R
0.0001 RFL --> S
0.0001 RFL --> T
0.0001 RFL --> V
0.0001 RFL --> W
0.0001 RFL --> Y
#-----------------------
# RFW  [FW]            2
0.0001 RFW --> A
0.0001 RFW --> C
0.0001 RFW --> D
0.0001 RFW --> E
0.4491 RFW --> F
0.0001 RFW --> G
```

```
0.0001 RFW --> H
0.0001 RFW --> I
0.0001 RFW --> K
0.0001 RFW --> L
0.0001 RFW --> M
0.0001 RFW --> N
0.0001 RFW --> P
0.0001 RFW --> Q
0.0001 RFW --> R
0.0001 RFW --> S
0.0001 RFW --> T
0.0001 RFW --> V
0.4491 RFW --> W
0.0001 RFW --> Y
#-----------------------
# RFY  [FY]            2
0.0001 RFY --> A
0.0001 RFY --> C
0.0001 RFY --> D
0.0001 RFY --> E
0.4491 RFY --> F
0.0001 RFY --> G
0.0001 RFY --> H
0.0001 RFY --> I
0.0001 RFY --> K
0.0001 RFY --> L
0.0001 RFY --> M
0.0001 RFY --> N
0.0001 RFY --> P
0.0001 RFY --> Q
0.0001 RFY --> R
0.0001 RFY --> S
0.0001 RFY --> T
0.0001 RFY --> V
0.0001 RFY --> W
0.4491 RFY --> Y
#-----------------------
# RIS  [IS]            2
0.0001 RIS --> A
0.0001 RIS --> C
0.0001 RIS --> D
0.0001 RIS --> E
0.0001 RIS --> F
0.0001 RIS --> G
0.0001 RIS --> H
0.4491 RIS --> I
0.0001 RIS --> K
0.0001 RIS --> L
0.0001 RIS --> M
0.0001 RIS --> N
0.0001 RIS --> P
0.0001 RIS --> Q
```

```
0.0001 RIS --> R
0.4491 RIS --> S
0.0001 RIS --> T
0.0001 RIS --> V
0.0001 RIS --> W
0.0001 RIS --> Y
#-----------------------
# RIV  [IV]            2
0.0001 RIV --> A
0.0001 RIV --> C
0.0001 RIV --> D
0.0001 RIV --> E
0.0001 RIV --> F
0.0001 RIV --> G
0.0001 RIV --> H
0.4491 RIV --> I
0.0001 RIV --> K
0.0001 RIV --> L
0.0001 RIV --> M
0.0001 RIV --> N
0.0001 RIV --> P
0.0001 RIV --> Q
0.0001 RIV --> R
0.0001 RIV --> S
0.0001 RIV --> T
0.4491 RIV --> V
0.0001 RIV --> W
0.0001 RIV --> Y
#-----------------------
# RLI  [LI]            2
0.0001 RLI --> A
0.0001 RLI --> C
0.0001 RLI --> D
0.0001 RLI --> E
0.0001 RLI --> F
0.0001 RLI --> G
0.0001 RLI --> H
0.4491 RLI --> I
0.0001 RLI --> K
0.4491 RLI --> L
0.0001 RLI --> M
0.0001 RLI --> N
0.0001 RLI --> P
0.0001 RLI --> Q
0.0001 RLI --> R
0.0001 RLI --> S
0.0001 RLI --> T
0.0001 RLI --> V
0.0001 RLI --> W
0.0001 RLI --> Y
#-----------------------
# RNK  [NK]            2
```

```
0.0001 RNK --> A
0.0001 RNK --> C
0.0001 RNK --> D
0.0001 RNK --> E
0.0001 RNK --> F
0.0001 RNK --> G
0.0001 RNK --> H
0.0001 RNK --> I
0.4491 RNK --> K
0.0001 RNK --> L
0.0001 RNK --> M
0.4491 RNK --> N
0.0001 RNK --> P
0.0001 RNK --> Q
0.0001 RNK --> R
0.0001 RNK --> S
0.0001 RNK --> T
0.0001 RNK --> V
0.0001 RNK --> W
0.0001 RNK --> Y
#-----------------------
# RQE  [QE]            2
0.0001 RQE --> A
0.0001 RQE --> C
0.0001 RQE --> D
0.4991 RQE --> E
0.0001 RQE --> F
0.0001 RQE --> G
0.0001 RQE --> H
0.0001 RQE --> I
0.0001 RQE --> K
0.0001 RQE --> L
0.0001 RQE --> M
0.0001 RQE --> N
0.0001 RQE --> P
0.4991 RQE --> Q
0.0001 RQE --> R
0.0001 RQE --> S
0.0001 RQE --> T
0.0001 RQE --> V
0.0001 RQE --> W
0.0001 RQE --> Y
#-----------------------
# RSG  [SG]            2
0.0001 RSG --> A
0.0001 RSG --> C
0.0001 RSG --> D
0.0001 RSG --> E
0.0001 RSG --> F
0.4491 RSG --> G
0.0001 RSG --> H
0.0001 RSG --> I
```

```
0.0001 RSG --> K
0.0001 RSG --> L
0.0001 RSG --> M
0.0001 RSG --> N
0.0001 RSG --> P
0.0001 RSG --> Q
0.0001 RSG --> R
0.4491 RSG --> S
0.0001 RSG --> T
0.0001 RSG --> V
0.0001 RSG --> W
0.0001 RSG --> Y
#-----------------------
# RST  [ST]            2
0.0001 RST --> A
0.0001 RST --> C
0.0001 RST --> D
0.0001 RST --> E
0.0001 RST --> F
0.0001 RST --> G
0.0001 RST --> H
0.0001 RST --> I
0.0001 RST --> K
0.0001 RST --> L
0.0001 RST --> M
0.0001 RST --> N
0.0001 RST --> P
0.0001 RST --> Q
0.0001 RST --> R
0.4491 RST --> S
0.4491 RST --> T
0.0001 RST --> V
0.0001 RST --> W
0.0001 RST --> Y
#-----------------------------------------------
# ATV       [ATV]      3
0.3328 RATV --> A
0.0001 RATV --> C
0.0001 RATV --> D
0.0001 RATV --> E
0.0001 RATV --> F
0.0001 RATV --> G
0.0001 RATV --> H
0.0001 RATV --> I
0.0001 RATV --> K
0.0001 RATV --> L
0.0001 RATV --> M
0.0001 RATV --> N
0.0001 RATV --> P
0.0001 RATV --> Q
0.0001 RATV --> R
0.0001 RATV --> S
```

```
0.3327 RATV --> T
0.3328 RATV --> V
0.0001 RATV --> W
0.0001 RATV --> Y
#-----------------------
# RDTN      [DTN]      3
0.0001 RDTN --> A
0.0001 RDTN --> C
0.3328 RDTN --> D
0.0001 RDTN --> E
0.0001 RDTN --> F
0.0001 RDTN --> G
0.0001 RDTN --> H
0.0001 RDTN --> I
0.0001 RDTN --> K
0.0001 RDTN --> L
0.0001 RDTN --> M
0.3327 RDTN --> N
0.0001 RDTN --> P
0.0001 RDTN --> Q
0.0001 RDTN --> R
0.0001 RDTN --> S
0.3328 RDTN --> T
0.0001 RDTN --> V
0.0001 RDTN --> W
0.0001 RDTN --> Y
#-----------------------
# RFIA      [FIA]      3
0.3328 RFIA --> A
0.0001 RFIA --> C
0.0001 RFIA --> D
0.0001 RFIA --> E
0.3327 RFIA --> F
0.0001 RFIA --> G
0.0001 RFIA --> H
0.3328 RFIA --> I
0.0001 RFIA --> K
0.0001 RFIA --> L
0.0001 RFIA --> M
0.0001 RFIA --> N
0.0001 RFIA --> P
0.0001 RFIA --> Q
0.0001 RFIA --> R
0.0001 RFIA --> S
0.0001 RFIA --> T
0.0001 RFIA --> V
0.0001 RFIA --> W
0.0001 RFIA --> Y
#-----------------------
# RFLI     [FLI]       3
0.0001 RFLI --> A
0.0001 RFLI --> C
```

```
0.0001 RFLI --> D
0.0001 RFLI --> E
0.3328 RFLI --> F
0.0001 RFLI --> G
0.0001 RFLI --> H
0.3327 RFLI --> I
0.0001 RFLI --> K
0.3328 RFLI --> L
0.0001 RFLI --> M
0.0001 RFLI --> N
0.0001 RFLI --> P
0.0001 RFLI --> Q
0.0001 RFLI --> R
0.0001 RFLI --> S
0.0001 RFLI --> T
0.0001 RFLI --> V
0.0001 RFLI --> W
0.0001 RFLI --> Y
#-----------------------
# RKIR      [KIR]      3
0.0001 RKIR --> A
0.0001 RKIR --> C
0.0001 RKIR --> D
0.0001 RKIR --> E
0.0001 RKIR --> F
0.0001 RKIR --> G
0.0001 RKIR --> H
0.3328 RKIR --> I
0.3327 RKIR --> K
0.0001 RKIR --> L
0.0001 RKIR --> M
0.0001 RKIR --> N
0.0001 RKIR --> P
0.0001 RKIR --> Q
0.3328 RKIR --> R
0.0001 RKIR --> S
0.0001 RKIR --> T
0.0001 RKIR --> V
0.0001 RKIR --> W
0.0001 RKIR --> Y
#-----------------------
# RLAS      [LAS]      3
0.3328 RLAS --> A
0.0001 RLAS --> C
0.0001 RLAS --> D
0.0001 RLAS --> E
0.0001 RLAS --> F
0.0001 RLAS --> G
0.0001 RLAS --> H
0.0001 RLAS --> I
0.0001 RLAS --> K
0.3327 RLAS --> L
```

```
0.0001 RLAS --> M
0.0001 RLAS --> N
0.0001 RLAS --> P
0.0001 RLAS --> Q
0.0001 RLAS --> R
0.3328 RLAS --> S
0.0001 RLAS --> T
0.0001 RLAS --> V
0.0001 RLAS --> W
0.0001 RLAS --> Y
#-----------------------
# RLIV    [LIV]        3
0.0001 RLIV --> A
0.0001 RLIV --> C
0.0001 RLIV --> D
0.0001 RLIV --> E
0.0001 RLIV --> F
0.0001 RLIV --> G
0.0001 RLIV --> H
0.3327 RLIV --> I
0.0001 RLIV --> K
0.3328 RLIV --> L
0.0001 RLIV --> M
0.0001 RLIV --> N
0.0001 RLIV --> P
0.0001 RLIV --> Q
0.0001 RLIV --> R
0.0001 RLIV --> S
0.0001 RLIV --> T
0.3328 RLIV --> V
0.0001 RLIV --> W
0.0001 RLIV --> Y
#-----------------------
# RPRQ      [PRQ]      3
0.0001 RPRQ --> A
0.0001 RPRQ --> C
0.0001 RPRQ --> D
0.0001 RPRQ --> E
0.0001 RPRQ --> F
0.0001 RPRQ --> G
0.0001 RPRQ --> H
0.0001 RPRQ --> I
0.0001 RPRQ --> K
0.0001 RPRQ --> L
0.0001 RPRQ --> M
0.0001 RPRQ --> N
0.3328 RPRQ --> P
0.3327 RPRQ --> Q
0.3328 RPRQ --> R
0.0001 RPRQ --> S
0.0001 RPRQ --> T
0.0001 RPRQ --> V
```

```
0.0001 RPRQ --> W
0.0001 RPRQ --> Y
#-----------------------
# RSNK      [SNK]      3
0.3328 RSNK --> A
0.0001 RSNK --> C
0.0001 RSNK --> D
0.0001 RSNK --> E
0.3327 RSNK --> F
0.0001 RSNK --> G
0.0001 RSNK --> H
0.3328 RSNK --> I
0.0001 RSNK --> K
0.0001 RSNK --> L
0.0001 RSNK --> M
0.0001 RSNK --> N
0.0001 RSNK --> P
0.0001 RSNK --> Q
0.0001 RSNK --> R
0.0001 RSNK --> S
0.0001 RSNK --> T
0.0001 RSNK --> V
0.0001 RSNK --> W
0.0001 RSNK --> Y
#-----------------------------------------------
# RDEQV     [DEQV]     4
0.0001 RDEQV --> A
0.0001 RDEQV --> C
0.2496 RDEQV --> D
0.2496 RDEQV --> E
0.0001 RDEQV --> F
0.0001 RDEQV --> G
0.0001 RDEQV --> H
0.0001 RDEQV --> I
0.0001 RDEQV --> K
0.0001 RDEQV --> L
0.0001 RDEQV --> M
0.0001 RDEQV --> N
0.0001 RDEQV --> P
0.2496 RDEQV --> Q
0.0001 RDEQV --> R
0.0001 RDEQV --> S
0.0001 RDEQV --> T
0.2496 RDEQV --> V
0.0001 RDEQV --> W
0.0001 RDEQV --> Y
#-----------------------
# RNLFP      [NLFP]     4
0.0001 RNLFP --> A
0.0001 RNLFP --> C
0.0001 RNLFP --> D
0.0001 RNLFP --> E
```

```
0.2496 RNLFP --> F
0.0001 RNLFP --> G
0.0001 RNLFP --> H
0.0001 RNLFP --> I
0.0001 RNLFP --> K
0.2496 RNLFP --> L
0.0001 RNLFP --> M
0.2496 RNLFP --> N
0.2496 RNLFP --> P
0.0001 RNLFP --> Q
0.0001 RNLFP --> R
0.0001 RNLFP --> S
0.0001 RNLFP --> T
0.0001 RNLFP --> V
0.0001 RNLFP --> W
0.0001 RNLFP --> Y
#-----------------------
# RSTAG      [STAG]    4
0.2496 RSTAG --> A
0.0001 RSTAG --> C
0.0001 RSTAG --> D
0.0001 RSTAG --> E
0.0001 RSTAG --> F
0.2496 RSTAG --> G
0.0001 RSTAG --> H
0.0001 RSTAG --> I
0.0001 RSTAG --> K
0.0001 RSTAG --> L
0.0001 RSTAG --> M
0.0001 RSTAG --> N
0.0001 RSTAG --> P
0.0001 RSTAG --> Q
0.0001 RSTAG --> R
0.2496 RSTAG --> S
0.2496 RSTAG --> T
0.0001 RSTAG --> V
0.0001 RSTAG --> W
0.0001 RSTAG --> Y
#-----------------------
# RYASP      [YASP]    4
0.2496 RYASP --> A
0.0001 RYASP --> C
0.0001 RYASP --> D
0.0001 RYASP --> E
0.0001 RYASP --> F
0.0001 RYASP --> G
0.0001 RYASP --> H
0.0001 RYASP --> I
0.0001 RYASP --> K
0.0001 RYASP --> L
0.0001 RYASP --> M
0.0001 RYASP --> N
```

```
0.2496 RYASP --> P
0.0001 RYASP --> Q
0.0001 RYASP --> R
0.2496 RYASP --> S
0.0001 RYASP --> T
0.0001 RYASP --> V
0.0001 RYASP --> W
0.2496 RYASP --> Y
#-------------------------------------------------
# RFTE6     [FTESAD]   6
0.1665 RFTE6 --> A
0.0001 RFTE6 --> C
0.1664 RFTE6 --> D
0.1664 RFTE6 --> E
0.1664 RFTE6 --> F
0.0001 RFTE6 --> G
0.0001 RFTE6 --> H
0.0001 RFTE6 --> I
0.0001 RFTE6 --> K
0.0001 RFTE6 --> L
0.0001 RFTE6 --> M
0.0001 RFTE6 --> N
0.0001 RFTE6 --> P
0.0001 RFTE6 --> Q
0.0001 RFTE6 --> R
0.1664 RFTE6 --> S
0.1665 RFTE6 --> T
0.0001 RFTE6 --> V
0.0001 RFTE6 --> W
0.0001 RFTE6 --> Y
#-------------------------------------------------
# RFKD7     [FKDSIPN]  7
0.0001 RFKD7 --> A
0.0001 RFKD7 --> C
0.1426 RFKD7 --> D
0.0001 RFKD7 --> E
0.1427 RFKD7 --> F
0.0001 RFKD7 --> G
0.0001 RFKD7 --> H
0.1427 RFKD7 --> I
0.1427 RFKD7 --> K
0.0001 RFKD7 --> L
0.0001 RFKD7 --> M
0.1427 RFKD7 --> N
0.1427 RFKD7 --> P
0.0001 RFKD7 --> Q
0.0001 RFKD7 --> R
0.1426 RFKD7 --> S
0.0001 RFKD7 --> T
0.0001 RFKD7 --> V
0.0001 RFKD7 --> W
0.0001 RFKD7 --> Y
```