Università degli Studi di Padova

Dipartimento di Fisica e Astronomia "Galileo Galilei"

Corso di laurea magistrale in

Fisica

Statistical inference methods in biophysics

Relatore interno: Prof. Amos Maritan

Relatore esterno: Prof. Silvio Franz (LPTMS, Paris-Sud)

Laureando: Nicola Quadri

Anno Accademico 2013/2014

Contents

Introduction

The present work lies within the field of biophysics, or, more generally, statistical physics of complex systems. Its final aim is to improve, using methods and approaches proper of this field, current knowledge about influenza virus evolution, focusing on a key-protein of the virus, hemagglutinin protein (HA), its major surface antigen. Starting from the selection of an appropriate ensemble of HA sequences retrieved from the international influenza database at NCBI, it presents and shows the application of different information theory techniques and statistical mechanics methods, in order to:

1. characterize the information content of the data sample, providing evidences that the latter is well suited to be subject to inferential procedures;

2. extract, using these procedures, information about the existence of substructures of co-evolving sites within the protein.

As will be discussed in the last section, methods and algorithms here introduced and applied in order to reach these results, can be in fact generalized to other fields and problems, different from the one here presented.

Let us outline the following chapters and their content.

First chapter. We give a brief overview on influenza A virus biology and ecology: we describe the virion structure, its fundamental proteins, their roles and the replication cycle of the virus; we discuss its presence in nature, the variety of existing subtypes and the complex phylogenetic relations between them, to end up with a review of the current knowledge about evolutionary behavior of influenza A hemagglutinin protein.

Second chapter. We present the data sample of HA protein sequences upon which are based all the subsequent analysis, we explain how and where this has been retrieved, under which choices, and what are the first standard procedures one has to apply to a sample of protein

sequences before any further step.

Third chapter. We explain the message-passing clustering algorithm used to cluster our sequences sample by similarity, the information theory framework within which the clustering outcome can be interpreted as an indicator of the information content of the sample and the positive conclusions one can draw from this analysis about how the data chosen for the present work (the HA sequences) are able to represent faithfully the actual system (the HA protein).

Fourth chapter. We present, compare and apply two different methods, called *Direct coupling analysis* and *Statistical couplings analysis*, both able, at least theoretically, to infer interactions between sites of a protein starting from empirical correlations computed on a sample of its sequences; we discuss the failure of the former method and we show the preliminary results obtained with the latter, i.e., the identification of co-evolving ensembles of sites (sectors) within the HA protein sequence.

Chapter 1

Influenza A virus: biology and ecology

In this first chapter, assuming the reader to be almost completely unaware about it, we briefly introduce the influenza A virus biology and ecology. However, the essential information needed to fully understand the biological subject of the present work and hence the following chapters are mainly the functional role of the Hemagglutinin (HA) protein, the major surface antigen of the influenza virus virion (explained in section 1.2.1), the existence in nature of many influenza A virus subtypes, indexed using their antigenic proteins HA and NA (explained in the same section) and the discussion on influenza A virus evolution (section 1.4).

1.1 A global public health challenge

As every one knows by his own experience, seasonal influenza is an acute viral infection. It is caused by a family of RNA viruses called *Orthomyxoviruses*[1].

There are three types of seasonal influenza, caused by three of the six *Orthomyxoviruses*: A, B and C, further subdivided into subtypes according to different kinds and combinations of virus surface proteins. Type C of influenza cases occur much less frequently than A and B (for this reason only the latter are included in seasonal influenza vaccines); the same is true for the B type with respect to the A one, the most common and infectious. Since in our study not only we chose to work only on type A of influenza, but also on a particular subtype of it, the

[1] From *orthos*, Greek for "straight", and *myxa*, Greek for "mucus".

H3N2, which is the most widespread influenza virus currently circulating among humans, in the following we will talk strictly about A type of influenza virus.

Seasonal influenza is characterized by high fever, cough, headache, muscle and joint pain, sore throat and runny nose. Although most people recover from these symptoms within few days without any kind of medical attention, for people at high risk influenza can cause severe illness or death. According to WHO [43] these people are children younger than two years old, adults of age 65 or older, and people of any age with frail medical conditions, e.g. chronic heart, lung, kidney, liver, blood and metabolic diseases (such as diabetes) or weakened immune systems.

Worldwide influenza A annual epidemics results in about three to five million cases of severe illness and about 250000 to half a million deaths per year, making it one of the major infectious diseases in humans [43]. Due to its impact on health, influenza is today a uniquely well-documented system of molecular evolution, even if not since a long time (see 2.1). Its entire viral gene sequence, subdivided in eight segments (each one of them encoding for specific proteins, as we will see in the next section), is now available for several thousand strains and can be freely downloaded from the NCBI database [42], [41].

Vaccination is the most effective way to prevent influenza disease or the severe outcomes from the illness, being able to avoid, among healthy adults, from 70% to 90% of the cases and to reduce, among elderly population, severe illnesses and complications by up to 60% and deaths by 80% [43]. Vaccination not only is strictly recommended for high risk individuals, but also for people who live with or take care of them. Obviously, the success of the vaccine in preventing diseases is deeply related with how the vaccine is well-matched with the influenza virus strain circulating among humans in that particular year.

In fact, seasonal influenza virus undergoes rapid evolution in order to escape human immune response. Being able to predict the following strain is the hard challenge scientists all over the world are facing every year. The WHO Global Influenza Surveillance Network (GISN), a partnership of National Influenza Center around the world, monitors the influenza viruses circulating in humans. Current strategies consist, roughly speaking, in the observation of the viral strain circulating in the south hemisphere during the winter season that precede the one in the north hemisphere, and vice versa.

However, the international database now available [41] gives us new power of insight in the genetic history of influenza virus. It contains an impressive amount of data that could be used to improve our ability to

foresee influenza virus evolution. Being able to extract useful information from it, is the aim of this work.

1.2 Biology of influenza A virus

In this section we present influenza virus structure and molecular biology, and we explain its replication cycle. Since this is simply a summary of the current knowledge of molecular biology of influenza A virus, detailed references are not given individually. The summary is mainly based on Webster and Lamb textbooks and fundamental review articles [31], [32], [34] and [33]; for replication cycle, the short but clear review by Samji [35] has been very useful.

1.2.1 Components of the virion

Influenza A viruses are enveloped single-stranded RNA viruses with a pleomorphic[2] appearance (that after isolation may be spherical) and an average diameter of 120 nm. The virion consist of a host-derived lipid bilayer envelope — in which are embedded the glycoproteins HA and NA, and the matrix protein M2 — , an inner shell of matrix protein M1, and, at the center of the virion, the nucleocapsids of the viral genome[3].

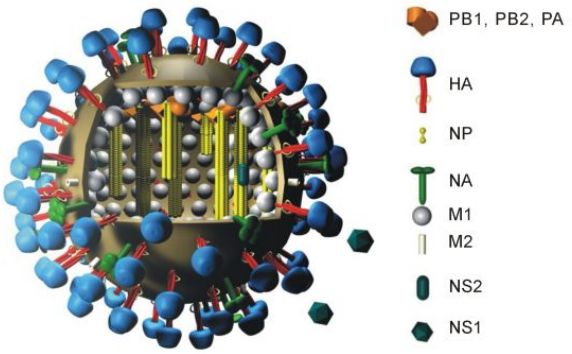

Figure 1.1: 3D representation of influenza A virus virion [38].

[2]Pleomorphism, in microbiology, is the ability of some bacteria or virus to alter their shape or size in response to environmental conditions.

[3]HA, NA, M2 and M1 proteins, together with the other six influenza A virus proteins (produced during virus replication cycle but not present in the full-formed virion), are encoded in the viral RNA and described in detail in the next pages.

The genome of influenza A virus consists of eight separate segments of RNA with negative polarity (i.e. complementary to mRNA sense). To be infectious, a virus must contains each of these segments, that codes the viral proteins:

1. polymerase B2 protein (PB2), segment 1;
2. polymerase B1 protein (PB1), segment 2;
3. polymerase A protein (PA), segment 3;
4. hemagglutinin (HA), segment 4;
5. nucleoprotein (NP), segment 5;
6. neuraminidase (NA), segment 6;
7. matrix proteins (M1 and M2), segment 7;
8. non-structural proteins (NS1 and NS2), segment 8.

Let us give a brief overview on these proteins and their structural functions.

Polymerase proteins

Proteins PB2, PB1 and PA, form the active RNA-RNA polymerase, which is responsible for replication and transcription of the genome. In particular: PB2 is known to work during initiation of viral mRNA transcription as the protein which recognizes and binds structures of host cell mRNAs to use them as viral mRNA transcription primers[4]; PB1 is believed to be responsible for elongation of viral mRNA; while PA function is less clear, but there are evidences suggesting a role as protein kinase, i.e., a protein that modifies other proteins by chemically adding phosphate groups to them, or as a helix-unwinding protein. All these proteins can be found in the nucleus of the infected cell.

Hemagglutinin

The HA protein is an integral membrane protein and the major surface antigen of the influenza virus virion. It spans the lipid membrane so that the major part, which contains at least 5 antigenic domains, is presented at the outer surface. It is responsible for binding the virion

[4]RNA and DNA polymerases can only add new nucleotides to an existing strand of nucleic acid, called primer.

to host cell receptors and for fusion between the virion envelope and the host cell membrane, followed by penetration of the interior of the virus particle into the host cell. Since it is further split into two subunits during virus replication cycle, HA protein is usually divided in two different domain: HA1 and HA2. The antigenic sites are placed on the head of the molecule (HA1 domain), while the feet are embedded in the lipid layer (HA2 domain). The body of the HA molecule contains the stalk region and the fusiogenic domain (consisting of both HA1 and HA2 domain sections), which is the one needed for membrane fusion.

So far 16 subtypes of HA protein (H1 to H16) have been found in nature, which differ by at least 30% in the amino acids sequence of HA1 domain. These subtypes are used to classify, together with NA subtypes (see in the following), subtypes of influenza A virus, as, for example, the one subject of our study: H3N2.

Nucleoprotein

NP plays a central role for virus infectivity, because it binds to and encapsidates viral RNA, forming the so-called viral nucleocapsids. NP is believed to be involved also in the switching of viral RNA polymerase activity from mRNA synthesis to cRNA synthesis. NP is abundantly synthesized in infected cells and is the second most abundant protein in the influenza virus virion.

Neuraminidase

Like HA, neuraminidase NA is a glycoprotein, which is also found as projections of tetrameric structure on the surface of the virus. It works as an enzyme, cleaving sialic acid from the HA molecule, from other NA molecules and from glycoproteins and glycolipids on the cell surface. Thus, it is fundamental to free virus particles from host cell receptors, to permit progeny virions to escape from the cell in which they have been produced, facilitating virus spread.

NA is the second major surface antigen of the virion, and for this reason, as HA protein, is highly mutable with variant selection partly in response to host immune pressure. Nine subtypes of NA (N1 to N9) have been identified in nature; as already said, based on the antigenicity of NA and HA glycoproteins, influenza A viruses are further subdivided into sixteen H (from H1 to H16) and nine N (N1 to N9) subtypes.

Matrix proteins

M1 protein is the most abundant protein in the influenza virus virion (and, in the infected cell, is present in both cytoplasm and nucleus). The protein forms a shell surrounding the virion nucleocapsids, underneath the virion envelope. It has no known enzymatic activity.

M2 protein, instead, is believed to act as a ion channel to control the pH of the particle, especially during HA synthesis and virus uncoating (the process that releases viral genome in the infected cell).

Non-structural proteins

The NS1 and NS2 proteins, particularly NS1, are abundant in the infected cell (nucleus and cytoplasm) but are not incorporated into progeny virions. These proteins appear to have a regulatory function to promote the synthesis of viral components in the infected cell. This function, however, has not been fully defined.

1.2.2 Replication cycle

Influenza virus has evolved a number of mechanisms that enable it to invade host cells and subvert the host cell machinery for its own purpose, that is the production of more virus. The ensemble of these mechanisms constitutes the virus replication cycle, a complex process that can be divided into three main stages:

1. entry of the virus into the host cell;
2. transcription and replication of the viral genome;
3. formation of progeny viral particles (and desertion of the host cell).

Entry of the virus

The replication process starts when HA binds to an host cell via interaction between the receptor-binding site of HA and the terminal sialic acid of the cell surface receptor glycoprotein or glycolipid. Since sialic acid of the needed type (with carbohydrates linkage) are present on several cells of the organism, multiple cell types may be infected. Following binding, the attached virion is endocytosed[5] by the cell. The low pH of the endocytotic vesicle start a conformational change in HA which

[5]Endocytosis is an energy-using process by which cells absorb molecules and particles from outside the vesicular membrane.

facilitate insertion of the hydrophobic free amino terminus of HA2 into the vesicular membrane, initiating fusion of the viral and vesicular membranes. The acidic environment of the endosome is not only important for inducing the conformation in HA and, thus, fusion of the viral and endosomal membranes, but also opens up the M2 ion channel. Opening the M2 ion channels acidifies the viral core. This acidic environment in the virion releases the nucleocapsids from M1 such that they are free to enter the host cell cytoplasm. Uncoating of the virus is completed within 20-30 minutes after virus attachment.

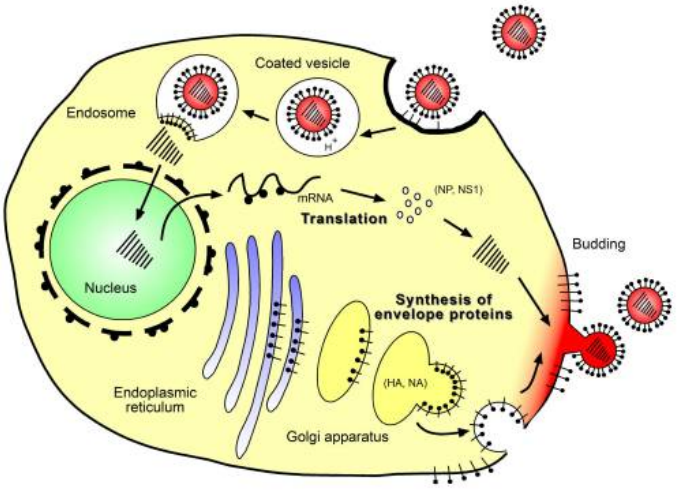

Figure 1.2: Schematic representation of influenza A virus replication cycle. Picture taken from [38].

Transcription and replication of viral RNA

The nucleocapsids of the parent virus migrate into the host cell nucleus, where polymerase complexes cleaves and elongates viral RNA, and starts primary transcription of mRNA (during the first stage the production of viral RNA is limited by the NP in favor of viral mRNA; translation of host mRNAs is blocked). mRNA is then transported to the cytoplasm, where viral proteins are synthesized at the ribosome. In the early stages of infection predominantly synthesized viral proteins are NP and NS1; later the principal translation products are M1, HA, and NA proteins. Newly synthesized NP and NS1 migrate to the nucleus, where the novel viral RNA is encapsidated by NP protein and function

as templates for secondary transcription of viral mRNAs, while HA, NA and M2 proteins are transported to the cell surface, where they integrate into the cell membrane.

Formation of progeny viral particles

Once the viral RNA has been replicated using the host cell replication machinery, the virus is ready to form progeny viral particles and leave the cell (and go on to infect neighboring cells). Since influenza is an enveloped virus, it uses the host cell plasma membrane to form the viral particles. Viral proteins normally found within the viral lipid bilayer, like HA, NA, and M2, must reach high enough concentration in the host cell plasma membrane. When the required concentration is reached, a viral core of nucleocapsids encased in a shell of M1 proteins aggregate and condense to produce the viral particle. The particle buds outward through the cell membrane, enclosing itself within a bubble of membrane as its own envelope.

The time from entry to production of new viruses is on average six hours.

1.3 Ecology of influenza A virus

1.3.1 Influenza A viruses in nature

Influenza A viruses infect a large variety of animals — including humans, pigs, horses, sea mammals and birds — divided, as shown in figure 1.3, into five different host groups, based on phylogenetic analysis of virus proteins (in particular the NP protein) from a large sample of influenza viruses.

The occurrence of interspecies transmission between these hosts group strongly depends on the species involved. Understanding it is of primarily importance, because is strictly related to pandemic disease outbreaks in humans. Although has been demonstrated only between pigs and humans, there is extensive evidence for transmission between wild ducks and other species [31]. For these reasons and because here we only want to give the reader a general idea of the complex landscape of influenza A virus hosts and ecology, we will talk mainly about influenza A virus in wild ducks (aquatic birds) and pigs, that are, from a human point of view, the most interesting and dangerous hosts in nature.

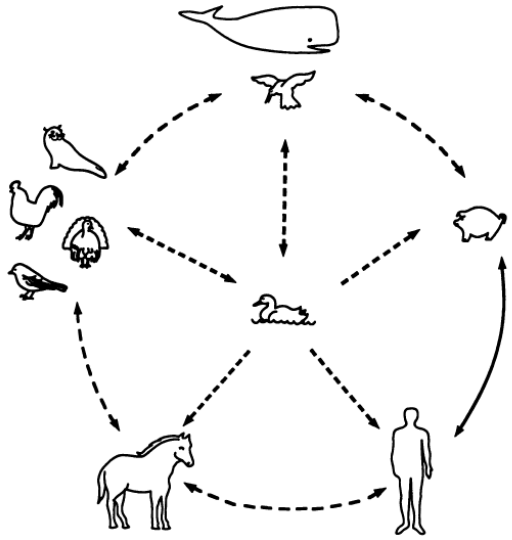

Figure 1.3: Wild aquatic birds are currently considered the primordial reservoir of all influenza viruses (the animals in picture are grouped in 5 hosts family by phylogenetic analysis of NP protein). Picture is taken from [31].

Influenza viruses in birds

Phylogenetic studies have shown, in the last twenty years, that wild aquatic birds may represent a primordial reservoir for all influenza viruses circulating in avian and mammalian species [31]. In fact all of the different subtypes of influenza A virus (from H1 to H16 and from N1 to N9) are perpetuated in aquatic birds [36]. This is a consequence of the fact that infection caused by most strains of influenza virus are completely asymptomatic in ducks and in many other avian species.

The avirulent nature of influenza infection in these animals may be a result of an adaptation process occurred over many centuries. If this is true, wild birds not only occupy a unique and very important position in the history of influenza viruses, but they also constitute the principal reservoir that ensures the perpetuation of the virus and a "laboratory" for its evolution.

Water plays a fundamental role in the spread of influenza virus through aquatic wild birds. In these animals, the virus usually replicates within the cells of the intestinal tract and it is expelled in high concentration in the feces. As a consequence, influenza virus can be easily isolated in lakes water, from where it is able to infect other birds.

This suggests that the water supply of aquatic wild birds represents the most efficient way to transmit influenza viruses within these species (for example, about 20%-30% of juvenile birds shows influenza virus infection when birds congregate in Canadian lakes before migration every year [36]). Direct transmission by feces also provide a way for ducks, as they migrate through an area, to spread the influenza viruses to other domestic and feral birds.

Recent phylogenetic analysis indicate also that avian influenza virus strains predominant in Eurasia and Australia could be distinguished genetically from those in North America, and this is presumably due to the confinement of birds to distinct flyways in each hemisphere. As expected, the evolution of the virus can be influenced by the interposition of physical barriers able to avoid intermixing between hosts.

Influenza virus have also been isolated sporadically from shorebirds — including gulls, terns, shearwaters, guillemots, sandpipers — and from domestic poultry, like chickens and turkeys. The predominant subtypes of influenza circulating in these birds are different from those of wild ducks. However, all avian viruses appear to originate from aquatic birds, because they have no other known reservoir. The fact that also all of the current mammalian influenza A virus strains appear to be derived from aquatic birds, strongly suggests that these animals constitute an influenza virus gene pool of worldwide extent [31], available for the future generation of influenza viruses in mammalian species.

Influenza viruses in pigs

There are only two subtypes of influenza viruses isolated from pigs: classic swine and avian-like H1N1, and human-like and avian-like H3N2 viruses.

The first evidence of swine influenza was observed in 1918, during the H1N1 influenza pandemic also known as Spanish flu[6]. In only two years the virus infected more than 500 million people around the world and killed from 50 to 100 millions of them, i.e. three to five percent of the world population at that time [40].

The origin of the the pandemic virus strain, even today, is not completely understood. The leading theory is that the virus strain originated at Fort Riley, in Kansas, through genetic drift and antigenic shift in poultry and swine. However, a recent reconstruction based on ini-

[6]The nickname is due to the neutrality of Spain in the first world war: it was the only country in which scientific publications about the mortality and illness were not silenced by wartime censors in order to maintain high morale in soldiers.

tial data suggests that the virus jumped directly from birds to humans, without traveling through swine, that caught the disease from humans.

Another pandemic of H1N1 influenza virus (although in a different strain version, consequence of a reassortment of bird, swine and human previous strains) happened in 2009, but it was not even comparable with the Spanish flu for incidence of the infections, virulence and mortality [43].

In pigs, in which influenza viruses H1N1 are primarily enzootic and H3N2 are either enzootic or periodically introduced from human, reassortants possessing the H1N2 virus have been also detected in Japan, proving that genetic reassortment can occur between influenza A viruses in pigs. Serological studies on slaughterhouse workers also demonstrates that swine influenza viruses can be transmitted to humans with high frequency (20% of workers had antibodies to swine influenza viruses [31]). As already said, this is the only interspecies transmission ever proved, although there are many factors suggesting other interspecies transmissions to be happened.

Influenza viruses in horses and other species

The first isolation of influenza virus in horses was made in 1956. Since then, two different subtypes of influenza virus has been detected in horses: H3N8 and H7N7.

Phylogenetic studies indicate that the common equine H3 HA gene (part of the H3N8 virus) was, once again, introduced into horses from birds long ago [31], [39]. These studies, on the other hand, show that, compared to frequent interspecies transmission involving pigs, exchange of influenza virus genes between horses and other species is very limited. Taking into account that the H7N7 virus is now thought to be extinct [37], horses may be an isolated and so a dead-end reservoir for influenza A viruses.

Influenza viruses isolated in seals (H7N7 and H4N5), whales (H13N2, H13N9 and H1N3) and mink (H10N4) were all recognized to be of avian origin, as shown by genetic analysis, as competitive RNA-RNA hybridization [31]. In particular the H7N7 virus in seals provide one of the strongest evidences that a strain deriving all of its genes from an avian influenza virus can produce severe disease in mammalians, enforcing the hypothesis that at least some human influenza viruses could be derived directly from avian ones and demonstrating that interspecies transmission is possible and occurs mainly from birds to other species.

1.3.2 Why these hosts and these viruses?

As just seen, influenza viruses in nature exhibit some sort of host range restriction: subtypes of the virus more common in some hosts are relatively rare in others, and vice versa. Specific reasons for these restrictions are not yet understood. Although it is possible that any of the influenza virus gene products has its own role in this restriction, some proteins/genes are probably more determinant than others, like HA and NA, the first and the second major surface antigen of the virion: HA has a a primary role in host cell recognition and attachment (penetration inside the cell by membrane fusion), so that especially for receptor-binding sites host specificity is of primary importance for the success of the infection; changes in NA protein can also alter virulence properties of the virus and its ability to form viral plaques (at least in cell culture).

In any case, beyond these general considerations, we do not know which unrecognized features are promoted by specific choices of NA and HA gene products and why these features make the virus well adapted for growth in some animals while not in others. Any kind of prediction about the virulence of a novel reassortant virus in an alternate host is very difficult (if not impossible). Although no simple linkage between NA and HA subtypes is known, it is interesting to notice that, compared to the total number of possible HA/NA subtypes combinations (126), only few of them are much more common than all the other ones, and 45 of them have not even been found in nature [31].

Host range restriction (lack of infectivity of a virus in new hosts) combined with isolation of host species (caused by different ecologies) allow independent host-specific evolution of virus strains and separation of the virus gene pool into host-specific ones. This subdivision of host populations provides great heterogeneity to the virus population and enhances the maintenance of a large number of virus subtypes.

1.4 Evolution of influenza A virus

In the previous section we have tried to give the reader a schematic idea about the ecology of influenza viruses in different hosts. Despite some observations about subtypes genes, we have mainly talked about influenza virus as a single evolving unit. However, each virus gene may evolve differently from others, because of different selective pressures and specific evolutionary constraints: while surface protein genes, like HA and NA, are subject to strong selective pressure by neutralizing antibodies of host immune system, genes coding for internal proteins (M1,

NP) undergo a slower evolution process, where instead of the compulsive mutations present in the former proteins (necessary to escape to host immune response) there may be a long term host-specific adaptive evolution. For this reason, talking about phylogenetic history of influenza A viruses in general does not make much sense and it is more meaningful to analyze separately the evolution of influenza virus genes. However, this is not worth to our aim, since we will deal in the following only with the 4th segment of the viral genome, the one encoding for hemagglutinin. Furthermore, being HA the first surface antigen of influenza A virus, evolution of HA gene is highly representative of the evolution of the whole virus and for this reason has been subject of many studies [26]. Hence, before moving on to the next chapters and entering the physical core of the current work, we want to close this biological section saying few important things about specific evolutionary features of hemagglutinin protein gene, on which is based our study.

1.4.1 HA gene phylogeny and evolutionary features

As said before, being the primary surface antigen of influenza virus and because of immune selection pressure, HA protein is expected to evolve more rapidly and to be replaced by reassortment more frequently than the other proteins: viruses with new genes for HA protein have a selective advantage over the parent virus to which the host has had already antigenic exposure. In fact every one or two years, new epidemic strains of influenza A arise by introduction of selected point mutations within the surface proteins, especially hemagglutinin [38]. These usually small and permanent mutations in the antigenicity of influenza A viruses are called *antigenic drift*. For human influenza viruses in particular, the H3 HA protein (which is the subject of our study) evolves much more rapidly than the internal ones (PB1, PB2, PA, NP and M1): silent mutations[7] in the former are about 57% of the total, while in internal proteins vary from 81 to 96% [31]. If the new viruses emerged from mutations in HA protein gene are sufficiently infectious, they can cause pandemics and replace the previous strains. For this reason, surface protein genes are not expected to have a long evolutionary history within hosts (like humans) having high immune selection pressure.

An extensively studied and evident feature of the evolution of HA

[7] Silent mutations (sometimes also called synonymous mutations) are nucleotide mutations that leave unchanged the correspondent amino acids. This is possible because of genetic code degeneracy: there are $4^3 = 64$ nucleotide triplets combination and only 20 amino acids.

gene is its punctuated pattern: there are periods of relative stasis (called antigenic clusters), separated by cluster transitions, which occur every few years and produce most of the antigenic adaptation [30], [26]. Clustering has also been observed in temporal distribution of amino acids fixation in HA protein gene [29], [26]. The explanation proposed for this particular evolution pattern is not unique: some authors has described it by a model of episodic evolution, in which antigenic clusters correspond to periods of neutral evolution and positive selection is restricted to cluster transitions [28]; others, as Lässig and Strelkowa [26], explain it using the so-called *clonal interference*, a mode of evolution in which high supply of beneficial mutations generates competition between coexisting clones, i.e., simultaneous competitive strains, so that also during antigenic clusters there is positive selection: although many beneficial changes reach substantial frequencies, only a fraction of them are fixed. A visual representation of the differences between the two models is shown in figure 1.4.

In order to demonstrate the plausibility of clonal interference evolution, in [26] many other features of HA evolution are also shown and analyzed using a sample of 2033 HA1 sequences of influenza A H3N2. They proved that antigenic sites are subject to a higher rate of non-synonymous beneficial mutations: 56% of the antigenic amino acids substitutions are strongly beneficial mutations and 44% are neutral, against 70% of mutations under *negative* selection in non-antigenic sites. They extract, using phylogenetic trees, that lifetime to fixation of a beneficial mutation is on average 2.9 years, so that the population of virus strains contains at least 3 simultaneous beneficial mutations on average (competition between strains under positive selection — in order to be the one to fix in the population — represent exactly the clonal interference evolution).

One of the crucial points of their discussion is the measure of genetic association inside HA1 domain. Genetic association is the codependency between mutations in couples of sites of a given genetic sequence. High genetic association means that, for example, mutation in one of the two sites is strictly related to mutation (or non-mutation) in the other one. If this happens, selection acts on genotypes level and not on individual mutations level, which is a demonstration of the pure asexual reproduction of influenza A virus and a well known prerequisite for clonal interference [27]. In [26] they define genetic association using normalized correlation in double-sites mutation frequencies, i.e. the difference between double-sites mutation frequency and the product of single-site

mutation ones

$$C_{ij} = \frac{f_{ij} - f_i f_j}{N_{ij}}, \tag{1.1}$$

where i and j run over sequence sites, f_i and f_{ij} are single and double site mutation frequencies and N_{ij} is a normalizing factor. In this way, $C_{ij} = 0$ means statistical independence and $C_{ij} = 1$ complete genetic association (and they found, for HA1 domain of H3N2 influenza virus, a mean correlation of $\bar{C} = 0.96$).

The problem in using correlations, however, is well known when dealing with a protein gene finite sample of sequences [3], [4], [1]. Correlation between two sequence sites mutations may arise from direct contribution, i.e., effective codependency between the two amino acids in these sites (consequence of some kind of physical interaction), as well as from indirect ones, i.e., codependency mediated by interactions with others amino acids, that constitute some sort of connection link between the first two. So far, even if the amino acids in two sites of the sequence are not related by any kind of functional bond, correlation between the two sites may arise from a web of even small interactions with surrounding sites. Traditional correlations are thus unable to distinguish between direct and indirect contributions and hence are not very informative about the real interaction between sequence sites, giving only a rough estimation of them (used in HA1 domain for example, as stated in [26], correlation analysis lead to the conclusion that every site is deeply interdependent with every one else).

The aim of the work presented in the following chapters is to disentangle these correlations (calculated for HA entire gene) in direct and indirect, relevant and irrelevant ones, being the former strong constraints to arbitrary mutations — in order to obtain positive selected ones — and hence crucial in conditioning the direction of influenza virus future evolution.

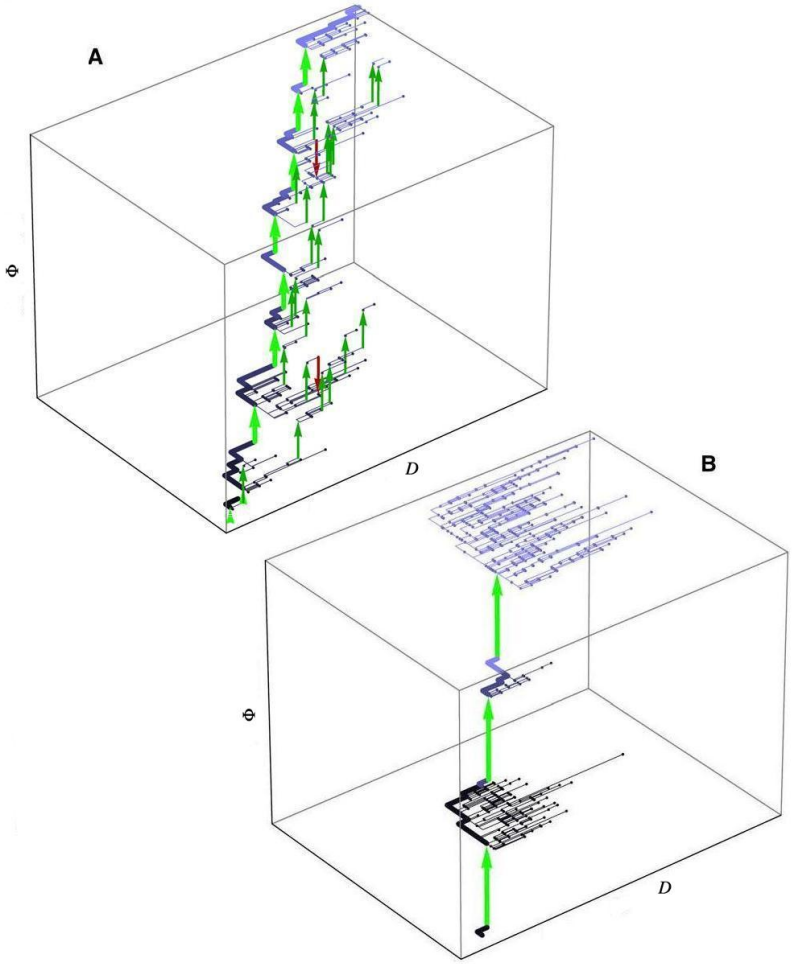

Figure 1.4: Representation of clonal interference evolution (A) and episodic selective sweeps evolution (B). Mutations are mapped on individual branches of the tree, all fixed changes appear on the trunk of the tree (thick line). The horizontal coordinate D counts the number of mutations from the root to its strain sequence, and the vertical coordinate Φ represent the sum of their selection coefficients. Upward (green) and downward (red) arrows indicate individual branches under positive and negative selection, respectively. In clonal interference mode (A), high supply of beneficial mutations generates competition between coexisting clones: many beneficial changes reach substantial frequencies within the population, but only a fraction of them are fixed (thick green arrows on the trunk), while others are eventually outcompeted (thin green arrows off the trunk). Besides, in episodic swipe mode (B), low supply of beneficial mutations generates selective sweeps interspersed with extended periods of neutral evolution (horizontal branches); all beneficial mutations reaching a substantial frequencies in the population are fixed (all green arrows are on the trunk). Picture taken from [26].

Chapter 2

Dataset

In this brief chapter, we introduce the dataset used for our study, consisting of several thousands of HA protein sequences. We start by presenting the database established at National Center for Biotechnology Information, from which our sequences have been extracted; we then discuss the guidelines followed to select the sequences, the adjustments needed to prepare them to further analysis (alignment and randomization of ambiguous letters) and the known systematic biases affecting our data sample.

2.1 Influenza Virus Resource at NCBI

As we already stressed in the first chapter, influenza disease is one of the major infectious diseases in humans, causing every year hundreds of thousands of deaths worldwide. Understanding its evolution and its molecular biology, and so conceiving new antiviral drugs and vaccines, represents an unavoidable challenge for both public health and science. To address this challenge, researchers must have free access to viral sequences in a timely fashion and need to use a unique, organized and formalized platform, where data can be exchanged and results easy controlled and compared. However, in contrast to these necessities, the number of influenza virus sequences in public databases has been historically far less than those of some well-studied viruses, such as human immunodeficiency virus, and the number of complete influenza virus genomes has been even smaller [41].

For this reason in 2004, the National Institute of Allergy and Infectious Diseases (NIAID) launched the Influenza Genome Sequencing Project, which aims to rapidly sequence influenza viruses from samples collected all over the world. Viral sequences were indexed and cataloged

at the National Center for Biotechnology Information (NCBI) and then deposited in GenBank. In just over 2 years after the beginning of the project, more than 2000 complete genomes of influenza viruses (A and B) were stored in the database [41]. To help the research community to make full use of the wealth of information from such a large amount of data, which will be increasing continuously, the Influenza Virus Resource was created at NCBI in 2004 [42], a simple and clear web interface where users can make queries, find complete genome sets and download sequences; database also offers some sequence analysis tools completely integrated, such as multiple-sequence alignment and clustering of protein sequences. The NCBI Influenza Virus Sequence Database contains nucleotide sequences of all influenza viruses in GenBank databases, as well as protein sequences and their encoding regions derived from nucleotide sequences.

2.2 HA (H3N2) sequences dataset

Our study is based on 3297 amino acids sequences of the 4th segment of influenza A H3N2 virus subtype. This segment, as already explained, codes for hemagglutinin protein, the major antigen of influenza virion. Sequences used are available at Influenza Virus Resource of NCBI [41]: since NCBI database classification system assigns to every sequence stored a unique access code, one can easily find the exact dataset using our access codes list.

The chosen sequences respect some simple accuracy requirements. First of all, we include in our dataset only sequences which contain the full HA coding region. Complete sequences, apart from being more informative, facilitate the alignment procedure and reduce alignment biases. We also used only sequences with known location and year of observation[1]. Lab strains and marked egg isolates are excluded.

The effective number of sequences obtained using these constraints is 6573. However, many of these sequences appear identically with some multiplicity, so that only 3297 of them are independent one to another.

The time span covered by our sequences is 1968 - 2013.

[1]These information are synthesized in a string of description associated to every sequence.

2.3 MSA construction

The sequences of H3 HA protein downloaded form NCBI are in fact nothing more than strings of letters. Every letter stand for a different amino acid, following the standard IUPAC amino acids alphabet (table 2.1).

A typical sequence of HA (H3) protein is:

```
MKTTIALSCILCSILAQKLPGNGDSTATLCLGHHAVPNGTLVKTITDDQIEVTNATELV
HSSSTGRICNSPHQILDGENCTLIDALLGDPNCDGFQNKEWDLFVERSTAYSNCYPYDV
PDYASLRSLVASSGTLEFTKEDFNWIGVTQGGTSNACKRGSDKSFFSRLNWLYQLSHKY
PALNVTMPNNDKFDKLYIWGVHHPSTDRDQISLYAQASGRVIVSTKGKQQTVIPNIGYR
PWVRGVSSIISIYWTIVKPGDVLLINSTGNLIAPRGYFKIRSGESSIMRSDAPIDNCNS
ECITPNGSIPNDKPFQNVNRITYGACPRYVKQNTLKLATGMRNIPEKQTRGIFGAIAGF
IENGWEGMVDGWYGFRHKNSEGTGQAADLKSTQAAINQITGKLNRVIKKTNEKFHQIEK
EFSEVEGRIQDLEKYVEDTKIDLWSYNAELLVALENQHTIDLTDSEMNKLFERTRKQLR
ENAEDMGNGCFKIYHKCDNACIESIRNGTYDHDVYRDEALNNRFQIKSVELKSGYKDWI
LWISFATSCFLICVVLLGFIVWACQKGNIRCNICI
```

The first thing to do is to perform a multiple sequence alignment (MSA), i.e. aligning the 3297 sequences one to another.

2.3.1 Aligning the sequences

In bioinformatics, a multiple sequence alignment is a way of rearranging the sequences of DNA, RNA, or protein to recognize regions of similarity that can be slightly shifted in different sequences as a consequence of evolution and experimental issues. Aligned sequences of nucleotide or amino acid residues are typically represented as rows within a matrix.

During the alignment procedure gaps are inserted between residues so that identical or similar characters are aligned in successive columns. Inserting gaps between residues has a penalty cost, so that alignment algorithms work to optimize the alignment of the sequences, but keeping low the sum of the penalty costs corresponding to gaps insertion.

Multiple sequence alignment is the starting point for phylogenetic analysis, but also, as in our case, for techniques of functionally important sites identification, such as binding sites, active sites, or sites corresponding to other key functions.

Although the alignment procedure between two (or at least three) very short and similar sequences can be done, with some patience, by hand, aligning thousand of sequences needs obviously a computational

Letters	Amino acids
A	Alanine
C	Cysteine
D	Aspartic Acid
E	Glutamic Acid
F	Phenylalanine
G	Glycine
H	Histidine
I	Isoleucine
L	Leucine
M	Methionine
N	Asparagine
P	Proline
Q	Glutamine
T	Threonine
R	Arginine
Y	Tyrosine
S	Serine
V	Valine
W	Tryptophan
K	Lysine
B	D or N
J	L or I
Z	Q or E
X	Unknown

Table 2.1: IUPAC code letters and amino acids. The first 20 letters code for the 20 amino acids; B, J and Z are "ambiguous letters", used when is not known which of the two associated amino acids is actually present in that particular sequence site. X stand for complete ignorance: a site with a X can correspond to any one of the 20 amino acids. Ambiguities or complete ignorance about sites occupation are consequences of errors in the sequence isolation procedure.

algorithm. Since sequence alignment is a key procedure for many biological and financial data analysis, there are many programs designed to do it, using different strategies and having different accuracy, computational velocity and stability.

Multiple alignment of our sequences has been done using MUSCLE algorithm [22]. MUSCLE (acronym of MUltiple Sequence Comparison by Log-Expectation) is a public domain, multiple sequence alignment

software for protein and nucleotide sequences, cited by more than 10000 papers. It's often used as a replacement for CLUSTALW, since it typically gives comparable sequence alignments but is also significantly faster, especially for larger alignments, according to published benchmark tests [22], [23]. The reason for its computational efficiency lies in the particular way the algorithm calculate distances between pair of sequences, the first step in any alignment algorithm. While other softwares, as the most common CUSTALW, perform a first alignment between any pair of sequences, MUSCLE counts the number of short sub-sequences (known as "k-mers", "k-tuples" or words) that two sequences have in common, without align them.

Since our sequences of HA gene are complete and also very similar one to another, alignment procedure generates very few gaps. The length of the resulting aligned sequences is 566 amino acids, i.e., 1701 nucleotide bases (stop codons[2] included). The alignment option used were the default-optimized ones.

2.3.2 Randomizing ambiguous IUPAC letters

Our sequences contain only one of the three possible ambiguous letters, B, standing for ambiguity between Aspargine (N) and Aspartic Acid (D). Since we have already to deal with a complex 21-alphabet MSA (21 amino acids letters and one, X, for both gap and unknown sites), we decided to randomize B letters, giving equal probability $1/2$ and $1/2$ to have, instead of B, D or N. This procedure does not change significantly the starting dataset: over 3297 sequences, each one of 566 letters, B appears only 67 times.

2.4 Known systematic biases

Before closing this chapter, let us make some remarks upon systematic biases affecting this kind of dataset. The analysis described in the following start from the assumption that sequences just presented are randomly sampled from influenza virus population and, as we will see, independent one from another. However, the available influenza sequences are not randomly sampled, which would be ideal for any kind of genetic analysis, nor independent.

The first assumption is weakened by two intrinsic database biases: yearly variation in sampling depth, because far fewer strains are available

[2]Stop codons (also called termination codons) are nucleotide triplets that signal the end of a protein translation. To these triplets is not assigned any amino acid.

for earlier years than for later years, and regional variation in sampling depth, since sequence projects — as the New York sequence project — lead to an over-representation of some geographical areas (US above all).

The second assumption is violated by the simple fact that these sequences are strongly related by a common phylogenetic history, by the presence of multiple-strain sequencing and because sequences are not sampled independently during evolution, but through a branching process, which introduces a sampling bias [1].

There are different strategies to address these biases and minimize their effects on the analysis performed in the following; we will present them time by time.

Chapter 3

Clustering and sampling regime

As already anticipated in 1.4.1, this work aim to disentangle correlations between HA protein sites in relevant and irrelevant ones, and, as we will soon see, in order to achieve this task one has to solve a statistical inference problem.

The success of this procedure lies in the informativeness of the sample on which we are working: does it contain enough statistics to make inference about the system? Following the guidelines of a recent work by Marsili et al. [7], here we try to answer that question. Since, as we will demonstrate in the following, information contained by a sample on the system behavior can be quantified by the entropy of the frequency with which different states occur [7], we need first of all to cluster our sequences by similarity.

In this chapter we present the algorithm used to cluster our MSA (the program performing the algorithm has been written in C++ and can be found in Appendix A), we then briefly explain the theoretical framework used to understand clustering results and the conclusion drawn from them about the information content of our sample.

3.1 Affinity propagation

3.1.1 Introduction

Clustering procedure consists in finding in a given dataset a subset of representative exemplars, such that the sum of a customary defined distance between data points and their nearest exemplar is small. Clusters are then in 1 to 1 correspondence with exemplars, since they are

defined as the ensembles of data points close to each one of them.

To cluster our sequences we used Affinity Propagation, a message passing algorithm developed by Frey and Dueck in 2007 [13].

Message-passing algorithms operate exchanging "messages" between the edges of a graph, and updating them recursively through local computations done at the vertices. They rely on belief propagation theory, a very powerful framework developed independently in several different contexts, such as statistical physics, coding theory and artificial intelligence (although with different names) [16], [19].

Affinity Propagation uses this technique to efficiently and rapidly cluster a starting dataset in view of data similarity, avoiding the typical problems that affect usual clustering algorithms. The most popular ones, in fact, begin with an initial set of randomly chosen exemplars and iteratively refine this set so as to decrease the sum of the distances [21]. However, this approach is quite sensitive to the initial exemplars choice, so that usually one has to rerun the algorithms many times with different initializations in an attempt to find a good result. It is clear that the sustainability of this procedure relies on the dimension of the dataset and in the number of clusters one is interested to find.

By contrast, Affinity Propagation simultaneously considers all data points as potential exemplars. By viewing each data point as a node on a graph, it recursively transmits real-valued messages along its edges until a good set of exemplars and the corresponding clusters emerges. This procedure finds clusters with higher accuracy than other methods and in less than one-hundredth the time [13].

3.1.2 About algorithm derivation

Since complete algorithm derivation can be found in the supplementary information of reference [13], here we just give the fundamental conceptual hints, without reporting all the (long) calculations.

The derivation consists in minimizing a score function depending on the exemplars selection. Calling $\boldsymbol{c} = (c_1, \ldots, c_N)$ the (unknown) exemplars to which the N points belongs, the algorithm search for configuration $\boldsymbol{c}$ that minimizes the score function

$$E(c_1, \ldots, c_N) = -\sum_{i=1}^{N} s(i, c_i), \tag{3.1}$$

where $s(i, c_i)$ is the similarity between a data point i and its exemplar c_i (the reason for this choice is self-evident).

However, not all the possible configuration $\boldsymbol{c}$ are allowed, since if

for some i $c_i = j$ then it must be that $c_j = j$. For this reason, instead of minimizing (3.1), is simpler to maximize the so-called net similarity S, defined as the opposite of the score function (3.1), plus a constraint potential enforcing valid configurations:

$$S(\boldsymbol{c}) = -E(\boldsymbol{c}) + \sum_{k=1}^{N} \delta_k(\boldsymbol{c}), \tag{3.2}$$

where the potential is

$$\delta_k(\boldsymbol{c}) = \begin{cases} -\infty & \text{if } c_k \neq k \wedge \exists\ i \mid c_i = k \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

The solution of this maximization problem can then be found representing equation (3.2) with a *factor graph*, in which functions and variables of (3.2) correspond to "function nodes" and "variable nodes" of the graph, and where connections, i.e., the edges of the graph, exist only between variables and functions depending on them, following standard graph theory [16]. In this way, $S(\boldsymbol{c})$ is the so called *global function* of the factor graph just described and can be maximize using the max-sum algorithm, that is the log-domain version of the well known max-product algorithm [16][1].

3.1.3 The algorithm

Input

Affinity Propagation takes as input the $N \times N$ symmetric matrix $s(i, k)$ of real-valued similarities between the N data points to be clustered, where similarity $s(i, k)$ is a measure of how well the data point k is suited to be the exemplar for data point i[2].

Rather than requiring, as other algorithms do, the number of clusters to be set by hand at the beginning, Affinity Propagation needs as input parameters only the so-called "preferences", i.e., the auto-similarity $s(i, i)$.

When the similarity matrix is computed starting from the actual

[1] Factor graph theory is a powerful instrument to represent and solve complex equations and logical problems. Since it is not the main subject of the present work and since it is a quite extended subject, it cannot be explained here. If the reader is interested, besides the exhaustive book by Mezard and Montanari [16], a very clear but shorter introduction to factor graph theory is [17].

[2] For a practical example, see 3.3, where we explain how $s(i, k)$ can be computed between two amino acids sequences i and k.

sequences, diagonal elements $s(i,i)$ are obviously equal to 1, since any element is identical to itself, so that diagonal entries of the similarity matrix do not carry any information. Besides, looking at the updating rule (3.5) in 3.1.3, one sees that, for any i, $s(i,i)$ is in fact used to enforce the i element probability to be chosen as an exemplar. For this reason, one has to replace them with real numbers within the interval $[0,1]$, indicating how every data point i is likely to be chosen as an exemplar. The number of clusters (equal to the number of exemplars) depends, of course, on the preferences chosen, but also emerges freely from the message-passing procedure.

Since we don't have any a priori information about which sequences are more likely to be exemplars, we have set a common value $s(i,i) = \alpha$ for every i. Changing this value sets the "granularity" of the clustering procedure. As suggested in [13], a good choice for a common value α is the minimum between the entries of the similarity matrix.

Initialization

The algorithm works exchanging two kinds of messages between data points, each one of them takes into account a different kind of competition: there is responsibility $r(i,k)$, sent from data point i to the candidate exemplar k, reflecting how k is well-suited to be the exemplar for i, taking into account the other potential exemplars for i; and the availability $a(i,k)$, sent from the candidate exemplar k to data point i, reflecting how available is k to be the exemplar for i, taking into account all the other data points interested in having the same k as exemplar.

Initialization consists in setting the availabilities to zero:

$$a(i,k) = 0 \quad \forall\ i,k. \tag{3.4}$$

Updating rules

Once availabilities have been initialized, the iterative procedure begins, respecting the presented order:

1. responsibilities are updated, following[3]

$$r(i,k) \leftarrow s(i,k) - \max_{k' \neq k}\{a(i,k') + s(i,k')\}; \tag{3.5}$$

[3] In (3.5), (3.6) and (3.7) the arrows $\leftarrow$ indicate that the quantity on the left is replaced by the quantity on the right.

2. availabilities with $i \neq k$ are updated, as

$$a(i,k) \leftarrow \min\{0, r(k,k) + \sum_{i' \neq i,k} \max[0, r(i',k)]\}; \tag{3.6}$$

3. self-availabilities $a(k,k)$ are updated, but following the different rule

$$a(k,k) \leftarrow \sum_{i' \neq k} \max\{0, r(i',k)\}. \tag{3.7}$$

The physical meaning of the updating rules just presented is simpler than it might appears at first sight.

During the first iteration (3.5), for example, because availabilities are all equal to zero, the responsibility $r(i,k)$, i.e., how k is well-suited to be the i exemplar, is simply equal to the similarity between i and k minus the largest of the similarities between i and other candidate exemplars. Hence, this first update is completely determined by the initial data. In later iterations, however, when some points have been already assigned to other exemplars (different from k) the relative availabilities will became negative because of the second updating rule and this negative availabilities will decrease the similarities $s(i',k)$, removing the corresponding candidate exemplars from competition.

Similarly the availability $a(i,k)$ of the point k as a candidate exemplar for i is set, in (3.6), to its "self-responsibility" $r(k,k)$, reflecting accumulated evidence that point k is an exemplar, plus positive responsibilities received from other points. If, for instance, the self-responsibility $r(k,k)$ is negative (thus k appears to be not suitable as exemplar, maybe as a consequence of a small auto-similarity $s(k,k)$), the availability of k can be increased by the presence of other points that see k as a good exemplar. And so on.

Convergence and iteration stop

For any point i, the point k that maximizes the sum $a(i,k) + r(i,k)$ identifies the data point that is the exemplar for point i (obviously if $k = i$, i is itself an exemplar). This can be done at any iterative step, viewing how clusters emerge from the starting data points during the procedure. In fact, the two messages $a(i,k)$ and $r(i,k)$ converge after some iterations, although with the help of a dumping factor λ used to avoid oscillations: at every step messages are set to λ times their previous value plus $(1-\lambda)$ times the current one. The algorithm stops when the message updates fall below a customary threshold.

3.2 Sampling regime

In this section we introduce some fundamental theoretical issues concerning complex systems and their predictability, in a very general fashion. Explaining these theoretical issues here, we will not have to discuss them later and in the following section the clustering results presented will be immediately clear to the reader.

The final goal of this section is the identification of some simple instruments allowing us to understand if a sample of a given complex system (in our case an MSA collecting different realizations of HA protein sequence) is representative of the real system, i.e., if it contains enough information about the latter.

3.2.1 Introduction and definitions

Complex systems are systems that we are able to represent, at best, with models that are not only approximate, but incomplete. Within these models, in addition to variables we know we are neglecting, there are unknown unknowns we do not even know they exist and have an effect. Here a complex system is defined as a system assumed to maximize an objective function $U(\boldsymbol{s})$ depending on a large number of variables $\boldsymbol{s} = (s_1, \ldots, s_n)$, a part of which are unknown.

For our application, this assumption is quite well suited with reality: we expect that for a given protein, the amino acids sequence realizations respond to the request of minimizing the energy function depending on chemical bonds, maximizing the entropy, or, more in general, optimizing the fulfillment of a specific task.

However, although the objective function of a given complex system depends on all the variables $\boldsymbol{s} = (\tilde{\boldsymbol{s}}, \hat{\boldsymbol{s}}) = (s_1, \ldots, s_{\tilde{n}}, s_{\tilde{n}+1}, \ldots, s_n)$, only $\tilde{n}$ of them are known to the modeler $(\tilde{\boldsymbol{s}})$ (being the other variables $\hat{\boldsymbol{s}}$ unknown), as well as only the part of the objective function, $u_{\tilde{s}}$, that depends exclusively on them.

We can then split the whole $U(\boldsymbol{s})$ function in two parts

$$U(\boldsymbol{s}) = u_{\tilde{s}} + v_{(\tilde{s},\hat{s})}, \tag{3.8}$$

the known one, depending only on the known variables, and the unknown one, depending, a priori, on all the variables $\boldsymbol{s}$.

Formally one can define the observed part $u_{\tilde{s}}$ as the objective function averaged over the unknown variables:

$$u_{\tilde{s}} = E_{\hat{\boldsymbol{s}}}[U(\boldsymbol{s})], \tag{3.9}$$

where $E_{\hat{\boldsymbol{s}}}[\dots]$ is the expected value with respect to an a priori distribution $p(v)$ on the dependence of $U(\boldsymbol{s})$ on $\hat{\boldsymbol{s}}$; in other words, we are assuming that the unknown function $v_{(\tilde{s},\hat{s})}$ is randomly and independently drawn for each $\boldsymbol{s}$ from a given distribution $p(v)$.

In this framework, the behavior of the real system is represented by

$$\boldsymbol{s}^* = (\tilde{\boldsymbol{s}}^*, \hat{\boldsymbol{s}}^*) = \arg\max_{\boldsymbol{s}} U(\boldsymbol{s}), \tag{3.10}$$

while the behavior predicted by the model, relying only on the known variables, is represented by

$$\tilde{\boldsymbol{s}}_0 = \arg\max_{\tilde{\boldsymbol{s}}} u_{\tilde{\boldsymbol{s}}}. \tag{3.11}$$

Hence, the predictability of the model can be defined as the probability that the known variables maximizing the known function are the same needed to maximize the whole objective function:

$$P(\tilde{\boldsymbol{s}}_0 = \tilde{\boldsymbol{s}}^*) \equiv E_{\hat{\boldsymbol{s}}}[\delta_{\tilde{s}_0,\tilde{s}^*}]. \tag{3.12}$$

Besides the abstract definition of (3.12), one can derive the probability distribution $p_{\tilde{\boldsymbol{s}}}$ for a generic configuration $\tilde{s}$ to be the true maximum $\tilde{s}^*$, from which probability (3.12) is obtained as $p_{\tilde{\boldsymbol{s}}_0} = P(\tilde{\boldsymbol{s}}_0 = \tilde{\boldsymbol{s}}^*)$, under very general conditions. It takes the form of a Gibbs-Boltzmann distribution.

3.2.2 The Gibbs-Boltzmann distribution

If we assume all the moments of the $p(v)$ distribution to be finite

$$E_{\hat{s}}[v^m_{(\tilde{s},\hat{s})}] \quad \forall m > 0 \tag{3.13}$$

and we take the thermodynamic limit $n \to \infty$, we are able to obtain straightforward the distribution

$$p_{\tilde{\boldsymbol{s}}} \equiv P(\tilde{\boldsymbol{s}} = \tilde{\boldsymbol{s}}^*), \tag{3.14}$$

i.e., the probability distribution that a generic observed configuration $\tilde{\boldsymbol{s}}$ is the true maximum $\tilde{\boldsymbol{s}}^*$.

This can be done simply using the maximum entropy principle. In fact, on the true maximum $\tilde{\boldsymbol{s}}^*$ the known function $u_{\tilde{\boldsymbol{s}}}$ will be, by definition, less than or equal to the one evaluated on the predicted maximum $\tilde{\boldsymbol{s}}_0$: $u_{\tilde{\boldsymbol{s}}^*} \leq u_{\tilde{\boldsymbol{s}}_0}$. In statistical mechanics, since energy is an extensive quantity, can be shown that imposing the constraint $E_{\tilde{s}}[u_{\tilde{s}}] \leq u_{\tilde{s}^*}$ gives

the same result as imposing $E_{\tilde{s}}[u_{\tilde{s}}] = u_{\tilde{s}^*}$. Assuming this conclusion to be valid also in this context[4], $p_{\tilde{s}}$ distribution can be searched as the maximum entropy distribution with the constraint $E_{\tilde{s}}[u_{\tilde{s}}] = u_{\tilde{s}^*}$[5]. As well known ([11],[12]), this distribution is the Gibbs-Boltzmann distribution

$$p_{\tilde{s}} = \frac{\exp(\beta u_{\tilde{s}})}{Z(\beta)}, \tag{3.15}$$

where $Z(\beta)$ is the partition function

$$Z(\beta) = \sum_{\tilde{\boldsymbol{s}}'} \exp(\beta u_{\tilde{s}'}). \tag{3.16}$$

Jaynes derivation of Gibbs-Boltzmann distribution using maximum entropy principle, however, give no information on the nature of the generalized β constant, while extreme value theory [14] shows a precise relation between β and the number $(n - \tilde{n})$ of unknown variables $\hat{\boldsymbol{s}}$. In fact, one can find ([14],[7]) that if the asymptotic behavior of $p(v)$ for $v \to \infty$ is $\ln p(v) \sim -|v|^{\gamma}$, then

$$\beta = [(n - \tilde{n}) \ln 2]^{1 - \frac{1}{\gamma}}. \tag{3.17}$$

Equation (3.17) is quite surprising. It tells us that for $p(v)$ decaying faster than exponential, i.e., for $\gamma > 1$ (as for Gaussian variables), β diverges with the number of unknowns and the predictability of the model $p_{\tilde{\boldsymbol{s}}_0} = P(\tilde{\boldsymbol{s}}_0 = \tilde{\boldsymbol{s}}^*)$ grows. Keeping the number of known variables $\tilde{n}$ finite, for $(n - \tilde{n}) \to \infty$ we have that $p_{\tilde{\boldsymbol{s}}_0} \to 1$.

This non-trivial behavior suggests that models are predictable only when the number of unknown variables is *large* enough, or, conversely, when the number of relevant known variables is *less* than a critical threshold [7]. Although this conclusion may seem counter-intuitive, if it wasn't true, scientific approach on complex system would simply be impossible.

3.2.3 Sample information

We have seen that, for a given known objective function $u_{\tilde{\boldsymbol{s}}}$, the probability to observe a certain state $\tilde{s}$ of a complex system follows the distribution (3.15). Bearing in mind what we just found, we can now

[4] In fact $p_{\tilde{s}}$ distribution can be derived as a Gibbs-Boltzmann distribution also using extreme value theory [7], so that this assumption can be justified *a posteriori*.

[5] Here $E_{\tilde{s}}[\dots]$ is the mean value calculated with respect to $p_{\tilde{s}}$ distribution, differently from $E_{\hat{s}}[\dots]$ of equation (3.13), evaluated with respect to $p(v)$ distribution.

work on the inverse situation, i.e., we observe some states of a complex system and we want to extract from them information about the function $u_{\tilde{\boldsymbol{s}}}$, now unknown.

In fact, we can think to a sample of N observations of the state of a complex system $(\tilde{s}^1, \ldots, \tilde{s}^N)$ as N independent realizations of the Gibbs-Boltzmann distribution (3.15) with some function $u_{\tilde{\boldsymbol{s}}}$.

If we call $K_{\tilde{\boldsymbol{s}}}$ the number of times the $\tilde{\boldsymbol{s}}$ configuration appears in our sample

$$K_{\tilde{\boldsymbol{s}}} = \sum_{j=1}^{N} \delta_{\tilde{\boldsymbol{s}}^j, \tilde{\boldsymbol{s}}}, \tag{3.18}$$

then the observed frequency $K_{\tilde{\boldsymbol{s}}}/N$ samples the distribution $p_{\tilde{s}}$ and so gives us a rough estimate of the function

$$u_{\tilde{\boldsymbol{s}}} \approx a + \frac{1}{\beta} \ln K_{\tilde{\boldsymbol{s}}}, \tag{3.19}$$

for some constant a. From this one can argue that is the multiplicity $K_{\tilde{\boldsymbol{s}}}$ with which $\tilde{\boldsymbol{s}}$ appear in our sample to bring information about the system, rather than $\tilde{\boldsymbol{s}}$ itself. This can be shown in a more formal and interesting way using the two entropies associated to the distribution of $\tilde{\boldsymbol{s}}$ and $K_{\tilde{\boldsymbol{s}}}$ as random variables.

Since *a priori* all the N realizations of the sample should have the same probability $1/N$, we have for $\tilde{\boldsymbol{s}}$ the already presented distribution $P(\tilde{\boldsymbol{s}}^i = \tilde{\boldsymbol{s}}) = K_{\tilde{\boldsymbol{s}}}/N$, and for $K_{\tilde{\boldsymbol{s}}}$ the distribution $P(K_{\tilde{\boldsymbol{s}}^i} = k) = km_k/N$, where

$$m_k = \sum_{\tilde{\boldsymbol{s}}} \delta_{k, K_{\tilde{s}}} \tag{3.20}$$

is the number of times we encounter, within our sample, a state $\tilde{\boldsymbol{s}}$ appearing exactly k times in the sample.

The respective empirical[6] entropies associated to these distributions are

$$H[\tilde{\boldsymbol{s}}] = -\sum_{\tilde{\boldsymbol{s}}} \frac{K_{\tilde{\boldsymbol{s}}}}{N} \ln\left(\frac{K_{\tilde{\boldsymbol{s}}}}{N}\right) = -\sum_{k} \frac{km_k}{N} \ln\left(\frac{k}{N}\right) \tag{3.21}$$

$$H[K] = -\sum_{k} \frac{km_k}{N} \ln\left(\frac{km_k}{N}\right) = H[\tilde{\boldsymbol{s}}] - \sum_{k} \frac{km_k}{N} \ln(m_k), \tag{3.22}$$

where, besides the misleading notation, $H[K]$ is a function of $\{m_k\}$ and where the second equality of (3.21) follows from the definition of m_k in

[6]The term "empirical" is referred to the fact that these entropies are computed using the finite data sample $(\tilde{s}^1, \ldots, \tilde{s}^N)$.

(3.20).

Because of (3.19) one can conclude that the information contained in our sample of data $(\tilde{s}^1, \ldots, \tilde{s}^N)$ on the behavior of the system, i.e., on the function $u_{\tilde{\boldsymbol{s}}}$, is quantified by the entropy $H[K]$ and not by $H[\tilde{\boldsymbol{s}}]$ [7].

This conclusion can be better understood in two simple and extreme examples: the situation in which each state appears in the sample at most one time (so that $K_{\tilde{\boldsymbol{s}}} = 1$ for all states $\tilde{\boldsymbol{s}}$ in the sample and $K_{\tilde{\boldsymbol{s}}} = 0$ otherwise, $m_1 = N$ and $m_k = 0\ \forall k \neq 1$); and the opposite situation, in which our sample is composed by only one state $\tilde{\boldsymbol{t}}$ appearing N times (so that $K_{\tilde{\boldsymbol{s}}} = N\delta_{\tilde{\boldsymbol{s}},\tilde{\boldsymbol{t}}}$, $m_N = 1$ and $m_k = 0\ \forall k \neq N$). In both these examples we cannot extract any information upon the function $u_{\tilde{\boldsymbol{s}}}$ and in both these examples $H[K] = 0$, while $H[\tilde{\boldsymbol{s}}] = \ln N$ in the first and $H[\tilde{\boldsymbol{s}}] = 0$ in the second.

Since all the other situations stand between these two extreme cases, for the former we expect $H[\tilde{\boldsymbol{s}}]$ to take an intermediate value in $[0, \ln N]$ and we expect to have a positive amount of information $H[K] > 0$ on the system behavior.

Let us show more in detail what happens in these intermediate cases.

3.2.4 $H[K]$ vs $H[\tilde{\boldsymbol{s}}]$ curve and Zipf's law

In order to solve this problem, instead of seeing $p_{\tilde{\boldsymbol{s}}}$ distribution (3.15) as the maximal entropy distribution subject to the constraint $E_{\tilde{s}}[u_{\tilde{s}}] = u_{\tilde{s}^*}$, we think to it as the distribution of maximal $E_{\tilde{s}}[u_{\tilde{s}}] = \sum_{\tilde{s}} p_{\tilde{s}} u_{\tilde{s}}$ but with fixed information content, i.e., fixed entropy $\bar{H}$. This gives us a natural upper bond for empirical entropy over states $\tilde{\boldsymbol{s}}$, i.e., $H[\tilde{\boldsymbol{s}}] \leq \bar{H}$, as a consequence of the asymptotic equipartition property [15].

We are then interested to search among the possible distributions $\boldsymbol{m} = \{m_k,\ k > 0\}$ respecting this inequality, the ones for which $H[K]$ is maximal

$$\boldsymbol{m}^* = \underset{\boldsymbol{m},\ H[\tilde{s}] \leq \bar{H}}{\arg\max}\ H[K] \tag{3.23}$$

with the constraint $\sum_k k m_k = N$. Furthermore, being $K_{\tilde{s}}$ a function of $\tilde{\boldsymbol{s}}$, data processing inequality [15] implies also that $H[K] \leq H[\tilde{\boldsymbol{s}}]$.

In the $H[k] < H[\tilde{\boldsymbol{s}}]$ region, the solution (3.23) can be found following the approximation used in [7], where m_k, instead of being considered a positive integer number, is treated as a positive *real* one, and maximizing the m_k function

$$H[K] + \mu(H[\tilde{\boldsymbol{s}}] - \bar{H}) + \nu(\sum_{k>1} k m_k - N), \tag{3.24}$$

where μ and ν are Lagrange multipliers related to the two conditions $H[\tilde{\boldsymbol{s}}] = \bar{H}$ and $\sum_k km_k = N$.

Substituting (3.21) and (3.22) in (3.24), and taking the derivative with respect to m_k, one obtains the power law function

$$m_k^* = zk^{-(1+\mu)}, \tag{3.25}$$

where $z > 0$ is a normalizing constant and where, obviously, $k \in [1, N]$.

Substituting the solution (3.25) in (3.21) and (3.22), and computing both the entropies for different values of μ, one obtains a curve in the $H[\tilde{\boldsymbol{s}}] \times H[K]$ plane that gives, for any value of $H[\tilde{\boldsymbol{s}}]$, the *maximal* value of the entropy $H[K]$. Empirical samples of data generate points $(H[\tilde{\boldsymbol{s}}], H[K])$ standing below this maximal curve. Two examples of this curve are shown in the right side of figure 3.1.

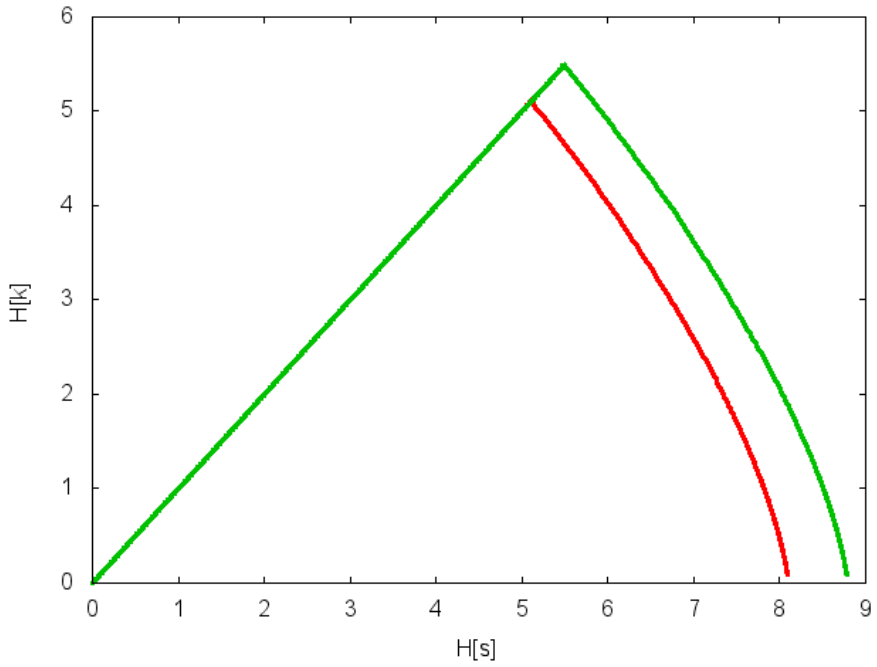

Figure 3.1: Maximal $H[K]$ versus $H[\tilde{\boldsymbol{s}}]$ for $N = 3297$ (green) and $N = 6573$ (red). Empirical samples of data generate points $(H[\tilde{\boldsymbol{s}}], H[K])$ standing below this maximal curve (physical region). The linear part of the red curve is hidden behind the green one.

The right part of the curve corresponds to the under sampled regime region, because $H[K] < H[\tilde{\boldsymbol{s}}]$. The points of the curve in this region

represent possible N-sized samples of the system, with different distributions: in fact, to any point is associated a different couple of entropies $H[\tilde{\boldsymbol{s}}]$ and $H[K]$ and corresponds to a different value of the μ parameter, that describes different power law distributions (3.25) of the data.

If we start from the non-informative point with $(H[\tilde{\boldsymbol{s}}], H[K]) = (\ln N, 0)$ at the extreme right of the curve, with $\mu = \infty$ (corresponding to the distribution $m_k = N\delta_{k,1}$), we can climb over the curve towards the peak decreasing μ. In fact, at any point of this curve, $-\mu$ represents the slope of the line tangent to that point: again from (3.21) and (3.22), using the solution (3.25), one can obtain with few trivial passages that $H[K] = -\mu H[\tilde{\boldsymbol{s}}]$.

At the peak, the entropy $H[K]$, derived maximizing (3.24), reaches the value of $H[\tilde{\boldsymbol{s}}]$ ($H[K] = H[\tilde{\boldsymbol{s}}]$), the slope of the curve becomes $\mu = -1$ and the power law describing m_k distribution over k is the Zipf's law

$$m_k \sim k^{-2}. \tag{3.26}$$

Since, for the data processing inequality, we have that $H[K] \leq H[\tilde{\boldsymbol{s}}]$, for $\mu < -1$ the solution of the maximization problem (3.25) is no longer valid. Hence, at the left of the peak (see figure 3.1) the curve is simply the line $H[K] = H[\tilde{\boldsymbol{s}}]$. Points in this region correspond to samples where every state $\tilde{\boldsymbol{s}}$ appears a different number of times with respect to any other, so that knowing the frequency $K_{\tilde{s}}/N$ one can recognize the state $\tilde{\boldsymbol{s}}$ itself: their distributions are equivalent.

Summarizing: in the under sampled regime, where $H[K] < H[\tilde{\boldsymbol{s}}]$, distributions with the largest information content show a power law behavior[7]; in particular, the distribution with the highest information content coincides with the Zipf's law, which appears at the crossover between the under sampled regime and the regime where $H[K]$ coincides with $H[\tilde{\boldsymbol{s}}]$.

3.3 Clustering HA sequences

We can now put together the clustering algorithm presented in the first section and the theoretical instruments illustrated in the second: with the clustering program we are able to split our sample in clusters of similar sequences; taking the number m_k of clusters of size k we can

[7] It is interesting to notice that power laws, that have the fundamental property of being scale invariant, are deeply related in statistical physics to critical points of phase transition diagrams.

calculate the entropies $H[\tilde{\boldsymbol{s}}]$ and $H[K]$ using (3.21) and (3.22) (this has been done directly within the clustering program, see Appendix A); then, interpolating the (k, m_k) points, we can also obtain an estimate of the power law exponent $-(1+\mu)$. Repeating this procedure several times, while coarse-graining the clustering, gives some interesting results.

However, since the outcome of the algorithm presented in 3.1.3 is very stable over the input parameter $\alpha = s(i,i)\ \forall i$ (whose coherent values are between the smallest similarity matrix entry and 1), to change the grain of the clustering we have to construct and gives to the algorithm different similarity matrix, calculated with a measure of similarity that became more and more "permissive". A simple way to do that is looking only at the most conserved sites over the sample of sequences.

3.3.1 Clustering over most conserved sites

The idea is to set a reference value $X \in (0,1)$ for the frequencies and then to consider as most conserved sites, within the total number of sites $n = 566$, the sequence positions i for which

$$\exists\ A \mid f_i(A) = \frac{1}{N}\sum_{j=1}^{N}\delta_{A,A_i^j} \geq X, \tag{3.27}$$

where j runs over the N sequences of the sample and A_i^j is the amino acid at position i of the j sequence. For a definite value of X, there will be a precise number of sites $\tilde{n}_X \leq n$ for which (3.27) is verified. To gain intuition, we can look at the extreme cases $X = 0$ and $X = 1$: for the first, obviously, $\tilde{n}_0 = n$; in the second, $\tilde{n}_1$ is the number of sites showing over the sample *always* the same amino acid.

Cluster the sequences looking only at the most conserved sites means to cluster the sequences using as input a similarity matrix where similarities between sequences are computed looking only at the most conserved sites. For different values of X, we can calculate the similarity $s(i,k)_X$ between the sequences i and k as $[1 - d(i,k)_X]$, where $d(i,k)_X$ is the normalized humming distance between the two sub-sequences composed only of the $\tilde{n}_X$ most conserved sites

$$d(i,k)_X = \sum_{j=1}^{\tilde{n}_X}\frac{\delta(A_k^i, A_j^k)}{\tilde{n}_X}, \tag{3.28}$$

being $\tilde{n}_X$ the sub-sequences length and A_j^i the amino acid in position j of the i sequence; j runs over the most conserved sites.

Obviously taking high value of X means to compare sequences looking at a small amount $\tilde{n}_X$ of sites, where, moreover, there is an amino acid A that has high frequency $f(A) \geq X$. For this reason, for high X, we are telling the algorithm to see as very similar sequences that can be, in fact, very different in the least conserved sites. For example, with $X = 1$, every sequence is *identical* to any other (where i and k sequences are identical if $s(i, k) = 1$), so that the clustering algorithm becomes useless, i.e., putting $X = 1$ we directly reduce all the sample to a unique cluster of size N ($m_N = 1$), reaching the point (0,0) of the entropy curve. With $X = 0$, besides, we compare sequences using their complete length, so that we maximize the distances (3.28) between the sequences, and, conversely, we lower down the similarities, obtaining a large number of small clusters and reaching the extreme right of the entropy curve.

We expect, raising the value of X from 0 to 1, to climb the empirical version of the curves in figure 3.1 towards the peak from the right and then to climb it down along the left side. And that is, in fact, what happens.

3.3.2 Clustering results

The procedure explained above has been repeated for 21 values of X, between 0.5 and 1. Plotting $H[K]$ and $H[\tilde{s}]$ for every clustering outcome, one obtains the curve of points in figure 3.2. As expected, empirical points stand below the maximal curve, but reproduce quite well its shape, and this is not an obvious result. In fact, lacking data samples are not able to reproduce the curve and their clustering entropies draw hills flattened to the axis $H[K] = 0$; very good data samples, on the contrary, are expected to generate high hills, closer to the maximal curve.

We know that, in the under sampled regime, high information content samples are related to power law distribution of the data, with an exponent that approaches -2 for distributions with highest information content. In order to show, in our sample, the emergence of a similar behavior, we looked at two distributions of our data: the "natural" distribution, where with "natural" we mean "obtained comparing the sequences looking at a large number of sites $\tilde{n}_X \approx n$", i.e., using a low threshold X; and the distribution related to the top of the curve in figure 3.2. We aspect to obtain power law behavior in both cases, but with $-(1 + \mu)$ exponent decreasing between the first and the second, in which, in particular, it has to be close to -2.

In figure 3.3 are plotted the point $(\ln k, \ln m_k)$ (where m_k is the number of clusters of size k), for two value of X: $X = 0.7$[8] (green dots), corresponding to a point on the low right side of the curve in figure 3.2; and $X = 0.97$ (red points), corresponding to the top of the curve, i.e., to the point with maximal information content $H[k]$.

They show, as expected, a linear dependency, with slope (the exponent of the power law (3.25)) close to -3 in the first (green) case and to -2 in the second (red), as shown by figure 3.4 and 3.5.

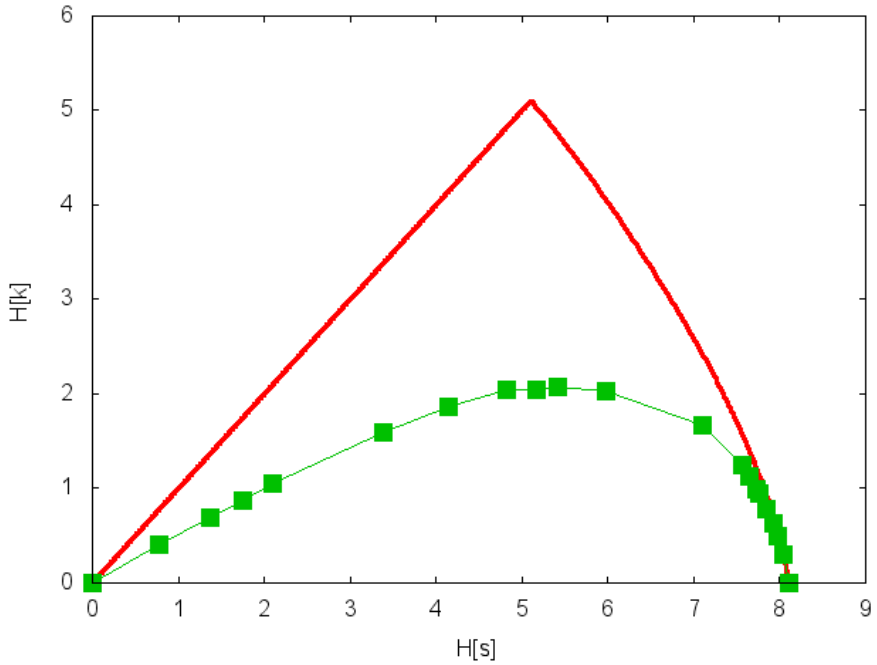

Figure 3.2: Maximal $H[K]$ over $H[\tilde{\boldsymbol{s}}]$ curve for $N = 3297$ (red); empirical points $(H[K], H[\tilde{\boldsymbol{s}}])$ for 21 different values of X threshold (green). As expected, empirical points stand below the maximal curve, but reproduce quite well its shape.

[8]Since our sequences show high conservation (as a consequence of the fact that they represent a specific subtype of a specific protein) values of the threshold X must stand in the interval $[0.6, 1]$ in order to see some changing in clustering "granularity". For this reason 0.7 can be considered a small value of X, in fact the relative number of sites to look is $\tilde{n}_{0.7} = 531$.

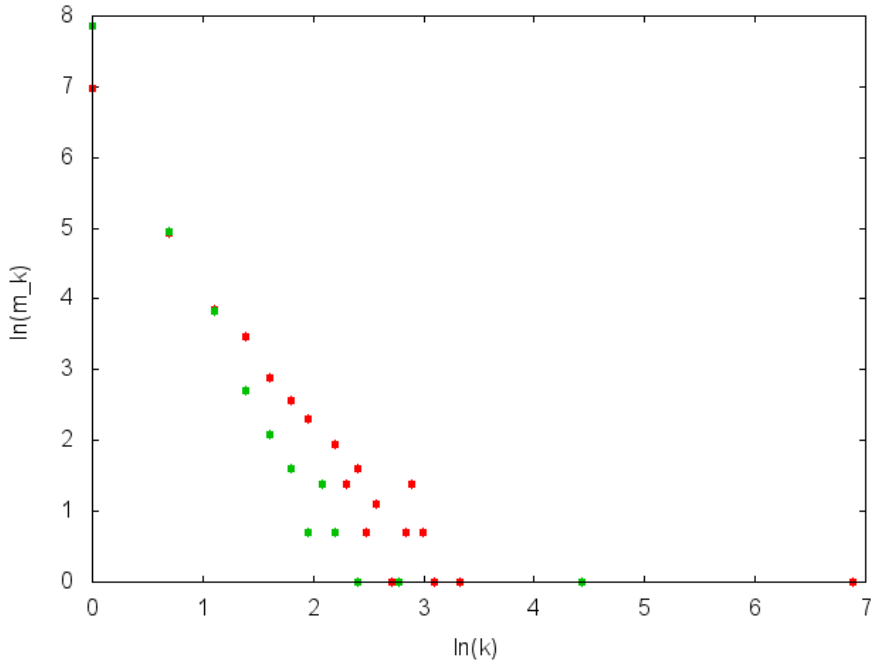

Figure 3.3: $(\ln k, \ln m_k)$ points for $X = 0.7$ (green) and $X = 0.97$ (red) frequency threshold; m_k is the number of clusters of size k resulting from a clustering procedure that starts from the input matrix $s(i,k)_X$ defined through equation (3.28) and with $s(i,i)_X = \alpha \equiv \min_{j,k}[s(j,k)_X]$, as explained in 3.1.3.

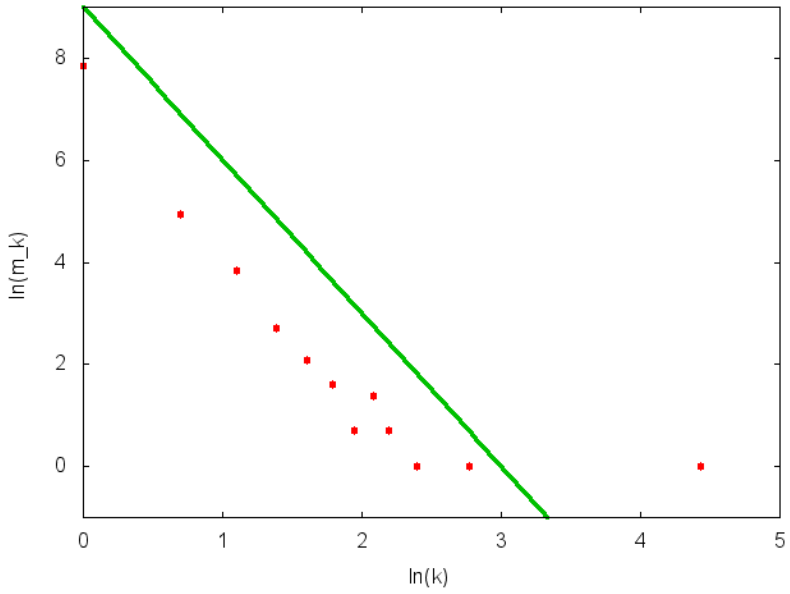

Figure 3.4: $(\ln k, \ln m_k)$ points for $X = 0.7$ clustering (red); reference line $\ln(m_k) = -3\ln(k) + 9$ (green).

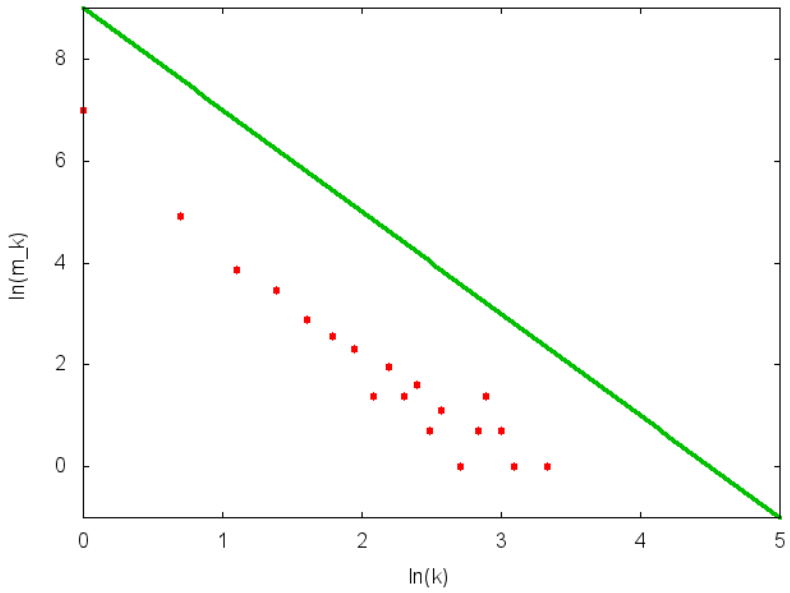

Figure 3.5: $(\ln k, \ln m_k)$ points for $X = 0.97$ clustering (red); reference line $\ln(m_k) = -2\ln(k) + 9$ (green).

3.3.3 Final observations

We have just seen that our sample behaves as theoretically predicted, showing power law distribution in frequencies and being able to climb the entropy curve $H[K]$ over $H[\tilde{\boldsymbol{s}}]$ if we coarse-grain the clustering procedure. However, using the frequency threshold X (and hence the number of most conserved sites $\tilde{n}_X$) to change the clustering granularity allows us to look at these results also from another perspective.

An HA sequence is composed by $n = 566$ amino acids, hence it can be represented by a vector of n components $\boldsymbol{s} = (s_1, \ldots, s_n)$, every one of them coding for the amino acid in the corresponding position. $\boldsymbol{s} = (s_1, \ldots, s_n)$ are, in fact, the variables describing our system (the protein). A statistical physicist can argue that within the n amino acids, i.e., the n variables $\boldsymbol{s}$, there may be a fraction $\tilde{n}$ of them ($\tilde{\boldsymbol{s}}$) more relevant for describing the protein and its functional role, and others ($\hat{\boldsymbol{s}}$) less relevant.

Looking at the $\tilde{n}_X$ most conserved sites for increasing values of X, in fact, one restricts the number of relevant variables useful to describe the system. Since doing that the entropy $H[K]$ climbs the curve in figure 3.2 and reaches the point of highest informativity for $X = 0.97$, corresponding to a number of relevant sites $\tilde{n}_{0.97} = 431 < n$, one can argue that our sample contains enough statistics to let us recognize, with respect to a specific issue, more informative variables ($\tilde{n}_{0.97}$) and less informative ones ($n - \tilde{n}_{0.97}$).

Such a conclusion is both interesting and reassuring, since in the next chapters we are going to address this variables-discerning work on the same sample of data, although, as we will soon see, with a different focus: correlations.

Chapter 4

DCA and SCA inference methods

Once the information content of the data sample presented in chapter 2 has been analyzed using the instruments illustrated in chapter 3, we can proceed in the attempt of successfully extracting such information.

As a large number of recent studies suggests ([1], [2], [3], [4], [5]), in order to infer information from a multiple sequence alignment (MSA) about the structure of the relative protein, natural quantities to look at are statistical pair-wise correlations between amino acids at different positions in the MSA[1].

Several methods have been developed in order to recognize, starting from correlations and studying their patterns, the existence of specific interactions[2] between sites of the sequences, i.e., between amino acids of the protein.

Since these interactions are of key importance in order to ensure the functionality of the protein, they constitute evolutionary constraints to free and independent single site mutations [1]. Knowing them allows us to better understand past and future evolution of a protein, as, in this specific case, the influenza virus HA protein.

In this chapter we briefly present the two methods we tried (the first with no success) to apply to our HA protein sequences, while first

[1] In fact, single abundances and pair-wise correlations correspond to the first two moments of the unknown distribution representing the system; higher moments, although useful to characterize the distribution, cannot be properly quantified using the (poor) statistics offered by a typical sample of protein sequences.

[2] With *interactions*, here, we don't mean only 3D contacts between amino acids in the folded configuration of the protein, but also other types of functional relations and interdependencies.

results (of the second method) will be presented in the final section. These two methods, called DCA — Direct Couplings Analysis — and SCA — Statistical Couplings Analysis — start from the same point, the matrix of empirical correlations, but proceed in different directions: while the first has been mainly conceived in order to infer, from an MSA, the 3D contacts of the folded protein and hence tries to obtain, as we will see, 1 to 1 interactions between single sites of the sequences, the second method searches for more collective interactions, i.e., aims to find bigger ensembles of interacting sites along the sequence called *sectors*.

4.1 Direct Coupling Analysis

Direct Coupling Analysis has been implemented by its authors using different strategies: a message-passing algorithm [5], pseudo-likelihoods maximization [4] and by means of a mean field approximation [3]. We chose to apply to our MSA the last algorithm, since it is the simplest one and presents the same (sometimes better [3]) predictive accuracy of the others.

4.1.1 Input

Mean field DCA algorithm takes as input a multiple sequence alignment (MSA), that is, as explained in 2.3, a rectangular $N \times L$ array

$$\boldsymbol{A} = A_i^a;\ i = 1, \ldots, L;\ a = 1, \ldots, N \tag{4.1}$$

where L is the protein length, N is the number of sequences and A_i^a is the amino acid in position i of the a sequence, i.e., a letter of the IUPAC alphabet in table 2.1. For simplicity, it is useful to translate the 20 amino acids plus 1 gap IUPAC alphabet into $q = 21$ consecutive numbers $1, \ldots, q$.

4.1.2 Single and double site frequencies

In order to build up the correlation matrix, one has first to compute single and double site frequencies, where the first gives the fraction of sequences showing amino acid B[3] in position i

$$f_i(B) = \frac{1}{N} \sum_{b=1}^{N} \delta_{B,A_i^b} \tag{4.2}$$

[3]From now on, to represent a *generic* amino acid we use a Latin capital letter. There is no relation between these letters and the IUPAC code in table 2.1.

and the latter the fraction of sequences showing at the same time amino acid B in position i and C in position j

$$f_{ij}(B,C) = \frac{1}{N}\sum_{b=1}^{N}\delta_{B,A_i^b}\delta_{C,A_j^b}, \tag{4.3}$$

with, obviously, $i,j \in [1,L]$ and $B,C \in [1,q]$.

From a theoretical point of view, DCA relies, as will be clearer in the following, on the assumption that MSA sequences are drawn independently from the same distribution. However, this is certainly not true: biological sequence data show a strong sampling bias due to phylogenetic relations, multiple strain sequencing and bias in the selection of the strains which are currently sequenced. For all these reasons and because the present method has to deal with a huge number of parameters, in [3] some possible corrections are proposed, i.e., the introduction of multiplicities and of the so-called pseudo-counts.

For the former, the idea is to count for every sequence $\boldsymbol{A}^a$ the number m^a of similar sequences $\boldsymbol{A}^b$ for which the overlap between the two sequences

$$\sum_{i=1}^{L}\delta_{A_i^a,A_i^b} \geq xL, \tag{4.4}$$

with $1 \leq b \leq N$ and where $x \in [0,1]$ is a *similarity threshold*: two sequences overlapping in a number of positions larger than xL are considered to carry almost the same information. Note that $m^a \geq 1\ \forall a$, since sequence A^a itself is also included.

Once multiplicities m^a have been computed, one can re-weigh the frequency counts (4.2) and (4.3) assigning weight 1 to sequences without similar sequences within the MSA and down-weighting sequences featuring m^a similar sequences in the MSA with a factor $1/m^a$. In this way, one gives smaller weight to strains which are more densely sampled and a higher weight to strains less densely sampled. If one takes $x = 1$ (as we have done) the effect is simply to remove repeats in the MSA (and hence, for our sample, to reduce the size from $N = 6573$ to $N = 3297$).

Besides, adding pseudo-counts (λ) to the empirical frequency counts (4.2) and (4.3), as in (4.5) and (4.6), is the same as adding extra observations to the real ones, in order to increase the size of the dataset. This is a standard tool in biological sequence analysis and can be justified in terms of Bayesian inference, under the hypothesis of having an a priori knowledge on sites occupations represented by a Dirichlet distribution[4].

[4]See Appendix B for details.

Putting together these two corrections, one obtains for single and double sites frequencies

$$f_i(B) = \frac{1}{\lambda + N_{ind}} \left(\frac{\lambda}{q} + \sum_{b=1}^{n} \frac{1}{m^b} \delta_{B,A_i^b} \right) \tag{4.5}$$

$$f_{ij}(B,C) = \frac{1}{\lambda + N_{ind}} \left(\frac{\lambda}{q^2} + \sum_{b=1}^{n} \frac{1}{m^b} \delta_{B,A_i^b} \delta_{C,A_j^b} \right) \tag{4.6}$$

where a good choice for pseudo-counts is $\lambda = N_{ind}$ [3] and where $N_{ind} = \sum_{b=1}^{n} (m^b)^{-1}$ is the effective number of independent sequences.

Since for statistically independent positions i and j, $f_{ij}(A,B) = f_i(A) f_j(B)$, to quantify correlation between i and j sites one can introduce [3] the Mutual Information

$$MI_{ij} = \sum_{A,B} f_{ij}(A,B) \ln \left(\frac{f_{ij}(A,B)}{f_i(A) f_j(B)} \right), \tag{4.7}$$

measuring how much of the information contained in $f_{ij}(A,B)$ is not already captured by single frequencies $f_i(A) f_j(B)$.

4.1.3 A statistical inference problem

However, statistical correlations between sites emerge as a consequence of direct interactions as well as from indirect ones, i.e., interactions mediated through different amino acids in other sites. To disentangle these contributions, the main idea proposed in [3] is to infer, from the actual MSA, a global statistical model, sampled by our MSA, from which one can obtain direct interactions.

This model is defined by the probability distribution $P(A_1, \ldots, A_L)$ of having a $(A_1, \ldots, A_L)$ sequence. In order to reproduce correctly the actual data sample, single and joint probability distributions are constrained to reproduce empirical single and double frequencies (4.5) and (4.6), i.e.,

$$P_i(A_i) \equiv \sum_{\{A_k | k \neq i\}} P(A_1, \ldots, A_L) = f_i(A_i) \tag{4.8}$$

$$P_{ij}(A_i, A_j) \equiv \sum_{\{A_k | k \neq i,j\}} P(A_1, \ldots, A_L) = f_{ij}(A_i, A_j). \tag{4.9}$$

It is straightforward to see that the simplest model $P(A_1, \ldots, A_L)$ respecting these constraints (via Lagrange multipliers) and maximizing

the entropy

$$S = -\sum_{\{A_k\}} P(A_1, \ldots, A_L) \ln P(A_1, \ldots, A_L), \tag{4.10}$$

is the 21-states Potts model

$$P(A_1, \ldots, A_L) = \frac{1}{Z} \exp\left\{ \sum_{i<j} e_{ij}(A_i, A_j) + \sum_i h_i(A_i) \right\} \tag{4.11}$$

where

$$Z = \sum_{\{A_i\}} \exp\left\{ \sum_{i<j} e_{ij}(A_i, A_j) + \sum_i h_i(A_i) \right\} \tag{4.12}$$

is the partition function. This is what expected, since, equivalently, one can notice that $f_i(A) = \langle \delta_{A,A_i} \rangle$, and the maximal entropy distribution with such a constraint, as already seen in 3.2.2, is a Gibbs-Boltzmann distribution.

Parameters $h_i(A_i)$ and $e_{ij}(A_i, A_j)$, introduced as Lagrange multipliers, can be interpreted respectively as local fields and coupling strengths and have to be tuned such that the constraints (4.8) and (4.9) are respected.

The number of these parameters is

$$\frac{L(L-1)}{2} q^2 + Lq. \tag{4.13}$$

However the two conditions (4.8) and (4.9) are not independent and our probability distribution is normalized, i.e.,

$$\sum_A f_i(A) = 1 \tag{4.14}$$

for every position $i = 1, \ldots, L$ and

$$\sum_B f_{ij}(A, B) = f_i(A) \tag{4.15}$$

$$\sum_A f_{ij}(A, B) = f_j(B) \tag{4.16}$$

for every couple of position (i, j) of the inequivalent $L(L-1)/2$. So that, holding these relations between the frequencies, one has to constrain only $(q-1)$ $P_i(A)$ for every i, and $(q-1)^2$ $P_{ij}(A, B)$ for every couple (i, j).

Hence, the number of *independent* parameters $h_i(A_i)$ and $e_{ij}(A_i, A_j)$ is only

$$\frac{L(L-1)}{2}(q-1)^2 + L(q-1), \tag{4.17}$$

and one can fix uniquely the solution of the model choosing arbitrary

$$e_{ij}(A, q) = e_{ij}(q, A) = h_i(q) = 0, \tag{4.18}$$

$\forall i, j = 1, \ldots, L$ and $\forall A = 1, \ldots, q$ [3].

Since the explicit computation of the marginal probability constraints (4.8) and (4.9) would require an exponential time, which grows like q^L, in order to successfully solve this inverse statistical problem we have to introduce an approximation.

4.1.4 Plefka expansion

Plefka expansion ([9],[10]) is nothing more than a Taylor expansion of the Hamiltonian of the Potts model (4.11) around zero couplings, i.e., treating the couplings $e_{ij}(A_i, A_j)$ as a perturbative term in the Potts model Hamiltonian

$$H(\alpha) \equiv -\alpha \sum_{i<j} e_{ij}(A_i, A_j) - \sum_i h_i(A_i), \tag{4.19}$$

where α is the perturbative parameter varying in $[0, 1]$. If $\alpha = 0$ we get an independent variable model, since couplings are neglected, while if $\alpha = 1$ we get back to the original Potts-model.

As anticipated at the beginning of this section, we are interested in the mean field approximation of this expansion. For practical purpose, we apply it to the Gibbs potential $G(\alpha)$, the Legendre transform of the free energy $F(\alpha) = -\ln Z(\alpha)$, rather than directly to $F(\alpha)$. The reason for this choice is simply that, while the free energy is a function of the couplings $e_{ij}(A, B)$ and of the fields $h_i(A)$, the Gibbs potential is a function of the couplings $e_{ij}(A, B)$ and of the conjugate variables of the fields $h_i(A)$, i.e., the probabilities $P_i(A)$[5], so that the first constraint (4.8) is satisfied for any value of α.

[5]This follows from the fact that

$$\frac{\partial F}{\partial h_k(C)} = P_k(C).$$

Summarizing, the idea is to expand, around $\alpha = 0$, the Gibbs potential $G(\alpha)$, that has the explicit form

$$\begin{aligned} G(\alpha) &= -\ln[Z(\alpha)] - \sum_{i=1}^{L}\sum_{B=1}^{q-1} h_i(B)P_i(B) \\ &= -\ln\left[\sum_{\{A_i|i=1,\dots,L\}} e^{-H(\alpha)}\right] - \sum_{i=1}^{L}\sum_{B=1}^{q-1} h_i(B)P_i(B). \end{aligned} \tag{4.20}$$

Truncating its expansion

$$G(\alpha) = G(0) + \left.\frac{\partial G(\alpha)}{\partial \alpha}\right|_{\alpha=0} \alpha + o(\alpha^2) \tag{4.21}$$

at the leading order in α, we obtain the mean field approximation

$$G^{MF}(\alpha) = G(0) + \left.\frac{\partial G(\alpha)}{\partial \alpha}\right|_{\alpha=0} \alpha. \tag{4.22}$$

Hence, the mean field approximation simplifies the problem to the computation of $G(0)$ and of $\partial G(\alpha)/\partial\alpha|_{\alpha=0}$ terms.

The first is the negative entropy of an ensemble of L uncoupled 21-states spins $A_1, \dots, A_L$ of fixed marginal probabilities $P_i(A_i)$ and has the well-known form:

$$G(0) = \sum_{i=0}^{L}\sum_{A=1}^{q} P_i(A)\ln P_i(A). \tag{4.23}$$

Deriving (4.20) with respect to α one obtains

$$\begin{aligned} \frac{\partial G(\alpha)}{\partial \alpha} &= -\sum_{\{A_i\}}\left[\sum_{i<j} e_{ij}(A_i, A_j) + \sum_i \frac{dh_i(A_i)}{d\alpha}\right]\frac{e^{-H(\alpha)}}{Z(\alpha)} \\ &\quad + \sum_i\sum_{A=1}^{q-1}\frac{dh_i(A)}{d\alpha}P_i(A). \\ &= -\sum_{\{A_i\}}\sum_{i<j}\frac{e_{ij}(A_i, A_j)e^{-H(\alpha)}}{Z(\alpha)} \\ &= \langle -\sum_{i<j} e_{ij}(A_i, A_j)\rangle_\alpha. \end{aligned} \tag{4.24}$$

Then, since for $\alpha = 0$ the joint probabilities $P_{ij}(A, B)$ factorize:

$$\begin{aligned} \frac{\partial G(\alpha)}{\partial \alpha}\bigg|_{\alpha=0} &= \langle -\sum_{i<j} e_{ij}(A_i, A_j)\rangle_\alpha \bigg|_{\alpha=0} \\ &= -\sum_{A,B}\sum_{i<j} e_{ij}(A, B) P_i(A) P_j(B). \end{aligned} \tag{4.25}$$

Putting together these two terms, and remembering that we have chosen the gauge (4.18), we can write the Gibbs potential in the mean field approximation as:

$$\begin{aligned} G(\alpha)^{CM} = \sum_{i=1}^{L} \Bigg[& \sum_{A=1}^{q-1} P_i(A) \ln P_i(A) + \\ & + \left(1 - \sum_{A=1}^{q-1} P_i(A)\right) \ln \left(1 - \sum_{A=1}^{q-1} P_i(A)\right) \Bigg] + \\ & - \left[\sum_{i<j}\sum_{A,B} e_{ij}(A, B) P_i(A) P_j(B)\right] \alpha. \end{aligned} \tag{4.26}$$

We can now use equations[6]

$$h_i(A) = \frac{\partial G(\alpha)}{\partial P_i(A)} \tag{4.27}$$

and

$$(C^{-1})_{ij}(A, B) = \frac{\partial h_i(A)}{\partial P_j(B)} = \frac{\partial^2 G(\alpha)}{\partial P_i(A) \partial P_j(B)}, \tag{4.28}$$

to obtain, within the approximation $G(\alpha) \approx G(\alpha)^{MF}$, for $\alpha = 1$, the fields and the couplings for our Potts model (4.11).

Without reporting all the calculations (standard, but quite long), making the derivatives in (4.27) and (4.28), one finds for the fields

$$h_i(A) = \ln\left(\frac{P_i(A)}{P_i(q)}\right) - \sum_{i\neq j}\sum_{C=1}^{q-1} e_{ij}(A, C) P_j(C), \tag{4.29}$$

[6]Following from the equivalent equations for the conjugate variables of $F(\alpha)$:

$$\frac{\partial \ln Z}{\partial h_i(A)} = -P_i(A);$$

$$\frac{\partial^2 \ln Z}{\partial h_i(A) \partial h_j(b)} = -P_{ij}(A, B) + P_i(A) P_j(B).$$

and, deriving again with respect to $P_j(B)$, one gets to the fundamental result for the couplings:

$$\begin{aligned}(C^{-1})_{ij}(A,B) &= \frac{\partial h_i(A)}{\partial P_j(B)} \\ &= \begin{cases} -e_{ij}(A,B) & i \neq j \\ \frac{\delta_{A,B}}{P_i(A)} & i = j \end{cases}.\end{aligned} \tag{4.30}$$

Hence, following the approach just presented, we are able to solve our complex inference problem only in one step: starting from the matrix of empirical correlations of the MSA, $C_{ij}(A,B) = f_{ij}(A,B) - f_i(A)f_j(B)$, we just need to invert it to obtain the couplings parameters $e_{ij}(A,B)$ for every couple of positions i and j and their respective states A and B.

4.1.5 Direct interaction

If one is then interested in obtaining, for fixed (i,j), a single scalar quantity DI_{ij} [3] from the $(q-1)\times(q-1)$ matrix $e_{ij}(A,B)$, representing the strength of the interaction between the two sites, one possible choice is to isolate these sites and build the two-sites model

$$P_{ij}^{dir}(A,B) = \frac{1}{Z_{ij}} \exp\left\{e_{ij}(A,B) + \tilde{h}_i(A) + \tilde{h}_j(B)\right\}, \tag{4.31}$$

where the couplings $e_{ij}(A,B)$ are the ones just inferred, Z_{ij} is a reduced partition function and the fields $\tilde{h}_i(A)$ and $\tilde{h}_j(B)$ follow from the conditions

$$f_i(A) = \sum_{B=1}^{q} P_{ij}^{dir}(A,B), \tag{4.32}$$

$$f_j(B) = \sum_{A=1}^{q} P_{ij}^{dir}(A,B). \tag{4.33}$$

DI_{ij} can be then defined as the mutual information MI_{ij} (4.7) associated to the reduced model (4.31), i.e., as:

$$DI_{ij} = \sum_{A,B=1}^{q} P_{ij}^{dir}(A,B) \ln\left(\frac{P_{ij}^{dir}(A,B)}{f_i(A)f_j(B)}\right), \tag{4.34}$$

measuring only the strength of the direct couplings and omitting any indirect effect.

4.1.6 Concluding remarks

In agreement with the assumption that the MSA sequences sample the Potts model (4.11), the couplings $e_{ij}(A, B)$ represent direct interactions between couple of amino acids and all the statistical correlations $C_{ij}(A, B)$ emerge as an effect of this direct interactions. For this reason, MI_{ij}, for some value of i and j, can have a not-negligible value even if DI_{ij} is small; by contrast for high value of DI_{ij}, we expect to have also high mutual information MI_{ij}, i.e., high correlations between the two sites.

Hence, within this theoretical framework, the parameters $e_{ij}(A, B)$ contain all the information on the system, i.e., on the protein sampled by the MSA, information hidden within the empirical correlations.

Although this method has been developed in order to infer 3D contacts between amino acids in i and j positions that show an high value of DI_{ij}, typically there are high values of DI_{ij} that do not correspond to any 3D contact of the protein [3] and so must be the results of other type of strong functional interactions, different than physical contact. As already explained, for our purpose all the high values of DI_{ij} are interesting. In fact, discrepancies between high DI_{ij} and effective i-j 3D contacts, seen as a weakness by the authors of DCA method, for us could be of remarkable importance, since an high value of interaction DI_{ij} that do not correspond to any 3D contact brings information we cannot find looking at the 3D structure of the protein, already synthesized and known.

4.2 Statistical Couplings Analysis

Statistical couplings analysis method (SCA) [2] takes as input the same quantities of DCA, i.e., the single and double sites frequencies $f_i(A)$ and $f_{ij}(A, B)$. These frequencies can be weighted using exactly the same corrections of DCA [1], so to obtain again (4.5) and (4.6). Instead of using them to compute the simple matrix of correlation $C_{ij}(A, B) = f_{ij}(A, B) - f_i(A)f_j(B)$, the main idea is to build up a conservation-weighted correlation matrix, corresponding to the classical matrix of correlations $C_{ij}(A, B)$ but rescaled by a functional of the positional conservations $D_i(A)$ and $D_j(B)$. Let us define these quantities.

4.2.1 Positional conservation

Positional conservation $D_i(A)$ measures the divergence between the observed frequency $f_i(A)$ of having amino acid A at position i from the

background frequency $q(A)$:

$$D_i(A) \equiv f_i(A) \ln \left[\frac{f_i(A)}{q(A)} \right] + [1 - f_i(A)] \ln \left[\frac{1 - f_i(A)}{1 - q(A)} \right], \tag{4.35}$$

where the background frequency $q(A)$ is the mean frequency of having A in any site of a sequence

$$q(A) \equiv \frac{1}{L} \sum_{i=1}^{L} f_i(A). \tag{4.36}$$

$D_i(A)$, as it is clear from (4.35), corresponds, in information theory, to the relative entropy between the two empirical distribution $f_i(A)$ and $q(A)$ [18], and its actual meaning as a measure of conservation can be better understood following its derivation in this particular context: under the assumption that A has independent probability $q(A)$ to appear at a site i in each of the N sequences, the probability $P_N[f_i(A)]$ of observing $f_i(A)$ in an MSA of N sequences is

$$P_N[f_i(A)] = \frac{q(A)^{Nf_i(A)}(1 - q(A))^{N(1-f_i(A))}N!}{[Nf_i(A)]![N(1 - f_i(A))]!}, \tag{4.37}$$

that for N sufficiently large, using Stirling approximation, takes the form

$$P_N[f_i(A)] \approx \left[\frac{q(A)}{f_i(A)} \right]^{Nf_i(A)} \left[\frac{1 - q(A)}{1 - f_i(A)} \right]^{N(1-f_i(A))} = e^{-ND_i(A)}, \tag{4.38}$$

where $D_i(A)$ is defined as in (4.35).

Significant frequencies $f_i(A)$ are the ones with low probability (4.38), i.e., frequencies $f_i(A)$ emerging from the background frequency $q(A)$. For these frequencies $D_i(A)$ is maximal, so that the value of $D_i(A)$ indicates how unlikely the observed frequency of amino acid A at position i would be if A occurred randomly with probability $q(A)$; for this reason $D_i(A)$ provides a definition of position-specific conservation.

4.2.2 Re-weighted correlation matrix

The conserved correlation matrix $\tilde{C}_{ij}(A, B)$ is then defined as:

$$\tilde{C}_{ij}(A, B) \equiv \phi(D_i(A))\phi(D_j(B))C_{ij}(A, B), \tag{4.39}$$

where the function $\phi(D_i(A))$ is given by

$$\phi(D_i(A)) = \frac{\partial D_i(A)}{\partial f_i(A)} = \ln \left[\frac{f_i(A)(1 - q(A))}{q(A)(1 - f_i(A))} \right]. \tag{4.40}$$

This choice for $\phi(D_i(A))$ is only one of the possible choices ([2],[6]), since DCA matrix is, in general, a weighted correlation matrix that measures the significance of amino acid correlations using the conservation of the residues involved. In fact, (4.40) has been chosen because it gives a functional that rises even more steeply than $D_i(A)$ as the frequency $f_i(A)$ approaches one, a property that reduces correlations arising from weakly conserved amino acids (since the gradient of $D_i(A)$ approaches zero as $f_i(A) \to q(A)$), and emphasizes conserved correlations.

Since $\tilde{C}_{ij}(A, B)$ matrix is a $qL \times qL$ matrix, with rows and columns that run both over site positions and possible amino acid states, and since we are interested in finding relations between sites along the sequence (and not between couples (i, A) of sites *and* amino acids showed at that sites), one can introduce the reduced $L \times L$ matrix $\tilde{C}_{ij}$:

$$\tilde{C}_{ij} \equiv \left[\sum_{A,B} \tilde{C}_{ij}(A, B)^2 \right]^{1/2} , \tag{4.41}$$

indexed only by physical positions i and j along the sequences.

4.2.3 Spectral decomposition and noise-undressing

Besides these initial definitions and the weighting choices, one can take in order to build a matrix with the correct statistics, the present method relies, from a more fundamental and physical point of view, on a simple spectral decomposition.

The idea is to diagonalize $\tilde{C}_{ij}$ matrix and to compare its eigenvalues with the ones one can obtain from a random correlation matrix, built using a random MSA generated respecting the real MSA frequencies $f_i(A)$[7]. Doing that one sees that the bulk of the $\tilde{C}_{ij}$ spectrum can be attributed to noise, since the same distribution of eigenvalues is obtained for the random matrix (the so-called Marchenko-Pastur distribution [25]). This comparison procedure of the two spectra, called noise-undressing [2], indicates in fact that only few eigenvalues (3,4 or 5; for our MSA, for example, 5) emerge from the noise bulk, i.e., have values above the cutoff established by the random matrix spectrum, and hence are really informative.

[7] A practical choice is to shuffle independently the elements along the starting MSA columns.

4.2.4 A coherent uninformative mode

Let us suppose that after the noise-undressing procedure, as in our case (see section 4.3), only 5 eigenvalues emerge from the noise bulk. Among them, the highest one, the first mode, has a distinctive property: it describes a coherent correlation of all positions, probably due to a common phylogenetic history, so that it does not carry any information about the structure of the protein and the relations between its sites.

In order to show this, one can take advantage of the fact that the first mode makes the dominant contribution to $\tilde{C}_{ij}$.

As a first order approximation, correlation matrix $\tilde{C}_{ij}$ can be written, using $S_i = \sum_j \tilde{C}_{ij}$, as:

$$\tilde{C}_{ij}^{(1)} = \frac{S_i S_j}{\sum_k S_k}, \tag{4.42}$$

a matrix that shows only one non-zero eigenvalue:

$$\lambda^{(1)} = \frac{\sum_i S_i^2}{\sum_k S_k}, \tag{4.43}$$

with an associated eigenvector $\left|\lambda^{(1)}\right\rangle$, whose components are

$$\left\langle i \,\middle|\, \lambda^{(1)} \right\rangle = \frac{S_i}{(\sum_k S_k^2)^{1/2}}, \tag{4.44}$$

where $|i\rangle$ is the vector with all but the i-th component equal to zero, so that $\left\langle i \,\middle|\, \lambda^{(1)} \right\rangle$ is the i-th component of $\left|\lambda^{(1)}\right\rangle$.

Computing (4.43) and (4.44) for an empirical $\tilde{C}_{ij}$, one can see that these expressions are in fact good approximations of the first eigenvalue and the first eigenvector of the same complete correlation matrix $\tilde{C}_{ij}$. This is due to the fact that the first mode has an high value and gives the dominant contribution to the matrix. Hence, the first eigenvector (4.44) gives the contribution of each position of the sequence to the total correlation. Since each position contributes with the same sign to that first eigenvector [2], one can conclude that it corresponds to a global, coherent mode, whose origin is purely historical. Giving no contribution to the grouping procedure of sequence sites in sectors, the first mode can be disregarded.

4.2.5 Sectors identification

Disregarding the first mode and bearing in mind that only the first five[8] modes are relevant, one can look at the correlation matrix along these relevant modes, built as a projective operator using the eigenvectors $|k\rangle$ of $\tilde{C}_{ij}$ for $k = 2, 3, 4, 5$, i.e.,

$$\tilde{\boldsymbol{C}} \simeq \sum_{k=2}^{5} \lambda_k \, |k\rangle \, \langle k| \,, \tag{4.45}$$

where λ_k is the eigenvalue relative to eigenvector $|k\rangle$ and where Dirac notation has been used for right and left eigenvectors. Following the same notation, elements of the matrix can be written as

$$\tilde{C}_{ij} \simeq \sum_{k=2}^{5} \lambda_k \, \langle i \,|\, k\rangle \, \langle k \,|\, j\rangle \,, \tag{4.46}$$

where, as before, $|i\rangle$ is the vector with all but the i-th component equal to zero, so that $\langle i \,|\, k\rangle$ is the i-th component of the k-th eigenvector, or, conversely, can be seen as the weight of position i along k mode.

Thinking to the correlation matrix in its diagonalized and relevant-modes approximation (4.46), one can define sectors as positions i having relevant weight $\langle i \,|\, k\rangle$ along a specific eigenvector.

Let us explain more in detail the procedure used to detect these positions [2], although it will be clearer looking at its application in the next section.

4.2.6 The projection procedure

Firstly one has to select, among the principal modes corresponding to eigenvectors $|2\rangle$, $|3\rangle$, $|4\rangle$ and $|5\rangle$ those couples showing more clearly the separation[9] of site weights along specific modes.

In fact, if one plot the weights along k-eigenvector $\langle i \,|\, k\rangle$ versus the weights $\langle i \,|\, k'\rangle$ along k'-eigenvector for every possible couple of $(k, k') \in (2, 3, 4, 5)$ one sees that most of positions i cluster near the origin, i.e., have components $\langle i \,|\, k\rangle$ and $\langle i \,|\, k'\rangle$ almost zero, but some of these positions form distinct groups (sectors) emerging, in the plot, along characteristic directions. The couple of modes k and k' providing the clearest basis for sector identification, i.e., whose plot shows more clearly sectors

[8] Here we are following, for simplicity, our case (results in section 4.3) as an example.

[9] See fig. 4.4.

separation, is chosen and used in the following.

Once the couple (k, k') has been selected, we can split positions along the sequence in three sectors[10]: the first is defined as the positions i for which $\langle i \,|\, k \rangle > \epsilon$ and $\langle i \,|\, k \rangle > \langle i \,|\, k' \rangle$; the second sector as those for which $\langle i \,|\, k \rangle < -\epsilon$ and $\langle i \,|\, k \rangle < -|\, \langle i \,|\, k' \rangle \,|$ and the third one as those for which $\langle i \,|\, k' \rangle > \epsilon$ and $\langle i \,|\, k' \rangle > |\, \langle i \,|\, k \rangle \,|$.

The threshold ϵ is selected in order to separate significant weights along the eigenvectors from statistical noise and follows from the comparison between components of random correlation matrix eigenvector and the real ones: plotting random eigenvector components one obtains a Gaussian distribution centered on zero with a width of approximately 2ϵ, so that only components of the actual eigenvectors whose absolute value is above ϵ are distinct from noise.

4.2.7 Concluding remarks

SCA method, as explained by its authors ([2],[1]), is an application to protein biophysics of instruments already successfully used in financial analysis in order to extract nonrandom correlations of stock performance over a finite time window [24]. Such studies show that only a small fraction of observed correlations are relevant, because most of them arise simply as a consequence of the limited period of time over which stock prices are sampled. Furthermore, as for protein sectors, the remaining significant correlations are organized in a few collective modes that decompose the economy into business sectors, group of business entities whose performance fluctuates together over time.

In [2], using S1A protein family, two main characteristics of protein *sectors* are identified: statistical independence and physical connectivity in the tertiary structure of the protein. Therefore the concept of business sectors in economy, translated in this different context, i.e., the study of coevolving elements in proteins, appears to be again meaningful.

As we have seen, DCA and SCA methods are really different in their principles as well as in their aims and results. Besides the fact that the first method is theoretically more elegant and intriguing for a statistical physicist (since it uses a Potts model distribution and a mean field approximation in order to obtain a simple one-step algorithm), it is not very versatile from a practical point of view: the starting dataset (the MSA) has to be very large and well sampled in order to infer the large number of parameters of the Potts model. Otherwise, SCA method is less ambitious but more powerful and versatile.

[10] See fig. 4.4.

In fact, analyzing influenza A HA protein, for which available sequences are less than those available for the protein families used in [3], DCA method fails, because of over-parametrization, while SCA gives some interesting preliminary results.

4.3 First results

Probably because the HA protein MSA available is not big enough, nor sufficiently diversified with respect to the bacterial protein families MSAs (upon which the method has been tested [3]), DCA algorithm, when applied to our dataset, is not able to give meaningful results. In fact, the correlation matrix $C_{ij}(A, B)$, even using pseudo-counts or different threshold x for the multiplicities definition, shows only few eigenvalues different from zero. This is due to the fact that our sequences are far fewer than qL^2. Since doing the pseudo-inverse of such a matrix, in order to search for the couplings, is meaningless, and performing a spectrum analysis is addressed by SCA method, in this final section we discuss only the application of the latter[11].

We start showing why binary approximation for SCA cannot be completely justified and hence used for our MSA (where with binary approximation we mean the translation of the original 21-alphabet MSA to a reduced one, presenting only two possible state, i.e., an Ising model) and then briefly present the first, preliminary results of SCA analysis.

4.3.1 Binary approximation discussion

Binary approximation corresponds to translate any of the N amino acid sequences of the MSA in a binary array of dimension L, showing 1 or 0 in positions where, respectively, the original sequence shows the most frequent amino acid $\tilde{A}_i$ or any other amino acid. More formally

$$I_i^a \equiv \delta_{A_i^a, \tilde{A}_i}, \tag{4.47}$$

where $\boldsymbol{I}$ is the binary MSA and a runs over the N sequences.

In 4.2.1 we introduced the positional conservation $D_i(A)$ (4.35). In fact, one can also introduce in a similar way the overall conservation D_i: in the assumption that any amino acid $A \in (1, \ldots, 21)$ has independent probability $q(A)$ to be present at site i in each of the N sequences, we

[11] The method has been implemented using a combination of C++ and *Mathematica*.

can write the probability of observing jointly at position i the amino acids frequencies $(f_i(1), \ldots, f_i(21))$ as

$$P_N[f_i(1), \ldots, f_i(21)] = \frac{N! q(1)^{N f_i(1)} \cdots q(21)^{N f_i(21)}}{(N f_i(1))! \cdots (N f_i(21))!} \approx e^{-N D_i}, \quad (4.48)$$

where now the overall conservation D_i is

$$D_i = \sum_{A=1}^{21} f_i(A) \ln \left(\frac{f_i(A)}{q(A)} \right). \quad (4.49)$$

The idea, in order to understand if binary approximation holds, is to compare the overall conservation D_i with the positional conservation $D_i(\tilde{A}_i)$ of the most frequent amino acid at position i.

As a general rule $D_i(A) \leq D_i$. Since $D_i(A)$ is maximal for $A = \tilde{A}_i$ and both $D_i(\tilde{A}_i)$ and D_i are non-linear functions of $f_i(A)$ that rise more and more steeply as $f_i(A)$ approaches one, $D_i(\tilde{A}_i)$ can, sometimes, be used as an approximation for D_i. If this is true, plotting $D_i(\tilde{A}_i)$ over D_i, one must obtains points close to the line $D_i(\tilde{A}_i) = D_i$, which justifies the use of binary approximation. Otherwise, the passage from 21-alphabet to the binary one generates a loss of conservation-related information.

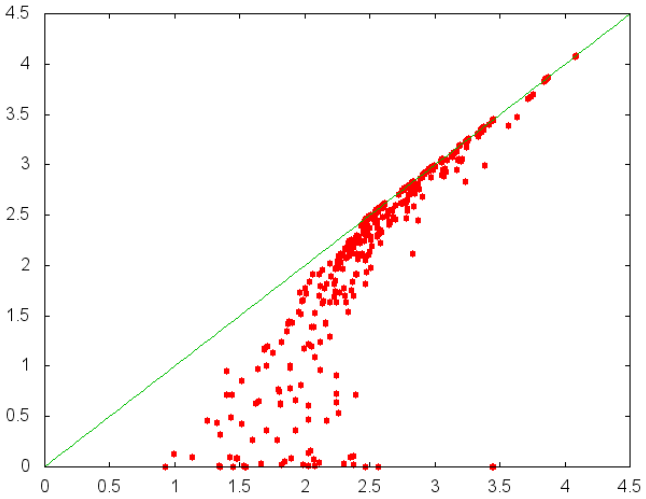

Figure 4.1: Most frequent amino acid conservation in position i, $D_i(\tilde{A}_i)$ (y-axis), versus total conservation in position i, D_i (x-axis). Especially for low conserved positions $D_i \in [1, 2.5]$, points fall far below the line $y = x$: we choose not to use binary approximation.

Our situation, in fact, is the latter, as can be argued looking at figure 4.1, especially at sites with low overall conservation $D_i \in [1, 2.5]$.

Since working with all the 21-alphabet is not problem from a computational point of view, SCA analysis, in the following, is performed without using binary approximation.

4.3.2 Influenza HA protein sectors

Conservation-weighted correlation matrix built for our HA protein MSA, the one presented in chapter 2 and whose information content has been studied in chapter 3, is shown in figure 4.2 in the reduced form of equation (4.41).

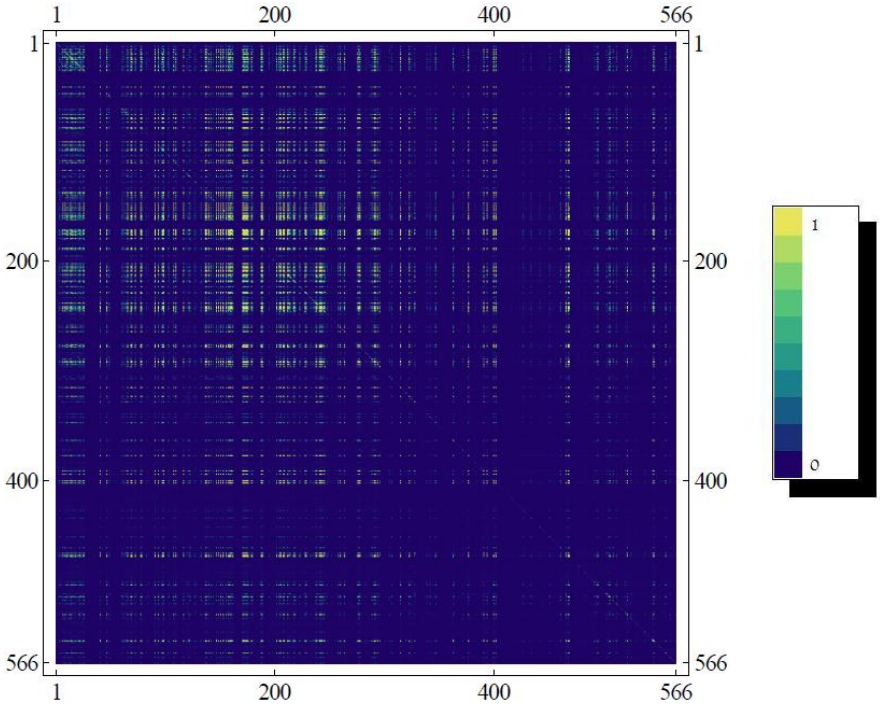

Figure 4.2: Reduced conservation-weighted correlation matrix $\tilde{C}_{ij}$ for HA protein MSA. As one can see, almost all correlations are close or equal to zero. A site with row (and hence column) completely blue, i.e., showing correlations equal to zero with respect to every other site, is a site showing always the same amino acid in all the sample, so that the pattern of horizontal (and vertical) blue lines indicates that there are many conserved positions in our sample.

The spectrum of this matrix is compared with three random correlation matrix spectra in figure 4.3, where, as already explained, a random matrix of correlation is a matrix constructed using again (4.41) but on a random MSA, built shuffling independently elements along the columns of the original MSA. From this spectra comparison one can see that only four informative eigenvalues emerge from the bulk of noisy eigenvalues (the first mode ($\lambda_1 \approx 118$) has been already removed from the plot in figure 4.3, since, following the discussion of 4.2.4, can be ignored in the sectors identification procedure).

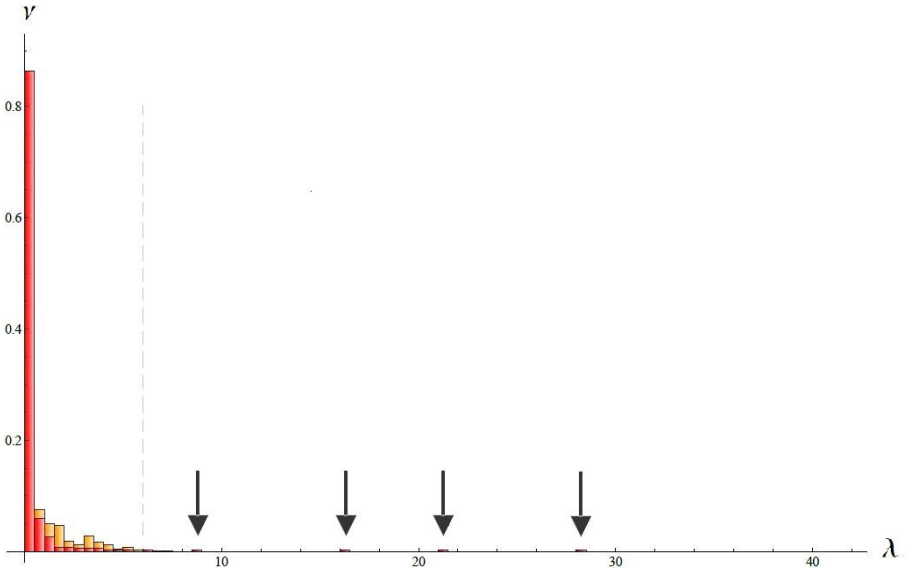

Figure 4.3: Reduced conservation-weighted correlation matrix $\tilde{C}_{ij}$ spectrum (red) compared with three random correlation matrix spectra (orange). Without taking into account the first one ($\lambda_1 \approx 118$, not shown), there are only 4 eigenvalues emerging from the bulk of the spectrum. The latter, in fact, is well reproduced by the random matrix spectrum and hence can be attributed to noise (finite-size of the sampling).

Plotting the components of the four eigenvectors associated to the other four relevant modes λ_2, λ_3, λ_4 and λ_5, as explained in 4.2.5, one finds that the couple of modes $(2, 3)$ is the one that shows more clearly sectors separation along different modes, as testified by figure 4.4.

Using these two modes, looking at figure 4.4, one can identify the first sector as the ensemble of positions i for which $\langle i \,|\, 2 \rangle > \epsilon$ and $\langle i \,|\, 2 \rangle > |\, \langle i \,|\, 3 \rangle \,|$; the second sector as the ensemble of positions i for which $\langle i \,|\, 2 \rangle < -\epsilon$ and $\langle i \,|\, 2 \rangle < -|\, \langle i \,|\, 3 \rangle \,|$; and the third sector as the en-

semble of positions i for which $\langle i\,|\,3\rangle > \epsilon$ and $\langle i\,|\,3\rangle > |\,\langle i\,|\,2\rangle\,|$. $\epsilon = 0.025$, here, is the value above which the frequency of random correlation matrix eigenvector components fall below 1% and is used here as a cutoff to distinguish relevant components $\langle i\,|\,3\rangle$ and $\langle i\,|\,2\rangle$ from the ones related to noise.

Figure 4.4 shows that while the first and the second sector sites are clearly disposed along distinct directions, the first and the third sectors have many sites close to the line $\langle i\,|\,3\rangle = \langle i\,|\,2\rangle$. Following the rules just described these sites have been forcedly collocated in one of the two sectors. However, as a consequence of the "ambiguous" positions of these sites, we expected that the first and the third sectors will be partially correlated.

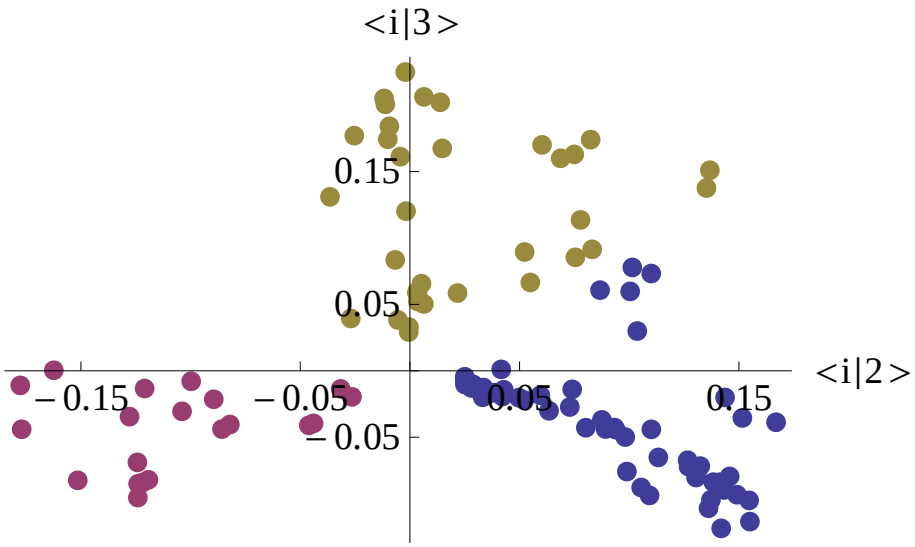

Figure 4.4: $\langle i\,|\,3\rangle$ versus $\langle i\,|\,2\rangle$ eigenvectors $|3\rangle$ and $|2\rangle$ components. Blue points represent the first sector, for which $\langle i\,|\,2\rangle > \epsilon$ and $\langle i\,|\,2\rangle > |\,\langle i\,|\,3\rangle\,|$; red points represent the second sector, for which $\langle i\,|\,2\rangle < -\epsilon$ and $\langle i\,|\,2\rangle < -|\,\langle i\,|\,3\rangle\,|$; green points represent the third sector, for which $\langle i\,|\,3\rangle > \epsilon$ and $\langle i\,|\,3\rangle > |\,\langle i\,|\,2\rangle\,|$. $\epsilon = 0.025$, as explained in 4.2.5, is the cutoff used to distinguish relevant components $\langle i\,|\,3\rangle$ and $\langle i\,|\,2\rangle$ from the ones generated by noise. While the first and the second sector sites are clearly disposed along distinct directions, the first and the third sectors have many sites close to the line $\langle i\,|\,3\rangle = \langle i\,|\,2\rangle$. Following the rules described these sites have been collocated in in one of the two sectors. However, as a consequence of the "ambiguous" positions of these sites, we expected that the first and the third sectors will be partially correlated.

The sectors so obtained are composed by, respectively, 59, 20 and 35 positions; these positions are not consecutive positions along the sequence (in fact they can be very distant one from another).

One can visualize the three sectors plotting the matrix

$$\tilde{C}^{23} = \lambda_2 |2\rangle \langle 2| + \lambda_3 |3\rangle \langle 3| \tag{4.50}$$

only for the sites belonging to the sectors, the ones showed in figure 4.4. This has been done in figure 4.5. As expected, while the first and the second sectors are completely uncorrelated one respect to the other, this is not strictly true for the first and the third.

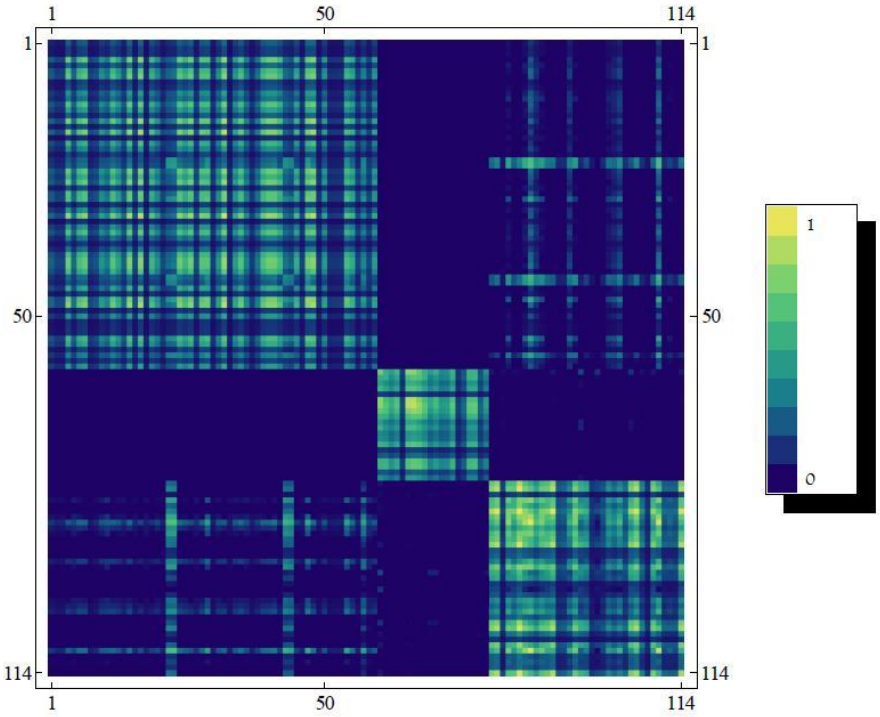

Figure 4.5: SCA matrix for HA protein MSA after removal of statistical noise, global and coherent correlations, and trimming the 114 positions that show significant weights along the eigenvectors $|2\rangle$ and $|3\rangle$. As expected looking at figure 4.4, first (green) and third (blue) sectors are not completely uncorrelated one respect to the other, while this is true for the other two couples of sectors.

Conclusions and further work

Starting from the selection of the richest sequences dataset of influenza A HA protein available at the time, the present work has achieved some interesting results.

Firstly, information content of the dataset has been investigated using theoretical instruments derived from information theory and statistical physics of complex systems. Looking how the clusters distribution of the sample behaves at different clustering scales, it has been concluded that the dataset contains enough statistics to let us infer, starting from it, the behavior of the actual system (the protein), at least partially.

This procedure, explained in chapter 3, is very general and can be applied to any data sample, simply using the clustering program written in C++ and presented in Appendix A. In fact, this program implements Affinity Propagation algorithm and computes the entropies needed for the information content discussion taking as input a very general matrix of normalized similarities $s(i, k)$ between elements of the data sample. These elements can be protein sequences, word sentences, picture (seen as arrays of pixels) and so on. One just needs to define a consistent measure of similarity between these objects, as we have done for protein sequences in (3.28).

After this preliminary study, two very different inference methods have been presented, compared and applied to the starting dataset, DCA and SCA. The results reached with the second method, despite being just preliminary, show that the direction taken is the right one and suggest the presence within HA protein of co-evolving sub-structures, the *sectors*, whose physical and biological role in HA protein has to be investigated; this investigation constitutes the natural following step of the present work.

Furthermore, as the financial origin of the SCA method testifies, these inference methods can be used in other fields different from the

biophysical one and also not belonging to physics at all. This is a consequence of the fact that these methods take as input an MSA, i.e., a rectangular array of characters built putting together N linear arrays, each one of them represents a different realization of the same object. In the present work, the object represented and studied is the HA protein of influenza A virus and the N linear arrays of letters stand for the possible amino acids sequences that represent different realization of the same HA protein. Any time one has an object that can be described, in principle, using an array of letters, one can translate a sample of realizations of this object in an MSA and so apply the methods here discussed.

Finally, since the sectors are ensemble of sites along the HA amino acids sequence showing strongly correlation within their self and almost no correlation with outer sites, one can think to build a model of dynamical evolution for the HA protein of influenza A in which sites belonging to the same sector are constrained to co-evolve. Hopefully, such a model will be able to better describe past evolution of the HA protein of influenza A virus.

Appendix A

Here we present the clustering program, written in C++, that implements Affinity Propagation algorithm and calculates the entropies (3.21) and (3.22) related to the clustering outcome.

```
#include <math.h>
#include <fstream>
#include <stdlib.h>
#include <iostream>
#include <stdio.h>
#define n 3297
#define nc 3297

double sim[n][n]={0}, ava[n][n]={0}, dist[n][n]={0},
       res[n][n]={0}, avaold[n][n]={0}, resold[n][n]={0},
       pi[n]={0}, pim[nc]={0}, molt[n]={0};
double mu, eps, damp, sum[n][n]={0},massimo,
       grad, minimo, smax,hk, hs, r;
int i, j, k, t, tmax, nclust, differ, control, dim, c,
    dim2, tot, decision, jmassimo;
int clu[n]={0}, oldclu[n]={0};

using namespace std;

int main()
{
    cout << "--------------------" << endl;
    cout << "AFFINITY PROPAGATION" << endl;
    cout << "--------------------" << endl;
    cout << "What strategy for computing mu?
              (tap 0 for minimum, 1 for direct input)"
         << endl;
    cin >> decision;
    if(decision==1){
        cout << "Tap the value for mu: ";
```

```
    cin >> mu;
}
cout << "Reading distance matrix, please wait." << endl;
//READING FILES
FILE * distance;
distance=fopen("distances.txt","r");
for(j=0;j<n;j++){
    for(i=0;i<n;i++){
    fscanf(distance, "%lf", &dist[i][j]);
    sim[i][j]=1-dist[i][j];
    }
}
fclose(distance);
FILE * mol;
mol=fopen("moltep.txt","r");
for(j=0;j<n;j++){
    fscanf(mol,"%lf",&molt[j]);
}
fclose(mol);
cout << "Controlling data..." << endl;
tot=0;
for(j=0;j<n;j++){
    for(i=0;i<n;i++){
        if(dist[i][j]==0){tot=tot+1;}
    }
}
if(tot==n){cout << "Done." << endl;}
else{cout << "ERROR!" << endl;}
if(decision==0){
    minimo=sim[0][1];
    for(j=1;j<n;j++){
        if(minimo>sim[0][j]){minimo=sim[0][j];}
    }
    for(i=1;i<n;i++){
        for(j=i+1;j<n;j++){
            if(minimo>sim[i][j]){minimo=sim[i][j];};
        }
    }
    mu=minimo;
    cout << "mu from minimum between similarities: "
         << mu << endl;
}
for(i=0;i<n;i++){sim[i][i]=mu;}
cout << "Initialising a and r..." << endl;
damp=0.90;
tmax=100000;
```

```
eps=0.0001;
grad=5;
//ALGORITHM
//Initializing responsibilities and availabilities
for(i=0;i<n;i++){
    for(j=0;j<n;j++){
        ava[i][j]=0;
        res[i][j]=0;
    }
    clu[i]=i;
}
// Starting clustering process
control=0;
cout << "Starting clustering..." << endl;
for(t=0;t<tmax and (grad/eps)>10;t++){
    for(j=0;j<n;j++){
        for(i=0;i<n;i++){
            avaold[i][j]=ava[i][j];
            resold[i][j]=res[i][j];
            sum[i][j]=ava[i][j]+sim[i][j];
        }
        oldclu[j]=clu[j];
    }
// updating responsibilities
    for(i=0;i<n;i++){
        massimo=sum[i][0];
        jmassimo=0;
        for(j=1;j<n;j++){
            if(massimo<sum[i][j]){
                massimo=sum[i][j];
                jmassimo=j;
            }
        }
        for(k=0;k<n;k++){
            if(k!=jmassimo){
            res[i][k]=damp*resold[i][k]+
            (1-damp)*(sim[i][k]-massimo);
            }
            if(k==jmassimo){
                if(k!=0){
                    massimo=sum[i][0];
                    for(j=1;j<n;j++){
                    if(massimo<sum[i][j] and j!=k){
                    massimo=sum[i][j];}
                    }
                }
```

```
                if(k==0){
                    massimo=sum[i][1];
                    for(j=2;j<n;j++){
                    if(massimo<sum[i][j]){
                    massimo=sum[i][j];
                    }
                    }
                }
                res[i][k]=damp*resold[i][k]+
                (1-damp)*(sim[i][k]-massimo);
            }
        }
    }
//updating availabilities
    for(k=0;k<n;k++){
        smax=0;
        for(j=0;j<n;j++){
            massimo=0;
            if(massimo<res[j][k] and j!=k){
            massimo=res[j][k];
            }
            smax=smax+massimo;
        }
        for(i=0;i<n;i++){
            minimo=0;
            if(i!=k){smax=smax-res[i][k];}
            if(minimo>(res[k][k]+smax)){
            minimo=res[k][k]+smax;
            }
            if(k!=i){
            ava[i][k]=damp*avaold[i][k]+(1-damp)*minimo;
            }
            if(k==i){
            ava[i][k]=damp*avaold[i][k]+(1-damp)*smax;
            }
        }
    }
    for(i=0;i<n;i++){
        k=0;
        massimo=ava[i][0]+res[i][0];
        for(j=1;j<n;j++){
            if(massimo<ava[i][j]+res[i][j]){
                massimo=ava[i][j]+res[i][j];
                k=j;
            }
        }
```

```
            clu[i]=k;
        }
        grad=0;
        for(i=0;i<n;i++){
            for(j=0;j<n;j++){
                grad=grad+(ava[i][j]-avaold[i][j])*
                (ava[i][j]-avaold[i][j])+
                (res[i][j]-resold[i][j])*
                (res[i][j]-resold[i][j]);
            }
        }
        grad=sqrt(grad)/n;
    // controlling advance
        differ=0;
        for(i=0;i<n;i++){
            if(clu[i]!=oldclu[i]){differ=differ+1;}
        }
        cout << "differ: " << differ << " grad/eps: "
             << grad/eps << " tempo: " << t << endl;
    }
    //END OF THE ALGORITHM
    //Statistics and outputs
    ofstream results;
    ofstream results2;
    ofstream results3;
    ofstream results4;
    results.open("clusters.txt");
    results2.open("numOnsize.txt");
    results3.open("numOnsizeMolt.txt");
    results4.open("sim.txt");
    nclust=0;
    //number of clusters
    for(i=0;i<n;i++){
        if(clu[i]==i){nclust=nclust+1;}
    }
    results << "# Number of clusters: " << nclust << endl;
    cout << "Number of clusters: " << nclust << endl;
    results << "Exemplar | size cluster |
                size cluster with molt | " << endl;
    for(i=0;i<n;i++){
        dim=0;
        dim2=0;
        control=0;
        for(k=0;k<n;k++){
            if(clu[k]==i){
                dim=dim+1;
```

```
                dim2=dim2+molt[k];
                control=1;
            }
            results4 << sim[k][i] << "\t";
        }
        results4 << endl;
        if(control==1){
            results << i << "\t" << dim << "\t"
                    << dim2 << endl;
            pi[dim]=pi[dim]+1;
            pim[dim2]=pim[dim2]+1;
            if(clu[i]!=i){cout << "Error! Sequence: "
                              << i << endl;}
        }
    }
    results2 << "# Size | Num of clusters" << endl;
    results3 << "# Size | Num of clusters
                (counting multiplicities)" << endl;
    for(i=0;i<nc;i++){
        if(i<n){
            if(pi[i]!=0){
            results2 << i << "\t" << pi[i] << endl;
            //r=i;
            //hs=hs-r*pi[i]/n*log(r/n);
            //hk=hk-r*pi[i]/n*log(pi[i]*r/n);
            }
        }
        if(pim[i]!=0){
            results3 << i << "\t" << pim[i] << endl;
            r=i;
            hs=hs-r*pim[i]/nc*log(r/nc);
            hk=hk-r*pim[i]/nc*log(pim[i]*r/nc);
        }
    }
    results3 << endl << "# mu | hs | hk | nclust
                    | n | nc" << endl;
    results3 << mu << "\t" << hs << "\t" << hk << "\t"
             << nclust << "\t" << n << "\t" << nc;
    results.close();
    results2.close();
    results3.close();
    results4.close();
    return 0;
}
```

Appendix B

As already said, pseudo-counts λ added to the empirical frequency counts in (4.5) and (4.6) can be seen as extra observations added to the real ones in order to increase the size of the dataset. Let us show more in detail what does it means.

Suppose to have N sequences. Among those, n_A show amino acid A at some specific i position, where, as before, $A \in (1, \ldots, q)$. If plenty of data are available, i.e., if N is large enough, the probability θ_A to have a sequence showing at position i amino acid A is well represented by the observed frequencies:

$$\theta_A \approx \frac{n_A}{N}, \tag{4.51}$$

with $N = \sum_A n_A$. This is the maximum likelihood solution.

For small values of N, this is no longer true. If, for example, $N = 2$, with $n_1 = 2$ and $n_A = 0\ \forall A \neq 1$, one cannot say that the probability of having amino acid $A = 3$ at position i is zero. In a similar case, one would like to assign some probability to the other residues and not rely entirely on so few observations. Since there are no more observations, these probabilities must be determined from prior knowledge. This can be done via Bayesian statistics.

The idea is to choose for the *a priori* distribution of the probabilities θ_A the Dirichlet distribution

$$D(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \frac{1}{Z(\boldsymbol{\alpha})} \prod_{A=1}^{q} \theta_A^{\alpha_A - 1}, \tag{4.52}$$

where α_A are constant parameters that characterize the distribution, $Z(\boldsymbol{\alpha})$ is the normalizing factor

$$Z(\boldsymbol{\alpha}) = \int \prod_{A=1}^{q} \theta_A^{\alpha_A - 1} d\theta = \frac{\prod_{A=1}^{q} \Gamma(\alpha_A)}{\Gamma(\sum_{A=1}^{q} \alpha_A)}, \tag{4.53}$$

and θ_A, being probabilities, respect the conditions $0 \leq \theta_A \leq 1$ and $\sum_A \theta_A = 1$.

Then, for given $\boldsymbol{\theta}$, the probability of getting $\boldsymbol{n} = (n_1, \ldots, n_q)$ counts is described by the multinomial distribution

$$P(\boldsymbol{n}|\boldsymbol{\theta}) = \frac{1}{M(\boldsymbol{\theta})} \prod_{A=1}^{q} \theta_A^{n_A}, \tag{4.54}$$

where the normalizing factor $M(\boldsymbol{\theta})$ is

$$M(\boldsymbol{\theta}) = \frac{\prod_A n_A!}{(\sum_A n_A)!}. \tag{4.55}$$

We can now use Bayes theorem

$$P(\boldsymbol{\theta}|\boldsymbol{n})P(\boldsymbol{n}) = P(\boldsymbol{n}|\boldsymbol{\theta})D(\boldsymbol{\theta}|\boldsymbol{\alpha}), \tag{4.56}$$

to write

$$\begin{aligned} P(\boldsymbol{\theta}|\boldsymbol{n}) &= \frac{P(\boldsymbol{n}|\boldsymbol{\theta})D(\boldsymbol{\theta}|\boldsymbol{\alpha})}{P(\boldsymbol{n})} \\ &= \frac{\prod_A \theta_A^{n_A+\alpha_A-1}}{M(\boldsymbol{n})P(\boldsymbol{n})Z(\boldsymbol{\alpha})} \\ &= \frac{Z(\boldsymbol{n}+\boldsymbol{\alpha})}{M(\boldsymbol{n})P(\boldsymbol{n})Z(\boldsymbol{\alpha})} D(\boldsymbol{\theta}|\boldsymbol{n}+\boldsymbol{\alpha}), \end{aligned} \tag{4.57}$$

where the last two equalities follow from the definitions (4.52) and (4.54).

Since all the distribution introduced are normalized, we must have

$$\frac{Z(\boldsymbol{n}+\boldsymbol{\alpha})}{M(\boldsymbol{n})P(\boldsymbol{n})Z(\boldsymbol{\alpha})} = 1, \tag{4.58}$$

so that the posterior distribution of $\boldsymbol{\theta}$ is again, as the prior one, a Dirichlet distribution

$$P(\boldsymbol{\theta}|\boldsymbol{n}) = D(\boldsymbol{\theta}|\boldsymbol{n}+\boldsymbol{\alpha}), \tag{4.59}$$

but with parameters $\alpha_A + n_A$.

Using this distribution, one can compute the posterior mean values for θ_A:

$$\begin{aligned} \langle\theta_A\rangle &= \int \theta_A D(\boldsymbol{\theta}|\boldsymbol{n}+\boldsymbol{\alpha})d\theta \\ &= \frac{1}{Z(\boldsymbol{n}+\boldsymbol{\alpha})} \int \theta_A \prod_B \theta_B^{n_B+\alpha_B-1} d\theta. \end{aligned} \tag{4.60}$$

Then, bringing θ_A inside the product, and using the vector $\boldsymbol{\delta^A}$ ($\delta_A^A = 1$, $\delta_B^A = 0$ $\forall\ B \neq A$), we write

$$\begin{aligned} \langle\theta_A\rangle &= \frac{Z(\boldsymbol{n}+\boldsymbol{\alpha}+\boldsymbol{\delta^A})}{Z(\boldsymbol{n}+\boldsymbol{\alpha})} \\ &= \frac{\prod_B \Gamma(n_B+\alpha_B+\delta_B^A)}{\Gamma[\sum_B(n_B+\alpha_B+\delta_B^A)]} \cdot \frac{\Gamma[\sum_B(n_B+\alpha_B)]}{\prod_B \Gamma(n_B+\alpha_B)} \\ &= \frac{\prod_B(n_B+\alpha_B+\delta_B^A-1)!}{[\sum_B(n_B+\alpha_B+\delta_B^A)-1]!} \cdot \frac{[\sum_B(n_B+\alpha_B)-1]!}{\prod_B(n_B+\alpha_B-1)!} \\ &= \frac{n_A+\alpha_A}{\sum_B n_B+\alpha_B} \\ &= \frac{n_A+\alpha_A}{N+\lambda}, \end{aligned} \tag{4.61}$$

where (4.53) has been used and where $\lambda = \sum_A \alpha_A$.

Hence, considering prior knowledge is like adding extra observations (α_A) to the real ones (n_A). In (4.5) and (4.6) we used $\alpha_A = \alpha = \lambda/q$ for every amino acid A.

Bibliography

[1] O. Rivoire, *Elements of coevolution in biological sequences*, Physical Review Letters, 110 (17), p. 178102 (2013).

[2] N. Halabi, O. Rivoire, S. Leibler, R. Ranganathan, *Protein Sectors: Evolutionary Units of Three-Dimensional Structure*, Cell, 138, p. 774-786 (2009).

[3] F. Morcos, A. Pagnani, B. Lunt, A. Bertolino, D.S. Marks, C. Sander, R. Zecchina, J.N. Onuchic, T. Hwa, and M. Weigt, *Direct-coupling analysis of residue coevolution captures native contacts across many protein families*, Proc. Natl. Acad. Sci. U.S.A. 108, p. E1293-E1301 (2011).

[4] M. Ekeberg, C. Lövkvist, Y. Lan, M. Weigt, E. Aurell, *Improved contact prediction in proteins: Using pseudolikelihoods to infer Potts models*, Phys. Rev. E, 87, p. 012707 (2013).

[5] M. Weigt, R.A. White, H. Szurmant, J.A. Hoch, and T. Hwa, *Identification of direct residue contacts in protein-protein interaction by message passing*, Proc. Natl. Acad. Sci. U.S.A., 106, p. 67-72 (2009).

[6] S.W. Lockless, R. Ranganathan, *Evolutionarily conserved pathways of energetic connectivity in protein families*, Science, 286, p. 295-299 (1999).

[7] M. Marsili, I. Mastromatteo, Y. Roudi, *On sampling and modeling complex systems*, J. Stat. Mech, P09003 (2013).

[8] D.J.C. Mackay, *Information theory, inference, and learning algorithms*, Cambridge University Press (2003).

[9] T. Plefka, *Convergence condition of the tap equation for the infinite-ranged ising spin glass model*, Journal of physics A: mathematical and general, 15(6), p. 1971 (1982).

[10] A Georges and J.S.Yedidia, *Howe to expand around mean-field theory using high-temperature expansions*, Journal of physics A: mathematical and general, 24(9), p. 2173 (1991).

[11] E.T. Jaynes, *Information theory and statistical mechanics*, Phys. Rev., 106, p. 620-630 (1957).

[12] E. T. Jaynes, *Theory and statistical mechanics, II*, Phys. Rev., 108, p. 171-190 (1957).

[13] B.J. Frey, D. Duek, *Clustering by passing messages between data points*, Science, 315, p. 972-976 (2007).

[14] J. Galambos, *The asymptotic theory of extreme order statistics*, John Wiley Ed., New York (1978).

[15] T.M. Cover, J.A. Thomas, *Elements of information theory*, John Wiley Ed., New York (1991).

[16] M. Mézard, A. Montanari, *Information, Physics, and computation*, Oxford University Press, Oxford (2009).

[17] F.R. Kschischang, B.J. Frey, H.-A. Loeliger, *Factor Graphs and the Sum-Product Algorithm*, IEEE transactions on information theory, 47 (2), p. 498-519 (2001).

[18] T.M Cover, J.A Thomas, *Elements of Information Theory*, Wiley Interscience (2006).

[19] M. Mézard, *Passing messages between disciplines*, Science, 301, p. 1685-1686 (2003).

[20] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, *Biological sequence analysis: probabilistic models of proteins and nucleic acids*, Cambridge University Press (1998).

[21] J. MacQueen, in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, p. 281-297 (1967).

[22] R.C. Edgar, *MUSCLE: multiple sequence alignment with high accuracy and high throughput*, Nucleic Acids Res. 32(5), p. 1792-1797 (2004).

[23] R.C. Edgar, *MUSCLE: a multiple sequence alignment method with reduced time and space complexity*, BMC Bioinformatics, 5, p. 113 (2004).

[24] J.P. Bauchaud, M. Potters, *Theory of financial risk and derivative pricing; from statistical physics to risk management*, Cambridge University Press, Cambridge (2004).

[25] V.A. Marchenko, L.A. Pastur, *Distribution of eigenvalues for some sets of random matrices*, Mat. Sb. (N.S.), 72(114), 4, p. 507-536 (1967).

[26] N. Strelkowa, M. Lässig, *Clonal interference in the evolution of influenza*, Genetics, 192, p. 671-682 (2012).

[27] R.A. Neher, B.I. Shraiman, *Competition between recombination and epistasis can cause a transition from allele to genotype selection*, Proc. Natl. Acad. Sci. USA, 106, p. 6866-6871 (2009).

[28] K. Koelle, S. Cobey, B. Grenfell, M. Pascual, *Epochal evolution shapes the phylodynamics of interpandemic influenza A (H3N2) in humans*, Science, 314, P. 1898-1903 (2006).

[29] A.C-C. Shih, T-C. Hsiao, M-S. Ho, W-H. Li, *Simultaneous amino acid substitutions at antigenic sites drive influenza A hemagglutinin evolution*, Proc. Natl. Acad. Sci. USA, 104(15), p. 6283-6288 (2007).

[30] D.J. Smith, A.S. Lapedes, J.C. de Jong, T.M. Bestebroer, G.F. Rimmelzwaan et al., *Mapping the antigenic and genetic evolution of Influenza virus*, Science, 305, p. 371-376 (2004).

[31] R.G. Webster, W.J. Bean, O.T. Gorman, T.M. Chambers, Y. Kawaoka, *Evolution and Ecology of Influenza A Viruses*, Microbiological reviews, 56(1), p. 152-179 (1992).

[32] K.G. Nicholson , R.G. Webster, A.J. Hay, *Textbook of Influenza*, Blackwell Science, Oxford (1998).

[33] R.A. Lamb, R.M. Krug, *Orthomyxoviridae: The viruses and their Replication*, from *Fields Virology* (4th edition; edited by D.M Knipe, P.M Howley) Lippincott & Co., Philadelphia, p. 1487-1531 (2001).

[34] R.A. Lamb, *Genes and proteins of the influenza viruses*, from *The influenza viruses* (edited by R.M. Krug, H. Fraenkel-Conrat and R.R. Wagner), Plenum Press, New York, p. 1-88 (1989).

[35] T. Samji, *Influenza A: Understanding the Viral Life Cycle*, Yale J Biol Med, 82(4), p. 153-159 (2009).

[36] V.S. Hinshaw, R.G. Webster, B. Turner, *The perpetuation of orthomyxoviruses and paramyxoviruses in Canadian waterfowl*, Can. J. Microbiology, 26, p. 622-629 (1992).

[37] R.G. Webster, *Are equine 1 influenza viruses still present in Horses?*, Equine Vet. J., 25, p. 537 (1993).

[38] B.S. Kamps, C. Hoffmann, W. Preiser, *Influenza Report 2006*, Flying Publisher, Paris (2006).

[39] P.R. Murcia, J.L.N. Wood, E.C. Holmes, *Genome-scale evolution and Phylodynamics of Equine H3N8 Influenza A Virus*, j. of Virology, 85(11), p. 5312-5322 (2011).

[40] K.D. Patterson, G.F. Pyle, *The geography and the mortality of the 1918 influenza pandemic*, Bulletin of the History of medicine, 95(1), p. 4-21 (1991).

[41] Y. Bao, P. Botolov, D. Dernovoy *et al.*, *The Influenza Virus Resurce at the National Center for Biotechnology Information*, J. of Virology, 82(2), p. 596-601 (2008).

[42] *NCBI Influenza Virus Resurce* http://www.ncbi.nlm.nih.gov/genomes/-FLU/FLU.html

[43] *World Health Organization* http://www.who.int/influenza

Acknowledgments

I would like to express sincere gratitude to my advisors Prof. Silvio Franz and Prof. Amos Maritan for their encouragement and vast knowledge, and to PhD student Silvia Grigolon, who made this long and difficult journey with me and without which this work would simply not have been the same.

I also wish to thank my family, my girlfriend Chiara and all my friends for their continuous support, patience and love.