Application of the Google matrix methods for characterization of directed networks

Vivek Kandiah

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *13/10/2014* **par :**

Vivek KANDIAH

Application of the Google matrix methods for characterization of directed networks

---

JURY

Sergey Dorogovtsev	Rapporteur
Bertrand Jouve	Rapporteur
Andreas Kaltenbrunner	Examinateur
Xavier Bressaud	Président du Jury
Klaus Frahm	Invité
Bertrand Georgeot	Directeur de thèse
Dima Shepelyansky	Directeur de thèse

École doctorale :
Sciences de la matière
Unité de Recherche :
Laboratoire de Physique Théorique de Toulouse
Directeur(s) de Thèse :
Bertrand Georgeot et *Dima Shepelyansky*

Acknowledgements

I am thankful to Sergey Dorogovtsev and Bertrand Jouve for accepting to be the referees and Andreas Kaltenbrunner and Xavier Bressaud for accepting to be part of the jury.

I am also immensely grateful to my thesis supervisors Dima Shepelyansky and Bertrand Georgeot who form quite an unusual pair. Indeed the former one brings the rigorous guiding line and the hard working spirit in the Russian style while the latter one brings the flexibility and delicacy of the French style. The constructive interference of both minds provided me with a well balanced environment from which I have learnt a lot. I appreciated all the advices and concern that Dima showed me with respect to my personal situation and I greatly enjoyed the stories, the discussions and the jokes that Bertrand shared with us.

A special thanks goes to Klaus Frahm who speaks the language of machines, his incredible knowledge is only matched by his enthusiasm to explain.

I am thankful to Clément Sire, the director of the LPT, for the warm welcome in the lab and to the other permanent members with whom I had the opportunity to chat. Let us not forget the people who helped me in every other aspects during these years, among them : Malika Bentour, who took care of the numerous adminitrative headaches and Sandrine Le Magoarou who helped me with many things and introduced me to linux.

The nice atmosphere in the lab is partly due to the other students from the lab, those who were there before me : Sylvain, Lorand, Vincent, Anil, Philippe, Michael and those who joined with me or later : Juan-Pablo, julien, Mehda, Guillaume and also Lionel and Nader (with whom I had many memorable exchanges). And of course Xavier and François who are really good friends helping me whenever needed.

I am very grateful to Young-Ho Eom with whom I spent a lot of time talking about various topics of network science but also various topics of life in general. Leonardo Ermann who provided me with some insights and interesting scientific discussions.

A special thanks to Olivier Giraud who is not only a colleague from Paris but also a kind of spiritual master who tirelessly questions everything. He also introduced me to classical music and participated in my discovery of Toulouse.

Thanks to M. Hubert Escaith, I had the opportunity to spend some time in the World Trade Organization in Geneva where I learnt a lot about the international trades and the related issues.

I also thank my friends who supported me and finally I cannot thank my mother enough for having given me so much with so little and has always done the best for me even during the worst times of her life.

As a final thought, here is one of the jokes told by Bertrand which automatically pops up at this moment when I am writing this section. *Question : How much time do you need to write a thesis ? Answer : ... more time !*

Contents

Chapter 1

Introduction

1.1 What is a Network ?

When asked "What does the word *Network* means to you ?" the first thoughts that come to people's mind are the World Wide Web and their social network of acquaintances. These ideas are naturally related to the society we are living in, where these concepts are strongly present in our day-to-day life. In fact behind these concrete examples there is a general intuitive idea that a network is made of some objects called *nodes* or *vertices* that have a relationship between themselves represented by bonds called *links* or *edges*. In the literature there are two equivalent terminologies depending on the field : *vertex* and *edge* are more likely to be found in mathematics and computer science when dealing with theoretical objects, *node* and *link* are often used in physics when describing real systems. We will use the latter terminology from the next section on, in other words a network is a collection of nodes that are linked together. The number N of nodes will be referred to as the size of the network and we will restrict ourselves to the simplest case of fixed size network with fixed number of links.

Despite the modern connotations to the concept of networks, the origin of this notion dates back to the XVIII$^{\text{th}}$ century with the famous Swiss mathematician Leonhard Euler who is believed to be the first to have mathematically treated a problem under a network perspective [Euler, 1736]. In mathematics, networks are called *graphs* and a formal definition of a graph G is given by the pair $G = (V, E)$ where V is the set of vertices and E is a set of edges that connect pairs of vertices.

The story goes that the old town of Königsberg (Kaliningrad) was build around two islands on the Pregel river which were connected to each other and to the riverside by seven bridges. The question was to know whether it was possible to walk around the town from any location and visit all the bridges only once, and get back to the departure location. Already at that time the solution to this problem involved the notion of paths in a graph and a careful investigation of its topological structure. Since the 1950s, thanks to Paul Erdős and Alfréd Rényi who developed the random graph model, graph theory flourished as a field of mathematics and numerous outstanding results where established regarding structural properties of various kind of graphs [Erdős, 1959, Erdős, 1960]. Later the same can be said about complex networks as a field of physics where important contributions were brought by several great physicists [Albert et al., 1999, Albert and Barabási, 2002, Dorogovtsev et al., 2008].

Due to the richness of graph theory, a comprehensive introduction to network science is out of the scope here. We will thus introduce a few basic concepts that are sufficient for understanding the whole thesis[1].

[1] Readers interested in a more detailed introduction to complex network theory from a physics approach are encouraged to go through [Dorogovtsev, 2010] and [Dorogovtsev and Mendes, 2003] which inspired this chapter.

Directed Networks

If unspecified, a link connecting two nodes is generally considered to be a simple bond between two vertices. However it is possible to assign a direction to the link giving a new perspective to the relationship among the nodes. With directionality we now have nodes pointing to other nodes therefore we can talk about two classes of links : *ingoing links* which are the links entering a node and *outgoing links* which are those getting out of a node. Of course every outgoing link is an incoming link for a different node and a directed network can in principle also have some undirected edges which are technically nothing more than a pair of nodes pointing to each other. Nodes pointing to themselves are also possible in directed networks, they form what we call *loops*.

In some cases it is sufficient and easier to consider undirected networks but the directionality adds more interesting information on the structural organization of a system provided that we find a proper meaning to the unidirectional edges. Moreover some systems are so naturally approachable from a directed network point of view that discarding the directions of links might result in a great loss of information or even lead to meaningless conclusions as the nature of the relationship between the nodes is fundamentally different from undirected bonds. For instance *Internet* and *WWW* are often mistakenly used interchangeably, in reality the former one is an undirected network comprised of millions of interconnected computers and the latter one is a way of accessing a collection of documents built on the Internet and is, as such, a directed network. Directionality adds up more complications to the network and a naive extension of results from undirected case to the directed one is often very difficult making the study of these networks a true challenge [Leicht and Newman, 2008].

In this thesis we will focus only on directed networks thanks to the tools coming from the Google matrix theory.

Weighted Networks

We can also assign a weight to a link whether it is directed or undirected. When a given pair of nodes has n links of the same type connecting them together, we can instead consider that they are tied by one link of weight n and the number of times the link is repeated is sometimes referred to as the *multiplicity* of the link.

Weighted networks bring an other kind of information compared to their unweighted counterpart, indeed it is very important to distinguish a system where the links describe existing bonds from a system where the links describe how tightly nodes are tied together.

When the number of connections of a node is normalized to one, the weight can be interpreted as the probability or percentage of connection of this node. In directed networks we then have weights or probabilities assigned to incoming links and outgoing links separately.

In-Degree and Out-Degree

The *degree* of a node, also called its *connectivity*, describes simply how many links it has in total, in other words a node of degree k means that it participates in k connections. If the multiplicity of the links are not taken into account, the degree of a node also indicates the number of its direct neighbours. This concept can be straightforwardly extended to directed networks where a node has an in-degree value k_{in} for the number of incoming links and an out-degree value k_{out} for the number of outgoing links.

A network of N nodes has therefore a set of N degree values ($2N$ for the directed case) and we can wonder how those values are distributed : The *degree distribution* $p(k)$ of a given network indicates the probability that a randomly chosen node possesses k connections, it is therefore a crucial quantity that describes the structure of the network on a statistical level.

The in-degree distribution $p^{in}(k_{in})$ and out-degree distribution $p^{out}(k_{out})$ are similarly defined for the directed case.

Path length

The concept of a path in a graph is as intuitive as it sounds : suppose we have a simple undirected graph from where we pick two vertices A and B, a path of length l is a sequence of l edges that brings us from a node A to a node B. It is also alternatively viewed as a generalization of the degree of a node in the sense that the degree only considers the number of direct neighbours and the path also considers the second nearest neighbours, the third, and so on. This notion is in principle considered for undirected networks where the important quantities are the shortest path lengths connecting two randomly chosen vertices in a given network.

The distribution $p(l)$ of these lengths is very informative about the structural properties of a graph. The typical distance separating two nodes in the sense of number of steps needed to reach node B from node A is given by the average shortest path length $\bar{l}$.

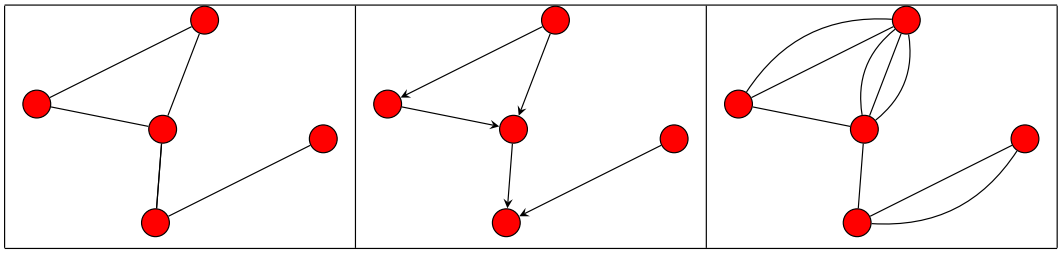

Figure 1.1: Illustrative examples of an undirected network, a directed network and a weighted network respectively.

1.2 From networks to complex networks

Thanks to the classical random graph model the structural properties, various characteristic quantities and even mechanism of network growth have been extensively studied analytically which required the graph to be simple enough to be handled rigorously. Most of the results were thus produced for undirected graphs such as simple graphs (graphs without multiplicity of links and without loops), regular graphs (graphs with same degree for all its vertices so that the degree distribution is a Dirac delta), tree graphs (graphs in which any pair of vertices is connected by a single unique path), complete graphs (graphs where each node is linked to every other node) or random graphs.

The random graph models are a statistical ensemble of all possible graphs that can be built with specific constraint such as a fixed number of nodes N or a fixed number of links L. The network is constructed by randomly assigning links to pairs of nodes, without entering into the details of these models, it is essential to note that as a result of this process the degree distribution $p(k)$ follows a binomial law. Since the binomial law converges to Poisson law in the limit of large numbers, the degree distribution $p(k)$ of a random graph will tend to a Poisson distribution $p(k) = e^{-\bar{k}}\bar{k}^k/k!$ in the limit of large network size with $\bar{k}$ being the average degree of a node.

In the late 1990s a revolution took place among the physicists in the network field when people started to study empirically real world networks such as the Internet and webpages networks [Albert et al., 1999]. It turned out that the degree distribution in those networks followed a power law $p(k) \propto k^{-\gamma}$ with the decay exponent being in the typical range of values $2 \leq \gamma \leq 3$. Instead of a rapidly decaying distribution we have the so called fat-tailed distribution. This unexpected observation triggered a lot of interest towards real-world networks rather than theoretical graphs and a great deal of different networks were found to be consistent with a fat-tailed distribution leading the community to investigate more deeply the topological properties of such systems[Caldarelli, 2007].

However because of statistical fluctuations it is difficult to assess a power law distribution on small networks, therefore it is safer to assume a power-law tendency in some cases.

Contrasting with the random graphs where the typical measure is the average degree of node, the networks following a power law distribution do not have a natural scale, hence the name *scale-free* networks. This structural difference impacts drastically the behaviour and the organization of such networks because of a large variety of node degrees and because of the presence of few crucial nodes called *hubs* which are a small number of vertices with a very high degree.

Fortunately already in the 1980s people started to push the mathematical model of random graph further, known as the *configuration model*, by generalizing it to an arbitrary degree distribution of nodes and gave one of the possible recipe to build such a random graph [Bollobás, 1980]. This time to create one instance of the statistical ensemble we have to consider a set of numbers of nodes $\{N(q)\}$ of degree q and attach q half-edges to each node and then randomly connect them by pairs until no more half-edges are left alone. This process recreates the classical random graph ensemble if the values $\{N(q)\}$ are drawn from a Poisson distribution and generates *uncorrelated* scale-free networks if the distribution used is a power law. Uncorrelated means that non trivial preferences of association between high degree nodes or a high and a low degree nodes are not captured by this model but still several features can be qualitatively explained only by the degree distribution.

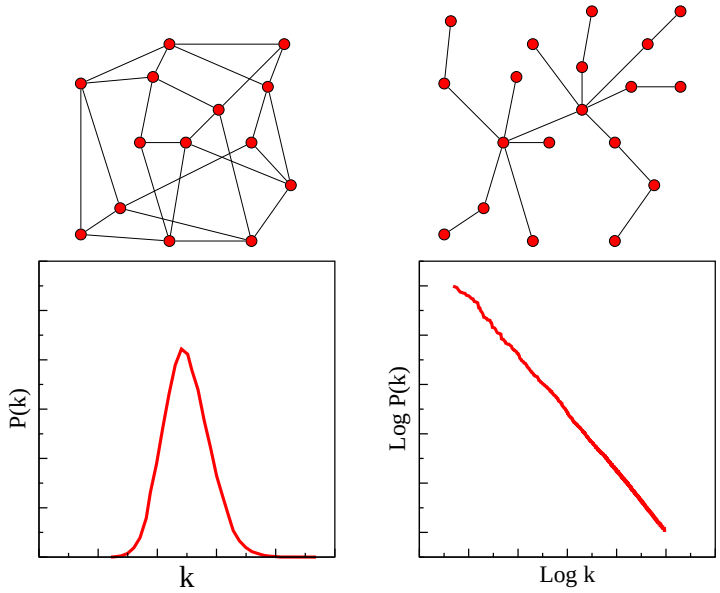

Figure 1.2: Illustrative examples of both types of graphs with their degree distribution below : a classical random graph (left) and a scale-free graph (right).

In addition to that improvement, to explain why such networks occur naturally in so many different systems Barabási and Albert proposed a simple and elegant mechanism of scale-free network formation and growth that is known today as the *preferential attachment* model and is generally considered as the most likely reason behind the structural organization of most real world networks [Barabási and Albert, 1999]. The idea is that from an initial small set of nodes we add one by one a new node which will be connected according to a specific rule so that when the number of nodes becomes larger and larger the degree distribution of the network tends to a power law. At a given time step the new added node is linked to an already existing node with a probability proportional to that node's degree. The higher a node degree is the more it attracts new links from newly added nodes leading to the formation of hub like structures which in turn become "centers" around which the network grows, hence the term "preferential".

The second major empirical observation was the measure of the compactness of real-world

networks [Watts and Strogatz, 1998]. The average shortest path length $\bar{l}$ turned out to be quite a small number in comparison with the size of the network considered. This fact was expected but the idea that one can typically navigate in a huge network in a very limited number of steps is surprising. This feature is characterized by a logarithmic dependence of the average path length with the network's size $\bar{l} \propto logN$ and is called the *small-world* effect which was nicely highlighted in an original social experiment conducted by Stanley Milgram in 1967 [Milgram, 1967].

Milgram wanted to study how closely people are related in the social network of acquaintances in the united states of America : He chose random people in the city of Omaha and a target man living in Boston. He gave letters to the people living in Omaha with the instruction that they should send the letter to the target man if they knew him directly or send it to someone, a messenger, who they think should be the most likely able to reach the target man but with the condition that they should know the messenger on a personal basis. After some time, some of the letters reached their destination and thanks to some tracking procedure Milgram observed that on average the letters went through five people before reaching the target man, hence the famous slogan "six degrees of separation" explaining that on average anyone is quite close to anyone else even in a large population. This experiment has been widely criticized for lack of rigorous protocols and weak statistical significance, nevertheless it succeeded in capturing the essence of the small-world effect in an unexpected and funny way. The origin of the effect lies in the existence of links connecting distant parts of the network producing effective shortcuts in the overall organization of the nodes. In the scale-free networks that are also small-world the hubs are playing the roles of shortcut relays, effectively reducing the length needed to cross the network. On the contrary the networks that have specific constraint so that long distance connexions are impossible, due to the geographical distance in road network for instance, do not exhibit small-world properties.

	scale-free	small world	directed
tree graph			
Avian Influenza outbreaks	✓		
Brainstem reticular formation		✓	
C.elegans interactome	✓	✓	
World Wide Web	✓	✓	✓

Table 1.1: Examples of networks that have been shown to be consistent with the three different specificities [Small et al., 2008, Humphries et al., 2006, Li et al., 2004].

It is clear that real-world complex networks show some common structural properties and non trivial behaviour which make them fundamentally different from classical random graphs. The massive shift of interest in the study of the former type of networks is understandable when we think about the wide variety of phenomenon that can be viewed as a system comprising nodes and links. Indeed the network approach offers the right amount of compromise between generalization and specificity, that is abstracting the actual objects represented by the nodes in order to find common features among globally different systems while keeping the complexity of the interactions and relationships between the objects. The possibilities of such an approach span several scales and areas and we can define networks for situations as diverse as in biology (gene regulation, protein interaction, neuron, metabolism, predation), in sociology (relationship, acquaintances, groups), in IT (webpages, scientific citations, social medias), in infrastructure (cities agreement[Kaltenbrunner et al., 2014], transportations, banks, mobile relays) and even in unexpected areas such as linguistics and games (semantic[Corominas-Murtra et al., 2009], football games, tennis games[Radicchi, 2011], medieval history[Rodier et al., 2014]) and this list is of course far from being exhaustive.

In this thesis we will apply the Google matrix tools to complex networks defined for various situations such as the DNA sequences of several species, the neural system of the *C.elegans* worm, the ancient strategy board game called *Go* and compare them with previously studied networks such as university webpages or Wikipedia articles.

1.3 Tools to study the complex networks

The richness of the complex network approach opens up a lot of new questions and problems to be tackled in many different angles. Let us mention briefly without much details that there exist several well documented approaches such as the percolation theory which draws analogies with the behaviour of fluids filtering through a porous material, the compartmental models that are widely considered for epidemic spreading problems and so on, among them the linear algebra approach is of particular interest for us.

Matrix representation is a powerful tool to model finite size networks and characterize their topological features by spectral or eigenvector analysis. Typically a given static network of size N is represented by a square matrix of size $N \times N$ where the nodes are labeled along the columns and rows of the matrix so that the edges are given by the matrix elements. The adjacency matrix A is a well-known example of such a representation where the elements $a_{ij} = a_{ji} = 1$ when the nodes i and j are connected and $a_{ij} = a_{ji} = 0$ otherwise. This definition can be easily extended to directed graphs by removing the symmetry $a_{ij} = a_{ji}$.

Other matrix representation variants include incidence matrix, degree matrix, Laplacian matrix and we will see in the next chapter that the Google matrix is constructed thanks to a variant of the adjacency matrix.

Centrality measures

In a given static graph we may wonder which vertices constitute the most crucial part of the network, for instance which ones are the most influential nodes, or which ones participate the most in the stability of the network and so on. To address those questions several quantities, called *centrality measures*, have been proposed which allow us to determine quantitatively the relative importance of the nodes within a network [Freeman, 1979]. Finding an appropriate centrality measure was in fact the key question asked by the founders of Google in the context of World Wide Web navigation.

The centrality is typically a real value, defined for a given node, which can be computed for every node of the system in order to compare their importance. Among the various measures the four main types are the degree centrality, the closeness centrality, the betweenness centrality and the eigenvector centrality.

The first one C_D is defined as the degree of the node i, formally denoted by $C_D(i) = deg(i)$. This is the most straightforward measure and translates a node's importance simply to how many neighbours that node can directly affect (or be affected by) when the information flows through it.

The second one C_C is defined thanks to the sum of the distances that separate a node i from every other nodes, $C_C(i) = (\sum_j d(i,j))^{-1}$. The distance d is taken to be the shortest path and this measure describes how far, in terms of number of steps, a node lies from all other nodes. This is therefore used to assert how long it takes for the information to reach the network from the considered node.

The third measure C_B is defined for a give node i thanks to the shortest path lengths between all the possible pairs of vertices :

$$C_B(i) = \sum_{s \neq i \neq v} \frac{\sigma_{st}(i)}{\sigma_{st}}$$

It considers the fraction of the shortest paths σ between two nodes that goes through node i among all the shortest paths between them[2]. This fraction is computed for every possible pairs and summed up, this measure describes to what extent a node plays the role of relaying point. In practice the betweenness centrality is known to highlight nodes essential to the robustness of the network meaning that one can disrupt a system very quickly by removing the nodes in the order given by this measure.

[2]A generalization based on random walk and including contributions of other than shortest paths is given in[Newman, 2005]

The last one is defined thanks to the matrix representation of a network and assigns a score to all the nodes so that the relative importance of each one of them can be deduced. This measure gives a higher score to the nodes that have connections to other high scoring nodes and helps to identify the most influential nodes in a given network. We will see in the following sections that the scoring system developed by Google is an eigenvector centrality measure.

Community structure

One of the most challenging analysis in the static network case is the detection of community structures. By community we mean a set of nodes, often referred to as a *cluster* in traditional network science terminology, that are more connected among themselves than to the rest of the network. Such a set of nodes form a group that might be interpreted in a concrete example as a class of objects with similar specificities[Girvan and Newman, 2002]. However there are no clear definition of the concept of communities and various algorithms have been proposed to detect such clusters (statistical inference, modularity maximization, clique based methods, ...) and some methods are tested on artificially produced communities so that it is still ambiguous and difficult to apply them in real life situations.

In [Fortunato, 2010] the author gives a complete overview of community detection techniques discussing the main algorithms and explaining why the problem hasn't been solved yet[3]. It is also known that the problem is even harder for the directed network case nevertheless we will explore a possible way of extracting communities thanks to the Google matrix tool that could help us in providing a different insight on cluster organization.

1.4 Technical aspects and challenges in I.T networks

In parallel to the development of the graph theory, the second half of the XX$^{\text{th}}$ century had a favourable political and historical context to intensifying scientific effort to materialize automated computing devices in order to perform mechanically or electronically tasks helping deciphering secret codes, encrypting communications, computing ballistics and so on. The theoretical foundations of logic and informatics were set by famous people like Alan Turing who introduced in 1936 a gedankenexperiment known later as Turing machine [Turing, 1937]. A Turing machine is an abstract model to implement a mechanical device to perform calculus following clear instructions depending on its state. Any problem that can can be treated with a clearly defined procedure solvable by a Turing machine means that a physical device, provided that it is powerful enough, can be built to solve it. On the contrary, there is no way of solving a problem if it is non solvable by a Turing machine.

This notion is thus an important precursor to the programming languages by providing a formal understanding of algorithms before the existence of actual computers and giving us an idea of what a computing machine should be to tell it apart from a simple automate. Besides the well known and advanced programming languages such as C/C++ (in which the simulations of this work were done), anecdotically, the typesetting system LaTeXwith which this document is written is also equivalent to a Turing machine.

A few years afterwards several pioneers, among whom was the famous scientist John von Neumann, proposed a scheme for concrete realization of a fully functional computer which will be known as the von Neumann architecture [Burks et al., 1946]. To sum it up briefly this scheme suggested that a computer should be made of four parts : the arithmetic unit performing the basic operations, the control unit preparing the basic operations, the memory to register the program and the current ongoing operations and an input/output device to communicate with the external world.

[3]See the survey[Malliaros and Vazirgiannis, 2013] for the directed networks.

Birth of the Internet

These foundational works greatly enhanced the technological development of the second half of XX^{th} century so much that research section of the USA defense agency promoted the implementation of efficient communication between their agents. The idea of communicating devices linked to each other through a standardised protocol is a precursor of the Internet. In the early 1990s the CERN came up with an elegant standard to access written documents but also image contents, videos and sound records that were addressed with a chain of characters known as URL and reachable through a browser [Berners-Lee, 1989]. The simplicity of the protocol along with the elaboration of tools to create such multimedia documents, called webpages, helped the growth and the popularization of the world wide web so much that estimates in 2013 put the number of active websites to $5 \cdot 10^8$ and the total number of webpages is perhaps ten times larger, at least for the indexed part of the Internet, leave alone the deep web.

The main success of the WWW nowadays comes from its numerous practical advantages over physical documents and its huge reservoir of knowledge accessible through not only computers but telecommunication devices as well. As the society tends to take the networking spirit even further with various enhancement of smartphone and mobile devices, signal processing technologies, cloud computing and so on it is crucial to come up with a profound understanding of the network properties and its behaviour.

Data challenge

Besides the intrinsic dynamical nature of the Internet which is in constant growth and undergoes constant modifications there is the fundamental question *How do we efficiently retrieve information ?* Indeed one of the major task in such a huge evolving network is to navigate efficiently in the ocean of webdocuments and find as quickly as possible the most relevant piece of information one is looking for. This question lead people to create *search engines*, software that use automated web crawlers to explore the WWW and collect information about webpages and their contents to build up a reference database from which a table of relevancy is constructed. Older versions of search engines worked on the basis of keywords query, that is, the database of indexed pages are analysed via their contents and a list of relevant words are registered so that each time somebody makes a query by typing one of the words, the search engine returns the webpages where these words appear frequently or in strategic positions such as titles and beginning of paragraphs.

These methods, despite all the improvements one can make about extracting keywords and scoring webpages according to content, suffer from serious issues related to the human way of thinking and judging the importance of a document and also by the inability of our computers to process languages in a semantic level. It is a non trivial task to solve stemming problems and defining similar meaning words in a specific context, those similarities might be natural for us but hard to catch for an automated computer program which often show up as completely unrelated search results. This technique is also highly dependant on spelling specificities such as words which have almost same spellings, case sensitivity and so on. These failures motivated a novel approach to information retrieval that was by the way the basis for the success of Google [Langville and Meyer, 2006].

1.5 Google and network approach to information retrieval

The fundamental reason behind the difficulty to retrieve needed documents on the WWW is partially related to its structural organization. Indeed, unlike traditional archives such as libraries where files and books are arranged in a specific manner with categorization for providing a methodical and easy access to their content, there are no centralization nor any kind of hierarchy within the Internet. To circumvent all these limitations it was necessary to approach the webpages

scoring problem from a different angle. Instead of relying solely on the content of the webpages, one can try to look at the network perspective of interconnected web documents and ask the question *What are the most important webpages corresponding to a given query ?* In addition to suggesting a couple of websites and listing them in order of importance, this centrality oriented method should also correspond to how human web users define the importance of a website. Indeed the average user relies on the first results returned by a search engine and usually does not bother to look further in the listing.

During the years 1995 and 1996, two PhD students in Standford University, Sergey Brin and Larry Page, met and came up with a brilliant idea of assigning a recursively computed score to list the webpages in order of importance that happens to correlate well with what people expect about a website's relevancy to answer their queries [Page et al., 1999]. This method, called the *PageRank score*, is based on the point of view that a hypertext link (link that people put in their websites as suggestions for visiting related or complementary materials or for reference materials) is some kind of recommendation system. In a sense if many people put a link from their webpages towards a particular website it means that they consider that one as a relevant source of documentation and worthy to be visited. Therefore the more a website has incoming links the more it is popular or important. The recursion is taken into account by the fact that the score of a website is higher if other important websites (with a high score themselves) point to it. Similarly the weight of a recommendation (a hypertext link) of a website is decreased if it tends to point to many other websites because the value of its recommendation would be lower. Another drastic advantage of considering such incoming links is that one cannot easily fake one's own website importance by artificially boosting the score which is also the reason why outgoing links are discarded in this analysis.

Around the same time a very similar conceptual approach was proposed by Jon Kleinberg in the form of HITS algorithm [Kleinberg, 1999]. This query dependant method is sometimes considered as a precursor of PageRank scoring system as it assigns a pair of values called *hub* and *authority* scores to the nodes of a network (webpages) based on their ingoing and outgoing link structures. A high hub score indicates a node pointing to many other nodes and a high authority score indicates a node pointed by many hubs. Similarly to the PageRank score, HITS values are computed recursively but using mutually the ingoing links and the outgoing links. This dependence on outgoing links and on the query eventually made PageRank algorithm preferred over HITS.

At the end of their PhD thesis Brin and Page became the founders of Google the now multi-million dollars company dominating various aspects of the Internet world in terms of search performance and information providing services[Ginsberg et al., 2009, Preis et al., 2010]. Even though to this day several dozens of equally important factors are taken into account by their search engine in order to provide a high quality tool, the idea of PageRank scoring was and still is at the core of its success thanks to its easy computability and efficient results making it a good compromise between relevancy and computing cost and therefore rendering it applicable to the evergrowing World Wide Web. On a funny ending note the name Google originates from a misspelling of the word Gogol representing the huge number 10^{100} probably as a metaphor for the huge database that Google can handle and the PageRank scoring is a word play using "webpage score" and the last name of Google's co-founder Larry Page.

1.6 Aim of the thesis

This chapter presented a brief overview of some basic concepts about the complex networks and grasp the main developments that lead to modern day information technologies and the related challenges. The founders of Google developed a highly efficient and promising tool to study the topology of large scale-free directed networks. As it works well with the webpages network, one can expect these tools to yield interesting results and shed a new light on various real-life systems that can be viewed as a directed network. In the next chapter we are going to present the mathematical theory behind the PageRank scoring system which can be viewed as an eigenvector of a matrix

called the *Google matrix*. We will see how to construct this matrix and discuss its eigenvalues and eigenvectors properties in chapter 2. In the following chapters 3, 4 and 5 we will discuss those properties in more details by applying the Google matrix analysis on some concrete examples of real world systems. It will in the same time bring the reader in a journey from small scale to large scale systems illustrating how network theory can be broadly used to gain some insight in many different situations. Before concluding this work, in chapter 6 we will discuss the use of PageRank in a different context related to the field of socio-physics and opinion formation study.

Chapter 2

The Google matrix

2.1 A brief reminder about Markov Chains

Before diving into the mathematics of the PageRank scoring system, which is a probability distribution vector, let us briefly explain the very closely related model of Markov chains. This tool of probability theory was developed around 1906 by the great Russian mathematician Andrei Markov to describe stochastic processes undergoing transitions [Markov, 1906]. Markov models have been extensively studied and found many applications in areas such as physics, biology, statistics and finance. Among the variants of the models, we will concentrate on the simplest of them : discrete-time finite state space homogeneous Markov chain.

Formally a Markov chain is a sequence of random variables $X_0, X_1, X_2, ...$ having the so-called *Markov property*, meaning that the probability of the future event X_{t+1} depends only on the current state X_t and not on the history of the sequence. This property is also referred to as *memoryless* process explicitly given by :

$$Pr(X_{t+1} = x | X_1 = x_1, X_2 = x_2, ..., X_t = x_t) = Pr(X_{t+1} = x | X_t = x_t) \tag{2.1}$$

The indices $1, 2, ...$ is generally considered to label the time evolution and here it describes the state of the system at *discrete* time steps. The outcomes of the random variables are called *states* and the set S of all possible states is called the *state space*, which will be considered finite in our case.

If the system is evolving between a fixed number of states N, the stochastic transitions can be represented by a matrix P of size $N \times N$ whose elements $P_{ij} = Pr(X_{t+1} = j | X_t = i)$ describe the transition probability from state i towards state j. By definition the elements are non negative with $0 \leq P_{ij} \leq 1$ and the sum of the elements along each row of P is equal to one $\sum_i P_{ij} = 1$ therefore the matrix P is said to be *row-stochastic*. If the conditional probabilities do not depend on the position along the sequence, that is if the matrix elements P_{ij} are independent of the time steps t, the Markov model is said to be *homogeneous*. Such a Markov process is very often schematically represented by a directed graph (cf. Fig 2.1).

One of the most crucial notions in the Markov chain model is the limiting behaviour of the random variable sequence. The limiting distribution π is a row vector of the same size as the state space and whose entry i corresponds to the time that the system spends in state i in the long run, which is expressed as :

$$\pi(i) = \lim_{t \to \infty} \frac{1}{t} \sum_{j}^{t} \delta(X_j, i) \tag{2.2}$$

If such a limit exists, after the proper normalization $\sum_i \pi_i = 1$, it is considered as a *stationary* probability distribution meaning that the measure π is left invariant by the transition matrix $\pi = \pi P$. Alternatively we can look for this invariant measure by solving the eigenvector equation for the transition matrix P.

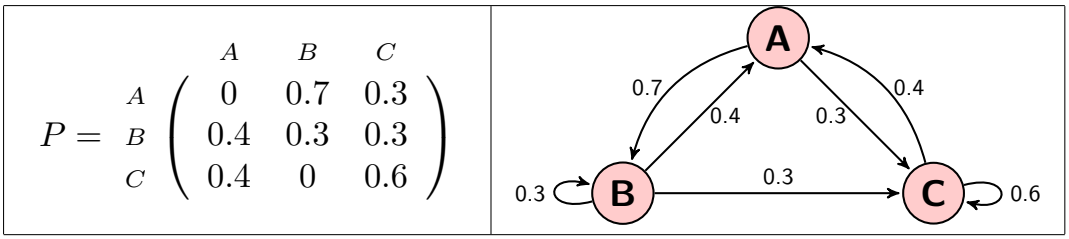

Figure 2.1: Illustrative example of a matrix representation of a 3 states homogeneous Markov chain with its directed graph representation.

The study of stationarity is closely related to the concept of first return time T_i which is the step when the Markov process returns back to the state i for the first time after having left it previously. Considering the lower bound of the ordered sequence of random variables $X_0, X_1, X_2, ...$ we have that $T_i = inf\{t \geq 1 : X_t = i | X_0 = i\}$. The set of first return times T_i are also random variables and the quantity $Pr(T_i = t)$ describes the probability that the Markov process starting from state i returns to state i after t iterations. If there is a finite probability $Pr(T_i = +\infty) > 0$ that the system will never return to state i, that is if $\sum_{t=1}^{\infty} Pr(T_i = t) < 1$ the state i is called *transient*. Otherwise it is called *recurrent* and if all states are recurrent the Markov chain is said to be recurrent.

Moreover if the expectation of the first return time of state i is finite $\mathbb{E}_i(T_i) < +\infty$, the state i is called *positive recurrent*. Similarly if all states share the same property the Markov chain is said to be positive recurrent.

One more useful definition is the *period* k of a state i which is nothing more than the greatest common divisor of the set of recurrence times $k = g.c.d.\{t : Pr(X_t = i | X_0 = i) > 0\}$. If the state i occurs at irregular times, $k = 1$ and the state is *aperiodic*.

There are several ways to determine the existence of a stationary state, the one that we are interested in involves the notion of irreducibility. A Markov chain is said *irreducible* if there exist an integer $t > 0$ such that $Pr(X_t = j | X_0 = i) = P_{ij}^{(t)} > 0$ for any pair of states i and j, in other words if there exist a probability of transition from any state i to any state j.

In terms of directed graph representation this requirement translates into the graph associated to the transition matrix P being strongly connected, meaning that for each pair of vertices (i, j) there is a path going from i towards j.

With those concepts, an important theorem states the existence of a stationary probability distribution which is by the way the root of the existence and unicity of the PageRank vector.

Theorem 1 (Existence and unicity of stationary state)**.**

Every irreducible Markov chain with a finite state space is positive recurrent, thus having a unique stationary distribution π. And if the chain is aperiodic, π is the limiting distribution $\pi = \lim_{k \to \infty} P^{(k)}\mathbf{v}$ for any probability distribution $\mathbf{v}$.

Random walk on graphs

A random walk is a mathematical description of a path formed by some stochastic process and modeled by random steps. Such a model is helpful to study complicated dynamical processes that look random but might not be so in reality such as stock market fluctuations or molecules

trajectories and they are usually represented by a Markov chain. Random walks can be performed on various objects such as a line, a plane or even on mathematical objects such as graphs [Rudnick and Gaspari, 2004].

The original idea of the PageRank inventors was to consider an Internet user as a random walker on a large network. The web surfer visits some webpages and clicks randomly on a hypertext link listed on the current website he is looking at. He does so at each step which can be a rough but still decent approximation to the average behaviour of human Internet users. Indeed usually people tend to navigate on the web following some links from the webpage they are currently looking at and in general this choice is unrelated to the websites visited previously. If we imagine a random surfer moving across the network at each step for sufficiently long time it will eventually revisit some webpages several times. We can interpret those webpages as important ones because they have many incoming links from other important websites and in the long run the time that the random surfer spends on each site would determine their relative importance. The PageRank scoring system is thus seen as one of the greatest applications of the Markov chain theory through the imagery of a random walk on a complex network.

We mentioned earlier that disabling the nodes following the betweenness centrality order rapidly destroys the network, in fact removing the nodes following the PageRank order is also quite an efficient way of disrupting the network. However contrary to the first centrality measure the PageRank measure is much more easier to compute as we will see in the following sections.

2.2 Summation formula of PageRank

Let us get back to the network science with a surprising anecdote, Brin and Page's first papers about their search engine did not even mention Markov chain models. Not knowing the strong connection between their PageRank scoring method and the Markov chain they derived a summation formula to assess a score of a webpage by analysing the structure of academic papers citations network [Langville and Meyer, 2006]. Their idea is that the PageRank score $p(i)$ of a website i should be the sum of all PageRank scores of websites pointing to i.

$$p(i) = \sum_{j \in B_i} \frac{p(j)}{|j|} \tag{2.3}$$

where B_i is the set of websites pointing to i and $|j|$ denote the number of total outgoing links from webpage j. This summation formula requires the unknown score of neighbouring webpages, to overcome this problem they rendered the formula iterative :

$$p_{t+1}(i) = \sum_{j \in B_i} \frac{p_t(j)}{|j|} \tag{2.4}$$

so that the scores which are computed for each webpage at step t use the scores computed previously at step $t-1$, starting from an initial distribution of values $\mathbf{p_0}$ which can be set for instance to the uniform vector $p_0(i) = 1/N \ \forall i$ with N being the number of websites indexed by the search engine.

From this point on we can naturally wonder if the iterative process does always converge or not, if the convergence is fast or slow and if the final PageRank vector is unique or depends on the initial values. In fact there are several situations leading to convergence problems (cf. Fig 2.2) in this iterative process, for example when the random surfer falls in a particular site with no outgoing links, called a *dangling node*, it stops there meaning that at each computing iteration that particular node absorbs more and more probability falsely increasing its PageRank score. Because of that behaviour, those nodes are sometimes referred to as *rank sink* and they are quite common on the Internet especially when we consider that many webpages have links to downloadable documents and multimedia contents that lead to nowhere.

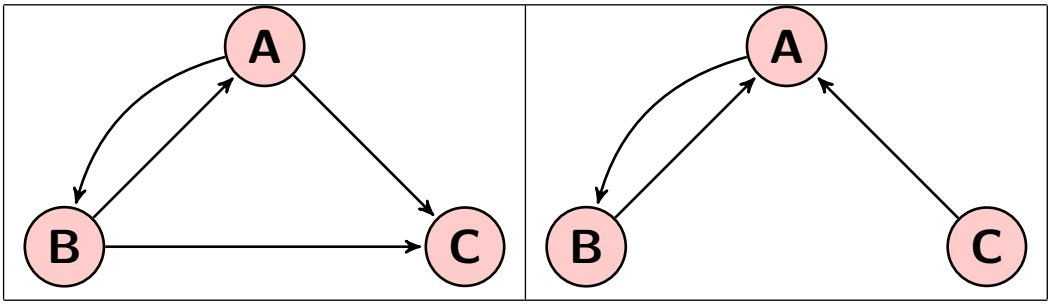

Figure 2.2: Simple examples of rank sink : node C is a dangling node (left), nodes A and B form a dangling group (right).

In addition to that there is also the case of a cluster of webpages with internal links only among themselves, in which case the random surfer gets trapped in an area formed by a subset of sites from which it cannot escape. To understand these issues mathematically we will switch to a matrix representation of the summation formula and give the recipe to carefully handle the modifications at the end of which the resulting matrix is the Google matrix, thereby ensuring the existence and unicity of the PageRank vector to which any initial distribution will converge.

2.3 How to construct the Google matrix ?

Throughout this section we will discuss the construction of the Google matrix G using a simple toy model example of a small directed graph pictured in Fig. 2.3, the recipe for larger networks is exactly the same.

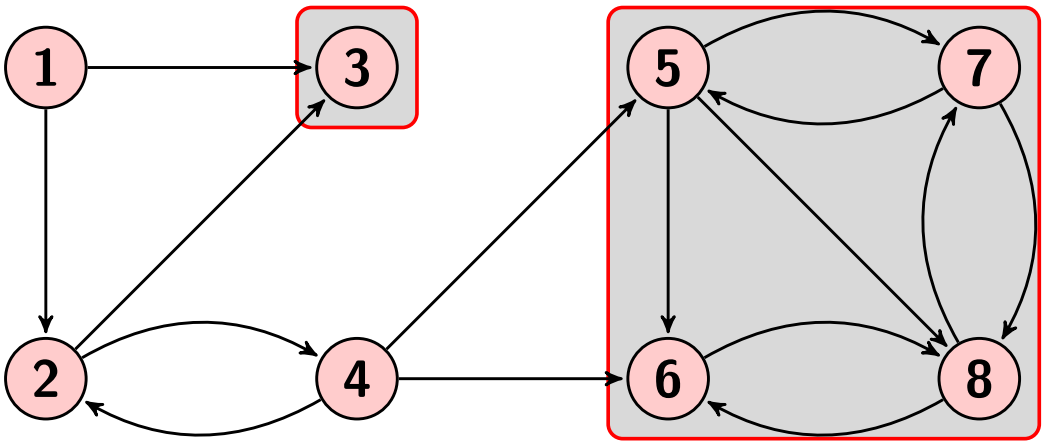

Figure 2.3: Directed graph used to illustrate the construction of the Google matrix G in this section. Probability absorbing areas (rank sink) are shaded in gray.

1 : Asymmetric adjacency matrix

The first step consists of building the matrix A which describes the connectivity structure of the directed network. It will be an asymmetric matrix of size $N \times N$ where N is the number of nodes in our system. In the literature there are differing conventions to represent outgoing links

in columns or in rows, here we use the column labels to designate the origin of a link and the row to designate the destination so that the matrix elements $A_{ij} = m$ if there are m links from node j pointing towards node i where m is an integer number representing the multiplicity of the link and $A_{ij} = 0$ otherwise. In our toy model we don't have multiple links, so $m = 1$ and the matrix corresponding to the network in Fig. 2.3 reads :

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} \tag{2.5}$$

Next we need to normalize the columns to one so that the matrix becomes similar to the transition matrix of a Markov chain. The elements would be the probability of getting from a node to another one and the matrix vector product using A' is the transcription of the summation formula in eq. 2.4 :

$$A' = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 1/3 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/3 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1/3 & 1/3 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1/3 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1/3 & 1 & 1/2 & 0 \end{pmatrix} \tag{2.6}$$

The physical motivation behind column normalization is that we want to treat all the outgoing flows on the same footing so that comparisons between nodes capture mostly their efficiency of connections rather than their volume of connections.

2 : Handling the dangling nodes

In the next step we have to deal with the dangling nodes that attract all the probability upon themselves. Those nodes are columns full of zero in the matrix A' because they are precisely the vertices without any outgoing links. Mathematically we need to render the A' matrix *stochastic* therefore the columns of zeros are replaced with columns of $1/N$ where $N = 8$, in our case, is the size of the system. Formally we then have the stochastic matrix $S = A' + (1/N)\mathbf{e}\mathbf{d}^T$ where $\mathbf{e}$ is the column vector of ones and $\mathbf{d}$ the column vector whose entry $d(i) = 1$ if node i is a dangling node and $d(i) = 0$ otherwise.

Physically the interpretation of this trick is that virtual links are put from the dangling node towards every other nodes of the system so that when the random surfer falls into the rank sink it will then go randomly and with equal probability to any other part of the network. Regarding our behaviour as Internet users it still makes sense as once we hit a link to download a pdf file for example we will then visit a totally different website.

For our example of Fig. 2.3 where the vertice number 3 is a dangling node, the stochastic matrix S now reads :

$$S = \begin{pmatrix} 0 & 0 & 1/8 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 1/8 & 1/3 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 1/8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 1/8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/8 & 1/3 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 1/8 & 1/3 & 1/3 & 0 & 0 & 1/2 \\ 0 & 0 & 1/8 & 0 & 1/3 & 0 & 0 & 1/2 \\ 0 & 0 & 1/8 & 0 & 1/3 & 1 & 1/2 & 0 \end{pmatrix} \tag{2.7}$$

Sometimes this matrix is used as the Google matrix however in this form we are still not guaranteed to converge towards the PageRank in the most general case.

3 : Handling the dangling group

In the final step, we have to deal with areas of the network where the random surfer cannot escape from. Those groups of nodes can have connections among themselves but none of them have connections getting out of the group as shown in our example by the large grayed area in Fig. 2.3. Mathematically the problem arises because S is not guaranteed to be *primitive*. A non-negative square matrix M is said to be primitive if there exists an integer $k > 0$ such that $M_{ij}^k > 0$ for all pairs (i, j). To fix this property we need to add a dense rank one matrix traditionally denoted by E and usually taken to be $E = (1/N)\mathbf{e}\mathbf{e}^T$. The final form of the Google matrix G is a linear sum of this matrix E and the stochastic matrix S :

$$G = \alpha S + (1 - \alpha)\frac{1}{N}\mathbf{e}\mathbf{e}^T \tag{2.8}$$

where α is an arbitrary parameter, called the *damping factor*, taken in the range $0 \leq \alpha \leq 1$ so that with probability α the transition between node j towards i is described by the structure of the network and with probability $1 - \alpha$ the transition from node j towards any other node is rendered possible with equal probability. In our example, the Google matrix associated to the network in Fig. 2.3 reads with $\alpha = 0.8$:

$$G = \begin{pmatrix} 1/40 & 1/40 & 1/8 & 1/40 & 1/40 & 1/40 & 1/40 & 1/40 \\ 17/40 & 1/40 & 1/8 & 7/24 & 1/40 & 1/40 & 1/40 & 1/40 \\ 17/40 & 17/40 & 1/8 & 1/40 & 1/40 & 1/40 & 1/40 & 1/40 \\ 1/40 & 17/40 & 1/8 & 1/40 & 1/40 & 1/40 & 1/40 & 1/40 \\ 1/40 & 1/40 & 1/8 & 7/24 & 1/40 & 1/40 & 17/40 & 1/40 \\ 1/40 & 1/40 & 1/8 & 7/24 & 7/24 & 1/40 & 1/40 & 17/40 \\ 1/40 & 1/40 & 1/8 & 1/40 & 7/24 & 1/40 & 1/40 & 17/40 \\ 1/40 & 1/40 & 1/8 & 1/40 & 7/24 & 33/40 & 17/40 & 1/40 \end{pmatrix} \tag{2.9}$$

The added matrix E is called the *teleportation* matrix thanks to its effect on the random surfer. Whenever the surfer gets trapped inside a dangling group, now it still has a non zero probability to jump elsewhere in the network thereby continuing its network exploration. It is a common image to mentally represent the flow of probability in the network and to obtain a stationary distribution it is essential that the whole network is continuously explored.

There is also a natural argument in favor of the teleportation matrix when we consider the behaviour of actual people surfing the Internet, usually when they get bored of following a particular thread they will start to look for a different topic, modify their queries and start surfing again in some other part of the WWW network.

Perron-Frobenius operators and dominant eigenvector

In the form of eq. 2.8 the Google matrix G is stochastic, irreducible and aperiodic which in the context of Markov chain theory (cf. theorem 1) ensures the existence and the unicity of a stationary probability distribution : the PageRank vector. Indeed thanks to its definition, G belongs to the class of so called Perron-Frobenius operators therefore the famous Perron-Frobenius theorem (cf. theorem 2) applies to it, consequently ensuring that a unique strictly positive eigenvector exists [Perron, 1907, Frobenius, 1912]. This theorem has several statements, here are some of them that are directly interesting for us :

Theorem 2 (Perron-Frobenius Theorem)**.**

Let A be a primitive matrix

- The spectral radius $r = \rho(A)$ is a simple eigenvalue of A.
- ρ is the only eigenvalue on the spectral circle of A.
- There is a unique eigenvector $\mathbf{v}$ such that $A\mathbf{v} = \rho\mathbf{v}$ and $v_i > 0 \quad \forall i$.

We can now confirm that the spectral radius ρ of the Google matrix G is equal to one which is also the dominant eigenvalue $\lambda_1 = \rho = 1$. Since G is asymmetric the eigenvalues are complex valued and distributed on the complex plane inside the unit circle. The eigenvector v corresponding to the eigenvalue $\lambda = 1$ satisfies $Gv = v$ and has all positive entries.

This vector can be normalized as $p = v/\sum_i v_i$ so that the sum of its entries add up to one, the resulting vector p is called the PageRank vector and has a meaning of a probability distribution over the nodes of the considered network. To derive a ranking from this probability distribution, since by definition the PageRank vector is positive definite, we can rearrange its elements $p(i)$, through a permutation $\sigma(K) = i$, in decreasing order to obtain a list of values $p(K)$ so that $p(K_1) > p(K_2) > ...$ whose indices K_i denote the rank of the nodes such that low values of $K = 1, 2, ..$ mean a high ranking thus indicating very important nodes. The normalized vector corresponding to our little example in eq. 2.9 is $p^T = (0.0318, 0.0594, 0.0683, 0.0556, 0.1187, 0.1948, 0.1800, 0.2914)$ which can be reordered decreasingly with the following permutation $\sigma = (8, 7, 6, 5, 3, 2, 4, 1)$. The most important node, the most highly ranked is $K_1 = 8$ followed by $K_2 = 7$ and so on.

The word PageRank is sometimes used ambiguously to designate the actual value or the ranking, to avoid any confusion we will refer to the probability distribution vector as *PageRank vector* and the ranking of its elements as *PageRank indices*.

PageRank vector numerical computation

The PageRank computation can be stated as the following eigenvector problem for $\mathbf{p}$ with the normalization constraint $\sum_i p_i = 1$:

$$G\mathbf{p} = \mathbf{p} \tag{2.10}$$

When dealing with relatively small size systems ($N \lesssim 10^4$) where the whole matrix G can be stored and handled easily, it is straightforward to diagonalize the matrix and obtain the eigenvector corresponding to $\lambda_1 = 1$. However for the huge networks such as the WWW handling and performing operations on the whole matrix is not feasible and the memory requirement would be far off the technical limitations.

An interesting alternative would be the *power method* which is an iterative method and one of the several numerical recipe used to find the stationary solution of a Markov chain[Stewart, 1995, Mises and Pollaczek-Geiringer, 1929]. Among all the methods to find the dominant eigenvector of a matrix, the power method is the simplest one and the easiest to implement making it the favorite candidate in PageRank vector computation despite its well-know algorithmic slowness. So what advantages does the power method give to be still interesting today ? The answer is threefold : First, in the context of WWW network it is suited to the *sparse* structure of the normalized adjacency matrix, second it has a *linear complexity* and converges quickly in some cases and third it is a matrix-free method.

Matrix sparsity

A matrix which has a large proportion of its entries equal to zero is said to be *sparse*, otherwise it is said to be *dense* and the fraction of non zero elements in the matrix is called the sparsity. For the general case there are no clear definition of the sparsity in the literature as this notion is used qualitatively in most cases[1].

This notion is important in numerical computing as there exists several methods accelerating and improving the efficiency of computations when applied to sparse matrices. Moreover there are specific sparse matrix representation formats which stores the minimal needed information content of the matrix by disregarding all the zero entries consequently saving a lot of memory space and thus allowing for larger matrix computations. In the case of the World Wide Web network, the matrix describing the connectivity structure would be huge in size, about $10^8 \times 10^8$, but on average a typical webpage has about ≈ 10 connections towards other websites making the number of non zero elements to be about $\approx 10^9$ so that their fraction is about $\approx 10^{-7}$ which is extremely sparse.

Algorithmic complexity

Computational complexity theory studies the intrinsic properties of mathematical problems and classifies them according to their difficulty [Arora and Barak, 2009]. The notion of *big O* is used to denote the asymptotic speed behaviour of an algorithm in an arbitrary unit of time in the worst case scenario. The complexity is usually expressed with the size of the input data n and gives the time needed for the algorithm to terminate in the case that all the computations must be done (disregarding simplifications and shortcuts due to a specific problem). For example if an algorithm has a complexity of $O(n^2)$ it means that if we give a two times larger dataset as an input to be processed, the algorithm will take four times longer to run and terminate on the same machine. There are many different behaviour such as the constant complexity, denoted by $O(1)$, meaning that the size of the input data is irrelevant to the computation duration or the linear complexity, denoted by $O(n)$, meaning that the runtime is proportional to the data size. Naive coding and simple solutions often results in high complexity, if we wish to optimize an algorithm it is crucial that we try to reach the lowest possible complexity.

Due to the peculiarities of the Google matrix, especially in the case of a sparse network, the eigenvector problem stated in eq. 2.10 can be rewritten thanks to the sparse connectivity matrix A' as :

$$\mathbf{p}_{t+1} = G\mathbf{p}_t = \alpha A'\mathbf{p}_t + (\alpha \mathbf{d}^T \mathbf{p}_t + 1 - \alpha)\mathbf{e}/N \tag{2.11}$$

where $\mathbf{d}$ is as before the vector indicating the dangling nodes. In order to compute the eigenvector at step $t+1$ we need to know it at step t and since the stationary solution is unique we

[1]However in the context of graph theory a rigorous measure is suggested in [Randić and DeAlba, 1997] in the form of compactness $\rho = 2en/((n-1)(n^2-2e))$ for simple undirected graphs with n vertices and e edges. As there cannot be a graph with $\rho = 1$, if $\rho < 1$ the graph and the corresponding adjacency matrix are sparse and if $\rho > 1$ they are dense. A similar argument can be derived for simple loopless directed graphs.

end up on the same vector whatever the initial probability distribution chosen at $t = 0$, usually the uniform vector $\mathbf{p}_0 = (1/N)\mathbf{e}$ is used.

Despite the fact that the indexed webpages are growing in number there are in principle no reason for the average number of connections per page to change much. It is safe to assume that one iteration with A' matrix is similar to a matrix vector product where the number of non zero elements in each row of the matrix is bounded by a constant $C \ll N$. In such a case the complexity of one iteration step is $O(n)$ linear and therefore the computation is quite fast. Moreover the whole procedure does not require the handling of the whole Google matrix, indeed only the sparse representation of A' and the current iteration of the vector $\mathbf{p}_t$ are stored in memory.

Finally Brin and Page originally stated that only about 50 to 100 iterations are enough for the level of precision needed for website indexation.

In practice the Internet is constantly evolving and growing so that a real time PageRank computation is practically impossible, instead Google is using huge servers to crawl the web and perform the computations once every two or three months to update the score of all the webpages.

2.4 Spectrum and PageRank properties

Eigenvalue spectrum properties

As mentioned earlier the directed network description makes the Google matrix asymmetric, therefore the matrix diagonalization produces complex conjugated pairs of eigenvalues that are distributed inside the unit circle in the complex plane (cf.theorem 2). As we will see in chapter 3 the eigenvalue cloud in itself provides some insight regarding the structural properties of the considered network and in chapter 5 we will discuss the next to leading eigenvalue and their ties to the community structures.

In Fig. 2.4 two concrete examples of spectrum of the Google matrix are shown for university websites in 2006[2] where some of the largest eigenvalues of the webpages of Cambridge university (left panel) and Oxford university (right panel) are displayed [Frahm et al., 2011]. This might seem to be a little restrictive to represent the entire World Wide Web nevertheless both of these networks are already very large with respective sizes of $N = 212710$ and $N = 200823$ nodes for Cambridge and Oxford, they also show scale-free behaviour and small world property. Moreover they show a sparse structure with a respective number of directed links of $N_l = 2015265$ and $N_l = 1831542$ which is about ≈ 10 connections per node for both of them.

Effect of α damping factor : These spectrum (in Fig. 2.4) were computed at $\alpha = 1$ which is technically the spectrum of the stochastic matrix S. However both the spectrum of S and of G are very closely related by the damping factor α as stated in theorem 3. This parameter determines to what proportion the Google matrix describes the actual network structure and the random hopping term and effectively scales all but the leading eigenvalue of S irrespective of the specificities of the teleportation matrix. The scaling introduces a spacing between dominant and next to dominant eigenvalues $|\lambda_1| - |\lambda_2|$ which is called a *gap* in physicists terminology.

Theorem 3 (Eigenvalue relations)**.**

If the spectrum of the stochastic matrix S is $\{1, \lambda_1, \lambda_2, ..., \lambda_N\}$, then the spectrum of the Google matrix $G = \alpha S + (1 - \alpha)\mathbf{e}\mathbf{v}^T$ is $\{1, \alpha\lambda_1, \alpha\lambda_2, ..., \alpha\lambda_N\}$, where $\mathbf{v}^T$ is a probability vector.

In the case where the gap is arbitrarily small or nonexistent (as in Fig. 2.4), that is when the spectrum shows eigenvalues very close to the unit circle in the complex plane, the system possesses

[2]Crawled data downloaded from [SCRG, 2006].

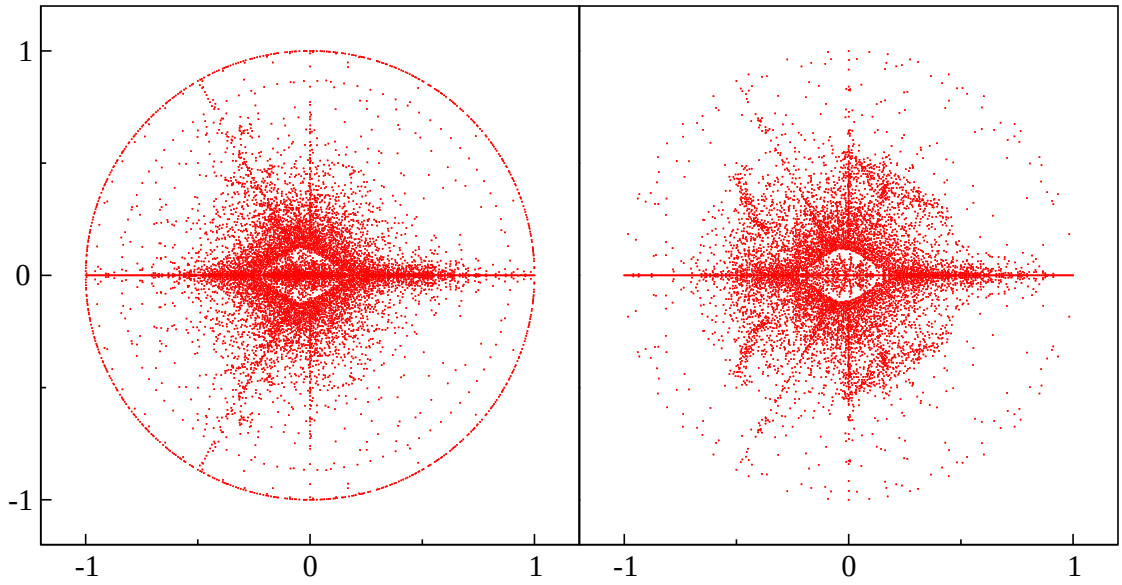

Figure 2.4: Spectrum of the Google matrix at $\alpha = 1$ for Cambridge university website (left panel) and Oxford university website (right panel). Not all the eigenvalues are displayed here and the plots are made with data from [Frahm et al., 2011].

distinct independent regions which are weakly linked to the rest of the network. If the eigenvalues are strictly located on the unit circle, the parts are disjoint which is mathematically expressed as a reducible system in which case the introduction of the teleportation matrix will connect those disjoint parts together by introducing a spectral gap.

Invariant subspace decomposition : To gain a deeper insight about those eigenvalues located around the unit circle, it is useful to consider the invariant subspace decomposition. In the typical WWW like networks the nodes can be separated into two subsets : *core space* nodes and *subspace* nodes. The first group constitutes the larger part of the network containing all the nodes from which every other nodes are reachable in a finite number of steps. If we explore the network in the same way starting from a node belonging to the second group we will eventually get stuck inside a small subset of nodes that form what is called an *invariant subspace* that is in fact left invariant with respect to the application of S. This separation allows to rewrite the S matrix in a block triangular form $S = \begin{pmatrix} S_{SS} & S_{SC} \\ 0 & S_{CC} \end{pmatrix}$ because it is possible to enter in a subspace from the core space but not possible to escape from a subspace. The subspace part S_{SS} itself can be composed of several invariant subspaces and therefore made of diagonal blocks of various dimensions which are each a Perron-Frobenius matrix thereby producing at least one eigenvalue $\lambda = 1$ for each block.

The size of the subspace part depends on the system but the distribution of the various subspaces size seems to follow an universal function. These properties along with the detailed algorithm to construct the core space and subspace are discussed in [Frahm et al., 2011].

PageRank properties

In Fig. 2.5 we show the PageRank probability vectors, eigenvectors of the Google matrix at $\lambda = 1$, computed for both Cambridge (left panel) and Oxford (right panel) university webpages. The plot shows that in logarithmic scales the probability distributions, when ordered decreasingly, have a consistent linear behaviour over a wide range of values meaning that the PageRank values $P(K)$ have a power law tendency on their ranking index K :

$$P(K) \sim \frac{1}{K^{\beta}} \tag{2.12}$$

$$\beta = \frac{1}{\mu - 1} \tag{2.13}$$

where β is the decay constant of the probability distribution which is tightly related to μ the decay constant of the in-degree distribution of the same network $p^{in}(k_{in}) \sim 1/k_{in}^{\mu}$. Indeed since the PageRank is determined based on the importance of incoming links both distribution behaviours are closely related. In the case of the webpages the in-degree distribution was found to be around $\mu \approx 2.1$ and shown to be consistent with $\beta \approx 0.9$. In later analysis those values will be used as reference values for the world wide web in order to compare the structural features of other kind of complex networks [Langville and Meyer, 2006, Zhirov et al., 2010].

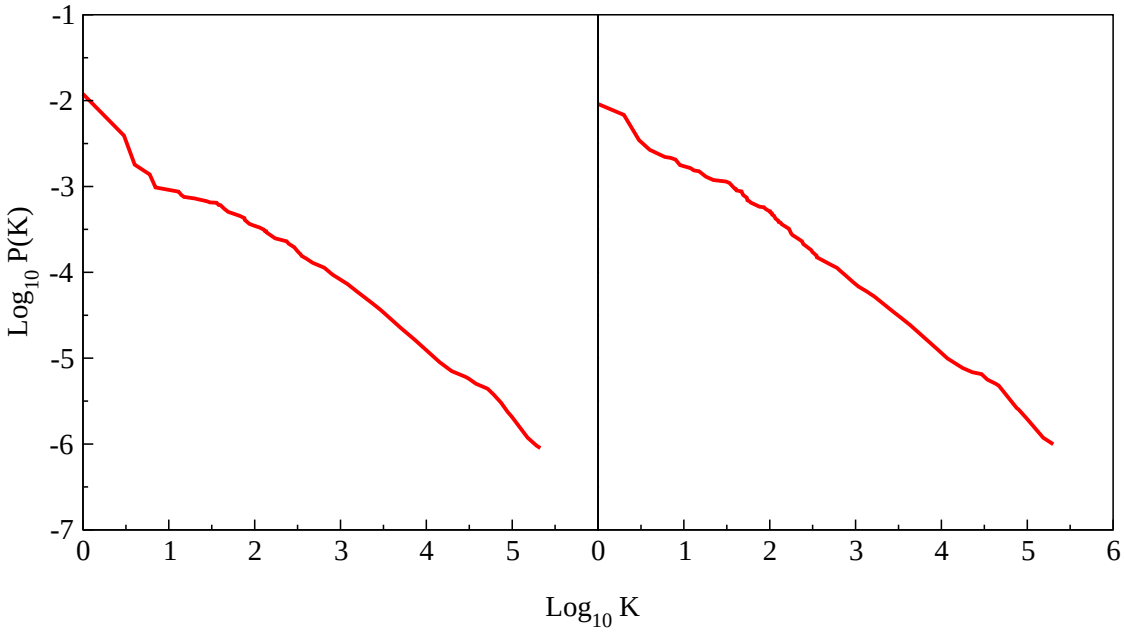

Figure 2.5: PageRank probability decay for Cambridge (left) and Oxford (right) computed with the power method at $\alpha = 0.85$.

Effect of α damping factor : The damping factor, which was necessary to make the Google matrix G primitive, effectively introduces a gap between the eigenvalues such that the next to leading eigenvalues are bounded by α, in other words $|\lambda_2| \leq \alpha$. The second largest eigenvalue is therefore at most $|\lambda_2| = \alpha$ and since in general the asymptotic rate of convergence towards the stationary solution in Markov chains is related to the ratio between the first and second largest eigenvalues in magnitude, $|\lambda_2/\lambda_1|^t \to 0$, in the case of the Google matrix the rate is the one with which α^t goes to zero in t iterations.

A low value of α results in a faster convergence but also a greater artificiality in the topology of the network since more importance is given to the teleportation matrix. Therefore the choice of this damping parameter must be a wise compromise between its usefulness in helping the convergence and keeping the network structure close to the reality. In some systems the structure of the network makes $|\lambda_2|$ naturally bounded and the introduction of α is not necessary but regarding the world wide web networks there are no such gaps so that the use of the damping factor is mandatory.

In practice the degree of the PageRank values fluctuation depends on the magnitude of the second eigenvalue, if the gap is large enough (as it is the case in the systems studied in this thesis) the PageRank vector is not very sensitive to the damping factor. However if the gap is very small the sensitivity of the PageRank values increases when α gets closer to 1 [Langville and Meyer, 2006].

Fortunately the fluctuations are not homogeneous, instead when looking at the ordered distribution decay we notice that for the largest range the distribution scarcely changes with α and those are in fact the nodes belonging to the invariant subspace. It is only the tail part, where the core space nodes are located, that varies significantly but depending on the applications this does not constitute a serious issue [Frahm et al., 2011].

The PageRank in function of α can be written as a rational function whose Taylor expansion provides a computational trick to easily determine the result of varying α without recomputing the whole vector again. Nevertheless usually the standard value of $\alpha = 0.85$ originally proposed by Brin and Page is still used today[3].

This standard value corresponds to a right compromise between accelerating the numerical computation of PageRank and keeping the network structure intact, it will be used to determine the PageRank vectors of universities webpages in the last chapter and it will also be used for small networks such as *C.elegans* neuron network. If the studied systems already contain a natural gap (such as DNA sequence networks in the second chapter) or if the matrix representation of the system is too large to be handeled in the full format when computing the spectrum and other eigenvectors, we use the value $\alpha = 1$ instead.

The google matrix, and subsequently the PageRank vector, are defined for a given directed network. For each directed network it is possible to construct a complementary directed network called the *inverted* network where the directionality of all the links are reversed. The same standard procedure can be used to compute the new google matrix denoted G^* and its principal eigenvector $\mathbf{p}^*$ called the *CheiRank* vector to avoid confusion with the principal eigenvector of the original network. In chapter 4 and 5 we will discuss in more detail the inverted network and the use of both PageRank and CheiRank to derive a more informative 2D ranking.

We have seen the mathematical ground behind the whole concept of the Google matrix which some researchers consider as one of the greatest applications of Markov chains and discussed how to construct it through a concrete example. We have explained key properties about the eigenvalue spectrum of that matrix and the use of its principal eigenvector, the PageRank vector, as a tool to assert the importance of each node within a directed network. We have also shown a concrete example of spectrum plot and PageRank probability decay typically observed for webpages networks. In the next chapter we will apply the Google matrix analysis to real DNA sequences from several species and compare the structural differences in their spectrum and PageRank distribution and discuss the statistics of the Google matrix elements.

[3] A more mathematical justification is provided in [Boldi, 2005].

Chapter 3

The analysis of DNA sequences

3.1 DNA : Building blocks of Life

Let us start our network analysis at the smallest scale : at the molecular level. All the organic life on Earth is made from the same basic compound called the *Desoxyribonucleic acid* or *DNA* for short. The DNA is a long double stranded polymer carrying the whole genetic information of the living organism. Both strands are made of consecutive sugar phosphate units on which a molecule called *base* is attached and the base together with the phosphate group is called a *nucleotide*, the DNA is therefore a long chain of repeating nucleotides.

There are four types of bases, namely the *Adenine*, the *Thymine*, the *Cytosine* and the *Guanine* respectively abbreviated by the letters A, T, C and G, which can form hydrogen bonds if they are correctly paired together, that is if A and T are paired together or C and G are paired together in which case they are referred to as *base pairs* (*bp*). Chemically the bases A and G belong to *purine* and T and C belong to *pyrimidine* categories of molecules. The two strands can therefore hold tightly together when the nucleotide sequence along one strand correspond to the complementary sequence on the second strand. This system is very handy to store and replicate a large quantity of information with a minimal mechanism, indeed whenever a copy of the information is needed the two strands are separated and each of them act like a template on which the complementary strand can grow giving rise to two identical DNA chains.

In its standard form, and free of any constraints, the DNA takes the conformation of a double helix coiled around the same axis as portrayed in Fig. 3.1. The pitch of the helix is $3.4nm$ and is comprised of 10 base pairs, so that the spacing between base pairs is $0.34nm$. Even though the size of a nucleotide is on the nanometer scale the information needed to code a multicellular complex organism is huge and consequently the sequence can be very long such that the polymer can reach lengths of several centimeters when stretched. The peculiar structural features of the DNA and its stability allow the long polymer to be coiled and compressed at several levels forming the well known microscopic scale elements, called *chromosomes*, that are the natural form in which we can find the DNA inside a cell. In fact all complex living creatures are made of cells possessing a nucleus inside which several chromosomes are present. The information commanding everything from the growth of the organism to the regulating processes necessary to its functioning are coded in several regions of the set of chromosomes which is present in each one of the creature's cell.

Historically the experimental work that led to the discovery of the DNA was made by the Swiss biologist Friedrich Miescher who identified the nucleic acids in the late 1800s. Thanks to the work of Rosalind Franklin and Maurice Wilkins on X-ray diffraction images of DNA polymer, the biologist and biophysicist James Watson and Francis Crick came up in 1953 with the full understanding of the structural topology of the DNA thereby winning the Nobel Prize shortly afterwards [Watson and Crick, 1953, Franklin and Gosling, 1953, Wilkins et al., 1953].

From that point on researchers have put massive efforts in unveiling the very complicated processes participating in the machinery of life and to this day there are still several aspects that are not fully understood despite the huge technological progress. For instance the invention of the

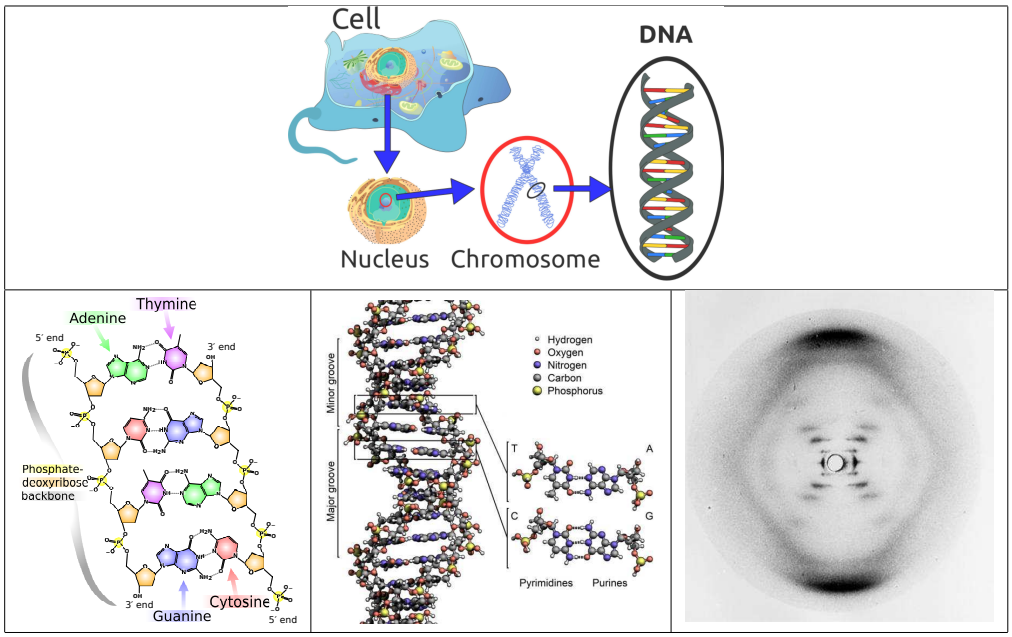

Figure 3.1: *Top :* Schematic illustration of the location of double stranded DNA strands in chromosome form inside a cell's nucleus. *Bottom left:* Illustration showing base pairing with hydrogen bonds and their chemical structure. *Bottom middle :* Illustration showing the consecutive stacking of nucleotides forming a double helix structure. *Bottom right :* Famous X ray diffraction picture number 51 taken at King's College in London. (*Pictures from Wikipedia Commons* and X-ray photography from [Franklin and Gosling, 1953]).

polymerase chain reaction (PCR) procedure helped the experiments a lot. This procedure, which also granted the Nobel Prize to its developer, is used to replicate a small sample of DNA by several orders of magnitude in a reasonable amount of time [Neuzil et al., 2006]. In practice for example a single sample could be replicated a billion times in an hour. This kind of device improved the development of DNA sequencing techniques, that is the determination of the exact arrangement of bases along the DNA backbone, so much that it becomes possible to sequence the entire genome of a given specie at an affordable cost and nowadays huge databases with more and more DNA sequences of a wide variety of species are available [EMBL, 2013]. The databases are also frequently updated with new releases with improved accuracy until every single base is known. With such a large database it becomes possible to perform rigorous statistical analysis and assert significant results regarding the various statistical features of DNA sequences and a few research groups are focusing on the statistical approach with the aim of specific sequences identification, pattern detection and so on [Robin et al., 2005],[Halpern et al., 2007],[Dai et al., 2008],[Reinert et al., 2009]. In [Mantegna et al., 1995] a method of analysis from the linguistics is used to study the frequency distribution of short DNA sequences up to 7 bases length, we propose an extension of this analysis using the Google matrix method.

Motivation : In [Frahm and Shepelyansky, 2012] the analysis of Poincaré recurrences in DNA sequences showed their similarities with the statistical properties of recurrences for dynamical trajectories in the Chirikov standard map and other symplectic maps, here we suggest that the directed network point of view is a new way of looking at the DNA sequences that can shed a new light on their statistical properties. The length of the DNA strands and the quantity of information available make the network rather large and therefore we expect the Google matrix to yield interesting information and allow a comparison between the structural organisation of the DNA sequences and the more commonly studied webpages networks.

3.2 The Network of Sequences

In general when we use the network approach it is essential to define properly the meaning of the nodes and edges. In the case of the DNA we will consider the nodes to be short fixed length sequences of bases, called *words* of length m. Since the alphabet of these words have 4 possible letters we can define the state space as the set of all possible words of length m, this will give rise to a system of finite size $N = 4^m$ considering of course repeating letters.

The biological nature of the DNA makes it easier to define directed edges between those nodes thanks to the fact that there are two possible chemical configurations at the extremities of a DNA strand, called $3'$ and $5'$. Indeed the double stranded polymer stores the information in one strand and uses the complementary strand as a template to produce a copy. This is possible thanks to a specific enzyme called *polymerase* which binds to a *primer*, an indicator sequence acting as a starting point from where the template strand needs to be paired, and slides along the strand in order to build the DNA. This enzyme has a universal behaviour in all DNA of all living organisms known until now : the reading goes from $3'$ to $5'$ so that the nucleosynthesis is performed from $5'$ to $3'$ and therefore a natural direction is assigned to any DNA sequence.

To build our network we will therefore read the DNA strands given in the database in the natural synthesis direction from $5'$ towards $3'$ and we will cut the sequences into words of length m and assign a link from word j to word i if the word i follows immediately the word j in the DNA sequence.

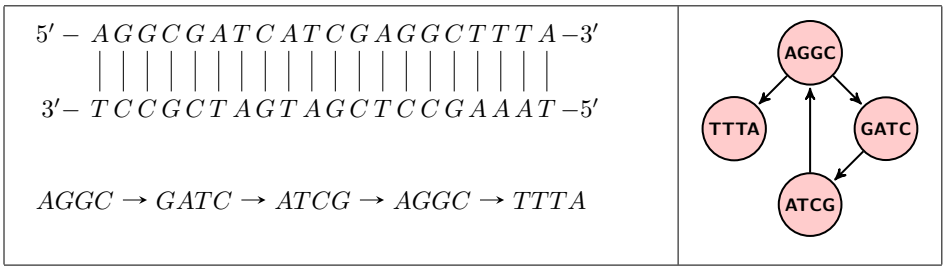

Figure 3.2: Example showing how to construct a directed network by cutting the strand into pieces of equal length words and by taking into account the natural direction of the DNA strand.

We then construct the transition matrix of size $4^m \times 4^m$ whose elements are $T_{ij} = w$ when there are w links from word j to word i and $T_{ij} = 0$ otherwise. From this connectivity matrix we can derive the stochastic matrix S and build the Google matrix $G = \alpha S + (1 - \alpha)1/N\mathbf{e}\mathbf{e}^T$.

It would be natural to ask whether words of length $m = 3$ should be the only choice making sense since a group of three bases has a direct biological meaning (c.f section 3.4). Even if it is technically correct and possible to consider a system of size $N = 4^3 = 64$, there are no particular interest in favoring the codon point of view, instead we are interested in the statistical properties of a symbolic chain. Moreover some sequences of DNA have a particular function by themselves such as promoters $TTGACA$ or $TATAAT$ who are located just ahead of a gene for instance indicating a coding portion of the DNA.

In this work we will use mainly $m = 6$ ($N = 4096$) but also study the effect of word lengths on our results by considering $m = 5$ ($N = 1024$) and $m = 7$ ($N = 16384$). Regarding the damping factor, the DNA sequence network has already a natural spectral gap so that the PageRank vector won't be much affected by values of $\alpha > 0.5$ therefore we perform all our analysis at $\alpha = 1$.

The table 3.1 shows the different sequences used and their approximate length in number of base pairs, some of them are not exactly known (at least for the version used in this work) so that in addition of A, C, G and T there is the unknown letter denoted by N_l. Words containing N_l were discarded from the analysis.

Species	Sequence length (bp)
Bos Taurus (Bull)	$2.9 \cdot 10^9$
Canis Familiaris (Dog)	$2.5 \cdot 10^9$
Loxodonta Africana (Elephant)	$3.1 \cdot 10^9$
Danio Rerio (Zebrafish)	$1.4 \cdot 10^9$
Homo Sapiens (Human)	$1.5 \cdot 10^{10}$

Table 3.1: Species used in this work and the length of their DNA sequences, the datasets were obtained from [EMBL, 2013].

The DNA sequences used in this work have a length L of about a billion base pairs except for the Human sequence which is a concatenation of sequences taken from 5 different individuals, the number of transitions are thus about $N_t \approx L/m$ because the fraction of transitions involving neglected words is negligible. We can see that since m is relatively small the number of transitions is large which is another argument in favor of word length larger than $m = 3$. Indeed the capacity of the Google matrix method to highlight structural specificities of a network is a balanced interplay between the number of available nodes in the state space and the number of links covering those nodes.

3.3 Matrix, Spectrum and The Principal Eigenvector

Before considering the eigenvalue spectrum we can observe that the G matrix at $\alpha = 1$ for the DNA sequences is dense since almost all the matrix is full which is drastically different from the case of webpages networks. The Fig. 3.3 shows for qualitative comparison a part of the G matrix of three different networks in PageRank basis where a strong connectivity between top nodes is visible in the DNA case. In a sense Google matrix method is commonly used to study networks in the sparse matrix limit cases but here we treat the other limiting case where the stochastic transition matrix is dense. It is therefore interesting to take a look at the statistical distribution of the matrix elements.

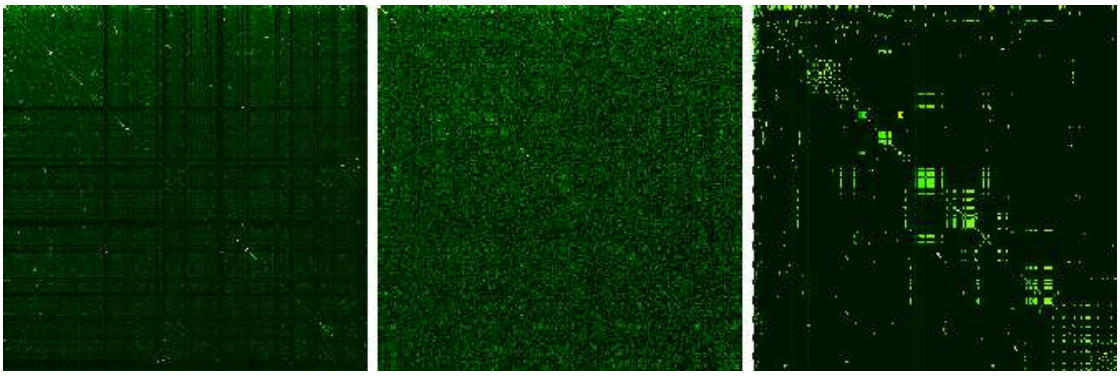

Figure 3.3: Images of a part of the G matrix for Human dna sequence network at $m = 6$ (left), Human proteome sequence at $m = 3$ (middle, see section 3.4) and Cambridge university 2006 webpages (right, image from [Ermann et al., 2012]). Matrix elements $G_{K,K'}$ are shown in the basis of PageRank index K (and K'). Here, x and y axes show K and K' within the range $1 \leq K, K' \leq 200$. The element $G_{1,1}$ at $K = K' = 1$ is placed at top left corner. Color marks the amplitude of matrix elements changing from black for minimum $(1 - \alpha)/N$ value to white at maximum value.

Statistics of The Google matrix elements

We show in Fig. 3.4 the integrated distribution of the matrix elements G_{ij} for different species in the left panel and the same quantity for *Homo Sapiens* at different word lengths in the right panel. Here N_g is the number of matrix elements such that $G_{ij} > g$ and we observe that the number of nonzero matrix elements G_{ij} is very close to N^2. The main fraction of elements has values $G_{ij} \leq 1/N$ and some elements are $G_{ij} < 1/N$ as there might be some cases where for a certain node j there exist many transitions to some node i' such that $T_{i'j} \gg N$ and very few transitions to other nodes for example only one transition to node i'' such that $T_{i''j} = 1$.

At the same time there are also transition elements G_{ij} with large values whose fraction decays in an algebraic law $N_g \approx AN/g^{\nu-1}$ with some constant A and an exponent ν. The fit of numerical data in the range $-5.5 < \log_{10} g < -0.5$ of algebraic decay are given in the following table 3.2.

Species	ν fitted values
Bos Taurus (Bull)	2.46 ± 0.025
Canis Familiaris (Dog)	2.57 ± 0.025
Loxodonta Africana (Elephant)	2.67 ± 0.022
Danio Rerio (Zebrafish)	2.22 ± 0.04
Homo Sapiens (Human)	2.48 ± 0.024
Homo Sapiens (Human) at $m = 5$	2.68 ± 0.038
Homo Sapiens (Human) at $m = 7$	2.43 ± 0.02

Table 3.2: Fitted values of the decay exponent ν of the integrated distribution of G matrix elements for various species.

There are some visible oscillations in the algebraic decay of N_g with g but in global we see that on average all species are well described by a universal decay law with the exponent $\nu \approx 2.5$.

For comparison we also show the distribution N_g for both the university of Cambridge and the university of Oxford webpages networks (reminding that they have $N \approx 2 \cdot 10^5$ with an average of 10 links per node) for which it has a very short range $-5.5 < \log_{10}(N_g/N^2) < -6$ where the decay is at least approximately algebraic contrasting with the long range in the case of DNA sequences.

Since in each column we have the sum of all elements equal to unity we can say that the differential fraction $dN_g/dg \propto 1/g^{\nu}$ gives the distribution of outgoing matrix elements which is similar to the distribution of outgoing links extensively studied for the WWW networks. Indeed, for the WWW networks all links in a column are considered to have the same weight so that these matrix elements are given by an inverse number of outgoing links [Langville and Meyer, 2006]. Usually the distribution of outgoing links follows a power law decay with an exponent $\tilde{\nu} \approx 2.7$ even if it is known that this exponent is much more fluctuating compared to the case of ingoing links. Thus we establish that the distribution of DNA matrix elements is similar to the distribution of outgoing links in the WWW networks with $\nu \approx \tilde{\nu}$. We note that for the distribution of outgoing links of Cambridge and Oxford webpages networks the fit of numerical data gives the exponents $\tilde{\nu} = 2.80 \pm 0.06$ and 2.51 ± 0.04 respectively.

On average the probability given by the PageRank vector is proportional to the number of ingoing links [Langville and Meyer, 2006], this relation is established for scale-free networks with an algebraic distribution of links when the average number of links per node is about 10 to 100 which is usually the case for Internet type of networks such as WWW, Twitter and Wikipedia articles network that have a very sparse connectivity matrix [Zhirov et al., 2010].

For the DNA sequences we find an opposite situation where the stochastic transition matrix is almost full in which case an analogue of the number of ingoing links would be the sum of ingoing matrix elements $g_s = \sum_{j=1}^{N} G_{ij}$. The integrated distribution of ingoing matrix elements with the dependence of N_s on g_s is shown in Fig. 3.5 where N_s is defined as the number of nodes with the sum of ingoing matrix elements being larger than g_s.

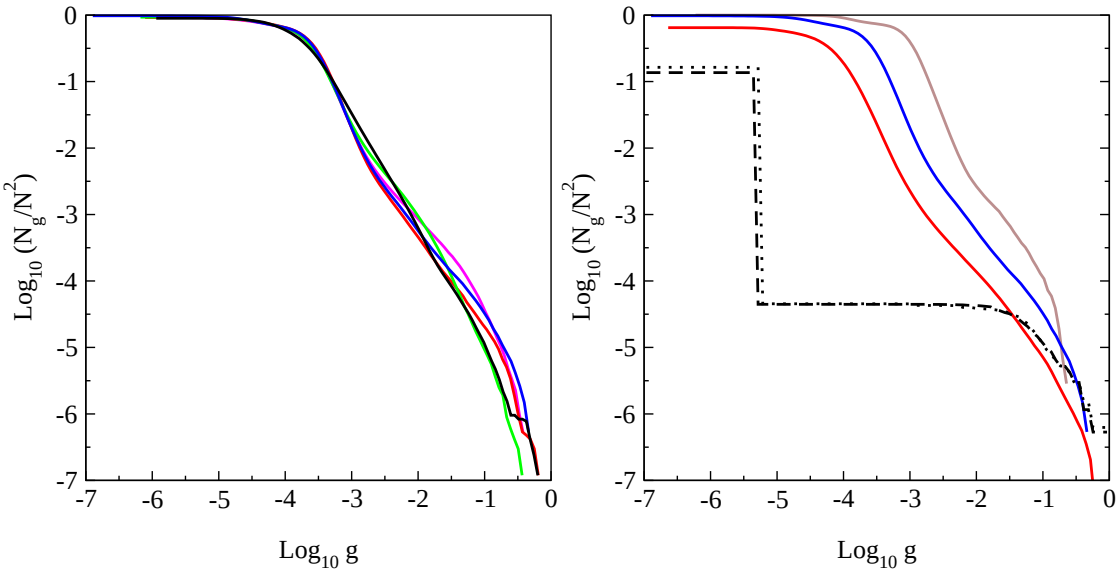

Figure 3.4: Integrated fraction N_g/N^2 of Google matrix elements with $G_{ij} > g$ as a function of g. *Left panel :* Various species with word length $m = 6$: bull BT (magenta), dog CF (red), elephant LA (green), human HS (blue) and zebrafish DR(black). *Right panel :* Data for HS sequence with words of length $m = 5$ (brown), 6 (blue), 7 (red). For comparison black dashed and dotted curves show the same distribution for the WWW networks of Universities of Cambridge and Oxford in 2006 respectively.

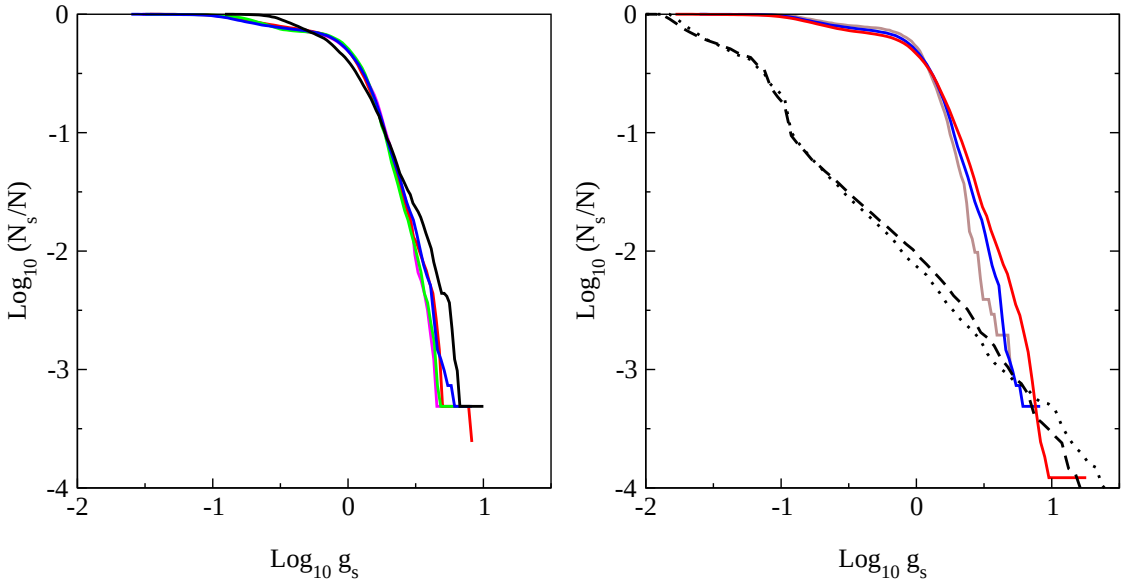

Figure 3.5: Integrated fraction N_s/N of sum of ingoing matrix elements with $\sum_{j=1}^{N} G_{i,j} \geq g_s$. Left and right panels show the same cases as in Fig. 3.4 in same colors. The dashed and dotted curves are shifted in x-axis by one unit left to fit the figure scale.

A significant part of this dependence, corresponding to large values of g_s and determining the PageRank probability decay, is well described by a power law $N_s \approx BN/g_s^{\mu-1}$. The fit of the numerical data in the range of algebraic decay are given in the following table 3.3.

Species	μ fitted values
Bos Taurus (Bull)	5.59 ± 0.15
Canis Familiaris (Dog)	4.90 ± 0.08
Loxodonta Africana (Elephant)	5.37 ± 0.07
Danio Rerio (Zebrafish)	4.04 ± 0.06
Homo Sapiens (Human)	5.11 ± 0.12
Homo Sapiens (Human) at $m = 5$	5.86 ± 0.14
Homo Sapiens (Human) at $m = 7$	4.48 ± 0.08

Table 3.3: Fitted values of the exponent μ of the sum of ingoing matrix elements for various species.

Usually for ingoing links distribution of WWW and other similar networks one finds the exponent $\tilde{\mu} \approx 2.1$. This value of $\tilde{\mu}$ is expected to be the same as the exponent for ingoing matrix elements of the matrix G. Indeed, for the ingoing matrix elements of Cambridge and Oxford webpages networks we find respectively the exponents $\mu = 2.12 \pm 0.03$ and 2.06 ± 0.02 (see curves in Fig. 3.5).

For ingoing links distribution of Cambridge and Oxford networks we obtain respectively $\tilde{\mu} = 2.29 \pm 0.02$ and $\tilde{\mu} = 2.27 \pm 0.02$ which are close to the usual WWW value $\tilde{\mu} \approx 2.1$. Thus we can say that for the WWW type of networks we have $\mu \approx \tilde{\mu}$.

In contrast the exponent μ for DNA Google matrix elements gets significantly larger value around $\mu \approx 5$. This feature marks a significant difference between DNA sequences and WWW networks. It is interesting to note that in addition to the universal linear behaviour in log scale plots we can observe some deviations which make the different species visibly distinguishable with the most pronounced one being the only non mammalian specie *Danio Rerio* considered here.

Spectral properties of DNA sequences

The eigenvalues were computed using the standard LAPACK code, it is possible to diagonalize those matrices exactly and the results are shown in Fig. 3.6 for the different species. All of them show a natural spectral gap separating $\lambda = 1$ from the other eigenvalues and we observe that only in the non mammalian case we find a small group of eigenvalues of large modulus on the real axis. As we will illustrate in the chapter 5 the large modulus eigenvalues indicate the presence of clusters of nodes that are more connected among themselves than to the rest of the network. The structural difference between the DNA sequences and the WWW types of networks is drastic as in the latter ones we find no gaps in the vicinity of $\lambda = 1$ (cf. Fig. 2.4).

In general a network with high structural complexity will have a wide eigenvalue cloud and the more it has random connections the more condensed its eigenvalue cloud will be. In the extreme case of a random stochastic matrix, apart from the dominant eigenvalue $\lambda = 1$, all the other eigenvalues are collapsed inside a circle of small radius around the origin since the second eigenvalue modulus asymptotically goes like $|\lambda_2| \propto 1/\sqrt{N}$ where N is the matrix size. In fact it is known that for asymmetric Gaussian random matrices[Mehta, 2004] the eigenvalue density in the complex plane is uniformly distributed inside a circle of radius $R = \sigma\sqrt{N}$ with σ^2 being the variance of the matrix elements and based on that principle, the random Perron-Frobenius operator models (RPFM) are discussed in [Frahm et al., 2014]. In the full matrix limit we can model the random Google matrix by drawing the matrix elements g_{ij} from a uniform probability distribution in the interval $[0, 2/N]$. The expectation value will then correspond to the mean by construction of the Google matrix $\langle g_{ij} \rangle = 1/N$ and the variance $\sigma^2 = \langle g_{ij}^2 \rangle - \langle g_{ij} \rangle^2 = 1/(3N^2)$ giving a radius of $R = 1/\sqrt{3N}$ (because $\sigma^2 = (b-a)^2/12$ for a uniform distribution in $[a, b]$).

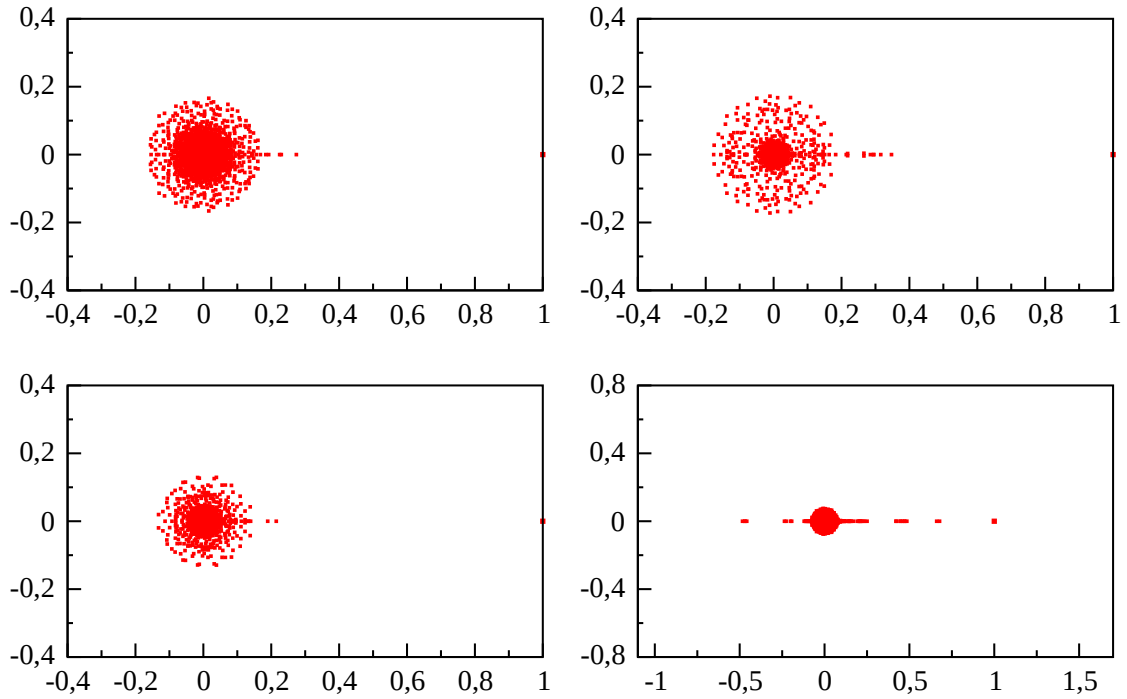

Figure 3.6: Spectrum of eigenvalues in the complex plane λ for DNA Google matrix of bull BT *(top left)*, dog CF *(top right)*, elephant LA *(bottom left)*, zebrafish DR *(bottom right)* shown for word length of $m = 6$.

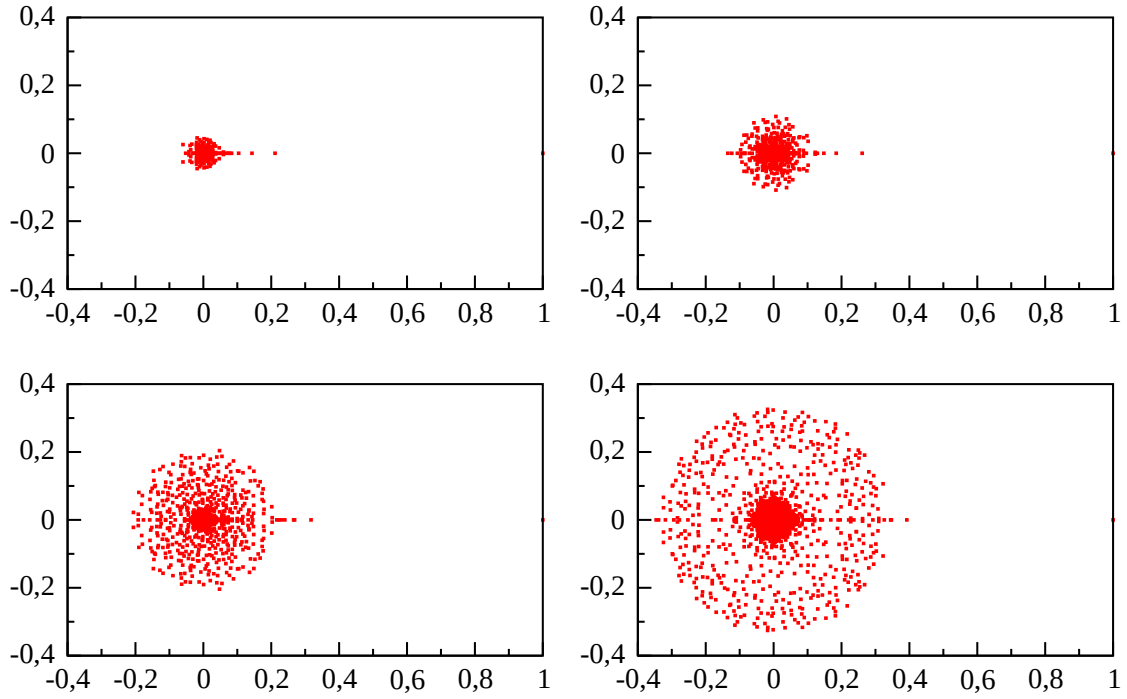

Figure 3.7: Spectrum of eigenvalues in the complex plane λ for DNA Google matrix of *Homo sapiens* (HS) shown for word length of $m = 4$ (top left), $m = 5$ (top right), $m = 6$ (bottom left) and $m = 7$ (bottom right).

At word length $m = 6$, this would correspond to a value of $R \approx 0.009$ which is too small even for the *Danio Rerio* case. The reason is that in reality the matrix elements are not following a uniform probability distribution, nevertheless as shown by the green circle in Fig. 3.8, the relation $R = \sigma\sqrt{N}$ is a good approximation with a numerically determined value of the variance σ^2. In a certain sense the spectrum of DNA sequences is similar to the spectrum of randomized WWW networks and the spectrum of the Albert-Barabási network model discussed in [Giraud et al., 2009], however as we will see in the next paragraph the properties of the PageRank vector are rather different.

Visually the spectrum is mostly similar between *Homo Sapiens* and *Canis familiaris* having approximately the same radius of circular cloud $|\lambda| < \lambda_c \approx 0.2$. For *Danio Rerio* this radius is the smallest with $\lambda_c \approx 0.1$ indicating a greater randomness in the connectivity structure. The spectrum of the G matrix can therefore allow to distinguish between mammalian and non mammalian species.

We show the effect of word length in Fig. 3.7 where the spectrum of the Google matrix for *Homo Sapiens* is plotted at $m = 4$, $m = 5$, $m = 6$ and $m = 7$. Increasing the word length leads to an increase of $\lambda_c \approx 0.03$, $\lambda_c \approx 0.1$, $\lambda_c \approx 0.2$ and $\lambda_c \approx 0.35$ respectively. This suggests that for a system of size $N = 64$ ($m = 3$) with so many transitions the chances are high that the network would resemble more to a randomized sequence giving rise to an even more condensed eigenvalue cloud and giving little insight about the statistical properties of the considered sequence. For $m = 7$ the number of nonzero matrix elements G_{ij} is close to N^2 and thus on average we have only about $L/(mN^2) \approx 8$ transitions per each element. This determines an approximate limit of reliable statistical computation of matrix elements G_{ij} for available *Homo Sapiens* sequence length L.

We verified for the *Homo Sapiens* case with word length $m = 6$ that two halves of the whole sequence L still give practically the same spectrum with a relative accuracy of $\Delta\lambda/\lambda \approx 0.01$ for eigenvalues in the main part of the cloud at $\lambda_c/3 < |\lambda| < \lambda_c$. This means that the spectra presented in Figs 3.6 and 3.7 are statistically stable at the values of L used in this work.

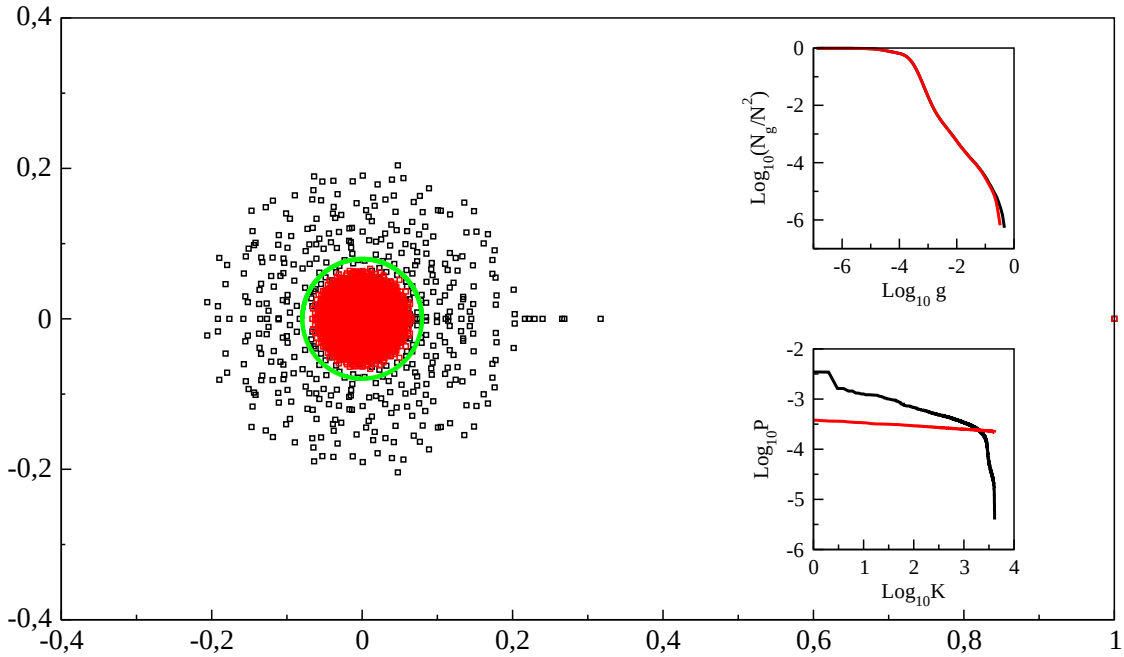

Figure 3.8: Comparison between the *Homo Sapiens* sequence at $m = 6$ (black) and a random matrix model (red). The eigenvalues of the Google matrix are displayed in the principal plot where the green circle has a radius $R = \sigma\sqrt{N} = 0.0795$ with standard deviation of the random matrix elements being $\sigma = 0.0012$. *Top inset :* Integrated distribution of the matrix elements G_{ij}. *Bottom inset :* PageRank vector probability decay.

We tried to reproduce the specificities of the eigenvalue spectrum of G and the PageRank probability decay in the context of random matrix model by generating a Google matrix whose elements G_{ij} have the same distribution N_g as for the *Homo Sapiens* sequence at $m = 6$ word length, the resulting spectrum and the PageRank decay are shown in Fig. 3.8.

We can see that all eigenvalues are homogeneously distributed in the radius $\lambda_c \approx 0.07$ being significantly smaller compared to the real data. Also in this case the PageRank probability $P(K)$ changes only by 30% in the whole range $1 \leq K \leq N$ being absolutely different from the real data. This suggests that the mere distribution of matrix elements do not account for the structural properties of the corresponding network and the organization of the links can drastically change the topology of the network.

PageRank as a measure of genetic closeness

The PageRank probability decay for various species and the effect of word length are shown in Fig. 3.9. This probability distribution describes the steady state of random walks on the Markov chain and thus gives a word ordering similar to the frequency of their appearance in the whole sequence. The frequencies or probabilities of words appearance have already been obtained in [Mantegna et al., 1995] by a direct counting of words along the available sequences which were shorter at that time, however with a significantly better statistics we find our distributions to be in good agreement with their results.

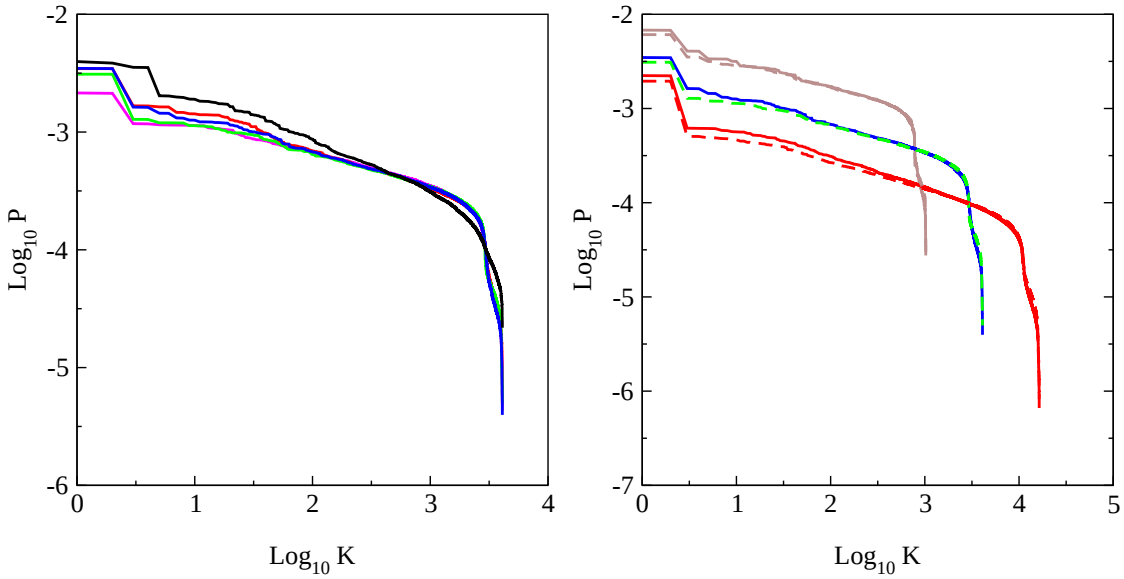

Figure 3.9: Dependence of PageRank probability $P(K)$ on PageRank index K. *Left panel :* Data for different species at $m = 6$ word length : bull BT (magenta), dog CF (red), elephant LA (green), human HS (blue) and zebrafish DR (black). *Right panel :* Data for HS (full curve) and LA (dashed curve) for word length $m = 5$ (brown), 6 (blue/green), 7 (red).

As explained in eq. 2.12 the dependence of the PageRank vector P on its ordered index K can be approximately described by a power law $P \sim 1/K^{\beta}$. For instance we find that for *Homo Sapiens* at word length $m = 7$ has an exponent of $\beta = 0.357 \pm 0.003$ in the fitting range $1.5 \leq \log_{10} K \leq 3.7$ which is rather close to the value found in [Mantegna et al., 1995].

Since on average the PageRank probability is proportional to the number of ingoing links, or the sum of ingoing matrix elements of G in our case, we have the relation between the exponent of PageRank β and exponent of ingoing links (or matrix elements): $\beta = 1/(\mu - 1)$. Indeed, for the *Homo Sapiens* DNA sequence at $m = 7$ we have $\mu = 4.48$ that gives $\beta = 0.29$ being close to the above value of $\beta = 0.357$ obtained from the direct fit of $P(K)$ dependence. The agreement

is not so perfect because of the visible curvature in Fig. 3.5. Also due to a small value of β the variation range of P is not so large thereby reducing the accuracy of the numerical fit even if a formal statistical error is relatively small compared to a visible systematic nonlinear variation. This relation between β and μ also works for the *Danio Rerio* sequence at $m = 6$ with $\mu = 4.04$ that gives $\beta = 0.33$ being in a satisfactory agreement with the fitted value $\beta = 0.426$ found from $P(K)$ dependence of Fig. 3.9.

In spite of this only approximate agreement we should say that in general the relation between β and μ works correctly. In average we find for DNA sequences networks the value of $\mu \approx 5$ being significantly larger than for the WWW networks with $\tilde{\mu} \approx 2.1$ [Langville and Meyer, 2006]. Consequently the value of $\beta \approx 0.25$ for DNA sequences is significantly smaller than the usual value for WWW, which is $\beta \approx 0.9$, and for randomized WWW networks as well as the Albert-Barabási model having $\beta \approx 1$.

The following table 3.4 shows the exponent β at word length $m = 6$ fitted in the range $1 \leq \log_{10} K \leq 3.3$.

Species	β fitted values
Bos Taurus (Bull)	0.273 ± 0.005
Canis Familiaris (Dog)	0.340 ± 0.005
Loxodonta Africana (Elephant)	0.281 ± 0.005
Danio Rerio (Zebrafish)	0.426 ± 0.008
Homo Sapiens (Human)	0.308 ± 0.005

Table 3.4: Fitted values of the exponent β of PageRank probability decay for various species.

There is a relatively small variation of β between various mammalian species. The data of Fig. 3.9 for *Homo Sapiens* shows that the value of β remains stable with the increase of word length. These observations are similar to those made in [Mantegna et al., 1995].

What are the nodes favored and those avoided by the PageRank vector ? In table 3.3 the top ten 6-letters words with largest probabilities $P(K)$ are given for all studied species where we notice that the two top words are identical for *Bos Taurus*, *Canis Familiaris* and *Homo Sapiens*. The ten most avoided words, that is the words with minimal PageRank probability, are also shown and we notice that the last two words are the same for the mammalian species but differ for *Danio Rerio*.

Top 10 PageRank entries					Last 10 PageRank entries				
BT	CF	LA	HS	DR	BT	CF	LA	HS	DR
TTTTTT	TTTTTT	AAAAAA	TTTTTT	ATATAT	CGCGTA	TACGCG	CGCGTA	TACGCG	CCGACG
AAAAAA	AAAAAA	TTTTTT	AAAAAA	TATATA	TACGCG	CGCGTA	TACGCG	CGCGTA	CGTCGG
ATTTTT	AATAAA	ATTTTT	ATTTTT	AAAAAA	CGTACG	TCGCGA	ATCGCG	CGTACG	CGTCGA
AAAAAT	TTTATT	AAAAAT	AAAAAT	TTTTTT	CGATCG	CGTACG	TCGCGA	TCGACG	TCGACG
TTCTTT	AAATAA	AGAAAA	TATTTT	AATAAA	ATCGCG	CGATCG	CGCGAT	CGTCGA	TCGTCG
TTTTAA	TTATTT	TTTTCT	AAAATA	TTTATT	CGCGAT	CGAACG	GTCGCG	CGATCG	CCGTCG
AAAGAA	AAAAAT	AAGAAA	TTTTTA	AAATAA	TCGACG	CGTTCG	CGATCG	CGTTCG	CGACGG
TTAAAA	ATTTTT	TTTCTT	TAAAAA	TTATTT	CGTCGA	TCGACG	CGCGAC	CGAACG	CGACCG
TTTTCT	TTTTTA	TTTTTA	TTATTT	CACACA	CGTTCG	CGTCGA	TCGCGC	CGACGA	CGGTCG
AGAAAA	TAAAAA	TAAAAA	AAATAA	TGTGTG	TCGTCG	ACGCGA	ACGCGA	CGCGAA	CGACGA

Table 3.5: Top ten PageRank entries (left part) and ten words with minimal PageRank probability (right part) of DNA sequences at word length $m = 6$ for species: bull BT, dog CF, elephant LA, human HS and zebrafish DR.

To observe the similarity between species on a global scale it is convenient to plot the PageRank index $K_s(i)$ of a given species s versus the index $K_{hs}(i)$ of *Homo Sapiens* for the same word i. For identical sequences all points should land on the diagonal while the deviations from the diagonal characterize the differences between species. Some examples of such PageRank proximity $K - K$ diagrams are shown in Fig. 3.10 for word length $m = 6$. Visually we get the impression that *Canis Familiaris* has the least deviations and that the non-mammalian *Danio Rerio* has the strongest

deviations from *Homo Sapiens* rank among the species compared here. For the mammalians we have a significant reduction of deviations from diagonal around $K \approx 3N/4$, this effect is also visible but less pronounced for *Danio Rerio*. It is worth to mention that those kind of rank correlation plots are heavily used and studied in statistics and copula theory in order to characterize the dependence between multiple random variables without worrying about their marginals [Nelsen, 2006].

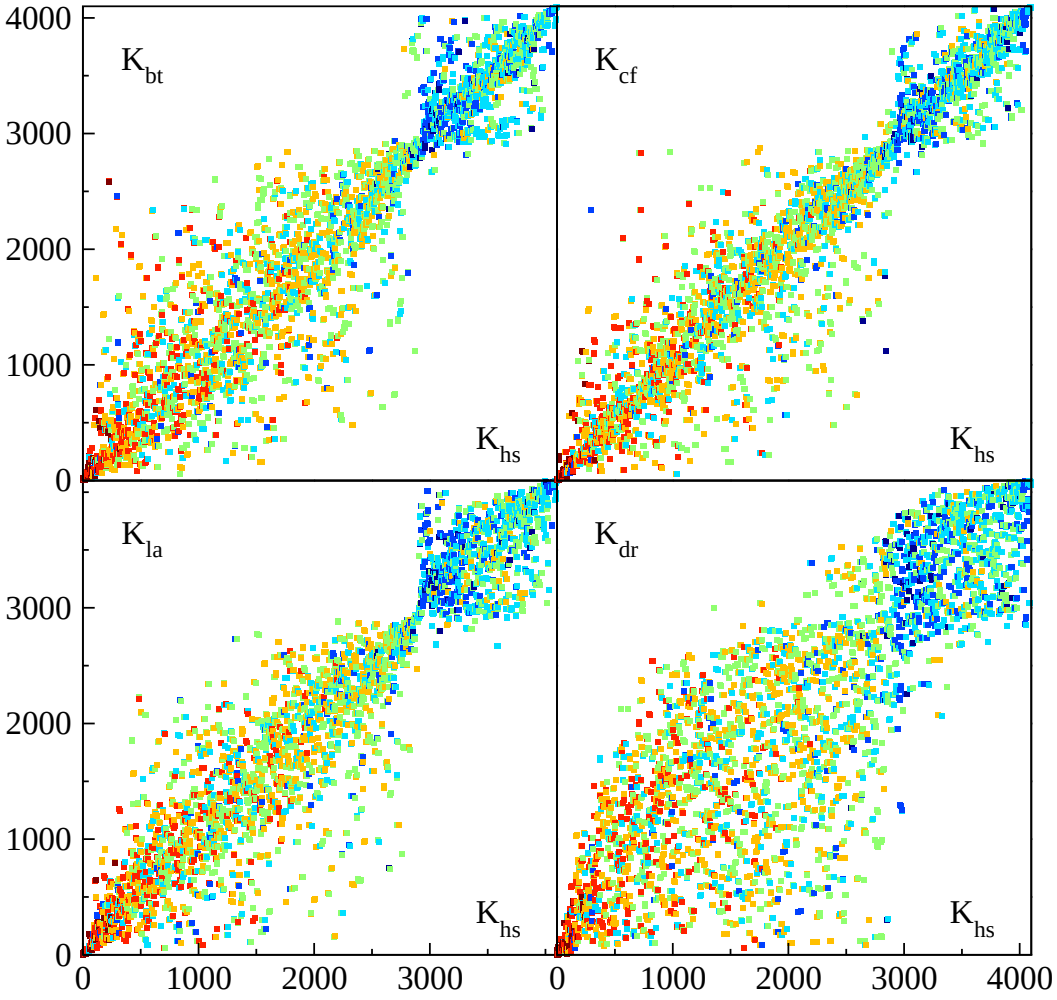

Figure 3.10: PageRank proximity $K - K$ plane diagrams for different species in comparison with *Homo Sapiens* : x-axis shows PageRank index $K_{hs}(i)$ of a word i and y-axis shows PageRank index of the same word i with $K_{bt}(i)$ of bull, $K_{cf}(i)$ of dog, $K_{la}(i)$ of elephant and $K_{dr}(i)$ of zebrafish; here the word length is $m = 6$. The colors of symbols marks the content of A or T in a word i (fractions of letters A or T in any order); the color varies from red at maximal content, via brown, yellow, green, light blue, to blue at minimal zero content.

The distribution of base content of a short sequence, that is the content of letters A or T inside a word is highlighted by the colors and shown to be inhomogeneous in K : their fraction is dominant for $1 \leq K < N/4$ where the words are mostly composed of A or T, approximately homogeneous for $N/4 \leq K \leq 3N/4$ and is close to zero for $3N/4 < K \leq N$. When classifying the word content chemically, by considering the fraction of purine letters A or G, we find an approximately homogeneous distribution over the whole range of K values.

We find that in the whole *Homo Sapiens* sequence the fractions F_a, F_c, F_g and F_t of A, C, G and T are respectively $F_a = 0.276596$, $F_c = 0.192576$, $F_g = 0.192624$, $F_t = 0.276892$ and $F_n = 0.061312$ for undetermined N_l. The fraction of A, G being close to $1/2 \approx (F_a + F_g)/(1 - F_n) = 0.499867$ and the fraction of A, T being $(F_a + F_t)/(1 - F_n) = 0.589640 > 0.5$ we have a higher probability to find A or T in the whole sequence, giving a possible explanation to the origin of the inhomogeneous

distribution of A or T along K and large fraction of A, T at top PageRank positions.

Since the whole *Homo Sapiens* sequence is composed from 5 individuals' sequences of length $L_i \approx 3 \cdot 10^9 \approx L/5$ we considered separately the first and the last fifth parts of the whole string making two independent sequences from two individuals HS_1 and HS_2, we then determined their corresponding PageRank indexes K_{hs1} and K_{hs2} and show their PageRank proximity diagram in Fig. 3.11. As expected we notice that the points are much closer to the diagonal.

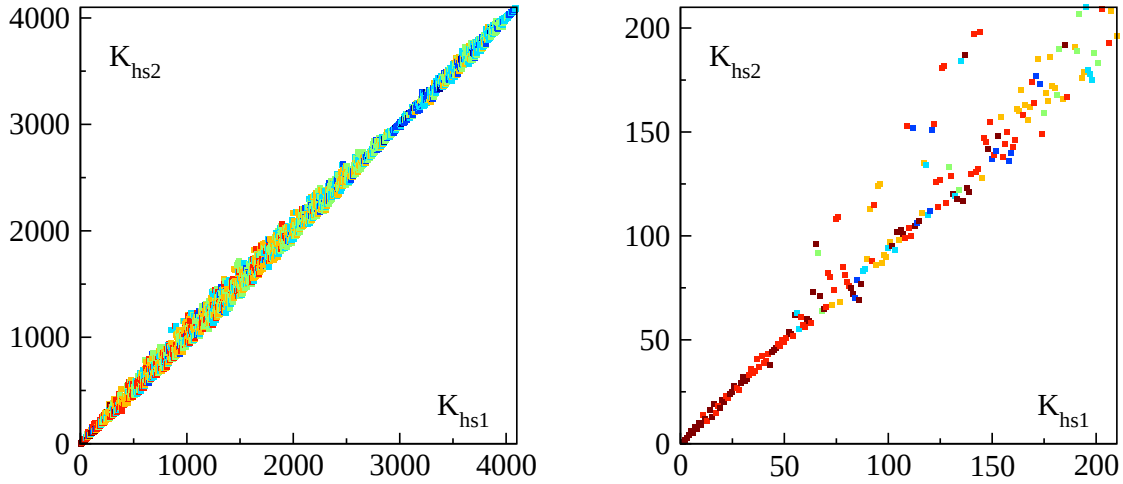

Figure 3.11: PageRank proximity $K - K$ diagram of *Homo Sapiens* HS_2 versus *Homo Sapiens* HS_1 at $m = 6$. Colors show the content of A, T in the same way as in Fig. 3.10. Right panel shows a zoom of the left panel.

To characterize the proximity between different species or different *Homo Sapiens* individuals we compute the average dispersion $\sigma(s_1, s_2)$ between two species or individuals s_1 and s_2 :

$$\sigma(s_1, s_2) = \sqrt{\sum_{i=1}^{N} (K_{s_1}(i) - K_{s_2}(i))^2 / N} \tag{3.1}$$

However this value of σ depends on the word length considered, therefore in order to represent the result in a form independent of m we compare the values of σ with the corresponding random model value σ_{rnd}. This value is computed assuming a random distribution of N points in a square $N \times N$ when only one point appears in each column and each line (for example at $m = 6$ we have $\sigma_{rnd} \approx 1673$ and $\sigma_{rnd} \propto N$). The dimensionless dispersion is then given by $\zeta(s_1, s_2) = \sigma(s_1, s_2)/\sigma_{rnd}$. From the ranking at $m = 6$ of different species we obtain the following values listed in table 3.6.

ζ	BT	CF	LA	HS	DR
BT	0.000	0.308	0.324	0.246	0.425
CF	0.308	0.000	0.303	0.206	0.414
LA	0.324	0.303	0.000	0.238	0.422
HS	0.246	0.206	0.238	0.000	0.375
DR	0.425	0.414	0.422	0.375	0.000

Table 3.6: Dimensionless average dispersion values ζ between the different species.

According to this statistical analysis of PageRank proximity between species we find that ζ value is minimal between *Canis Familiaris* and *Homo Sapiens* showing that these two are the most similar species among those considered here. For two *Homo Sapiens* individuals we find $\zeta(HS_1, HS_2) = 0.031$ which is significantly smaller than the dispersion values between two species.

3.4 The Network of Protein Sequences

In this section we will go up our scale a little bit by considering larger objects, namely the *amino acids*, which are complex chemical compounds vital to the appearance of life. If the DNA is the instruction containing the information to build an organism, the amino acids are the building blocks of the organism itself. Indeed the information contained in the nucleus of a cell in double stranded DNA format undergoes a transcription first and is translated next into amino acids which are 20 (among 22 existing amino acids) to be universally present in all life forms on Earth.

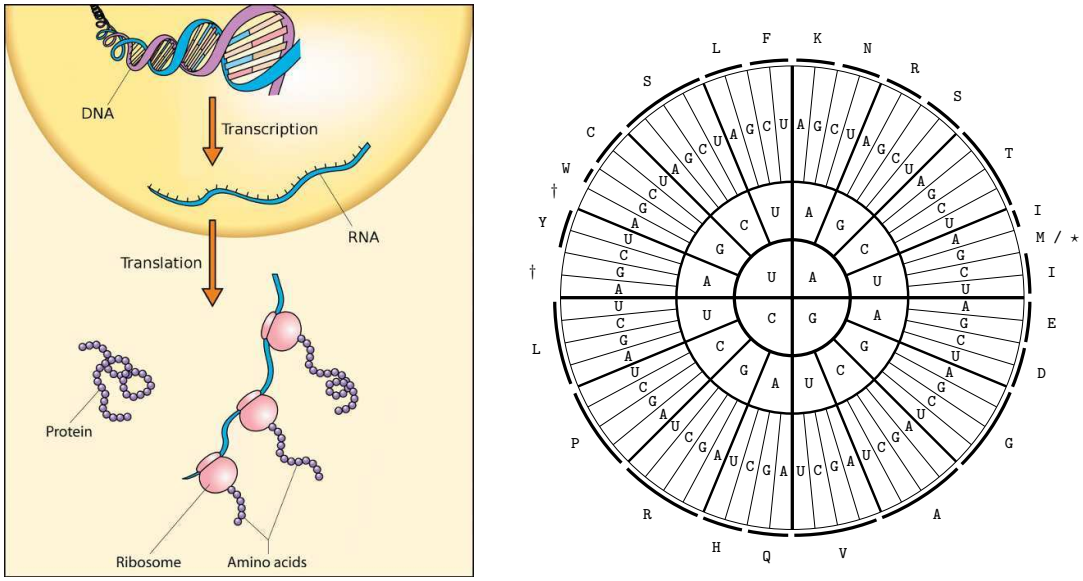

Figure 3.12: *Left :* Simplified schematic illustration of the transcription process creating RNA from DNA in the nucleus and the translation process creating an amino acid chain from RNA in the cytoplasm. *Right :* Circular table of codon to amino acid correspondence, the table is read from the center to the periphery where the path joining three letters correspond to the amino acid initial letter.

To describe the process schematically (as shown in Fig. 3.12) the transcription mechanism is performed in the nucleus where the double stranded DNA is temporarily unwound to create an opening, the area where the DNA strands are separated is free so that an enzyme, called the *RNA polymerase*, can bind itself to one of the strands and start synthesizing a single strand slightly different from the DNA strand. The synthesized chain contains the same A, C and G bases but the thymine is replaced by the *uracil* denoted by the letter U which pairs with A. The newly produced chain with A, U, C and G is called *ribonucleic acid* or RNA.

The RNA then travels out of the nucleus carrying the part of the information in the DNA that needs to be processed then the translation process takes place anywhere in the cytoplasm. A special protein complex called *ribosome* can bind itself around the RNA chain and slides along it while reading the bases three by three. A group of three letters is called a *codon* and effectively corresponds to one amino acid. The correspondence between the codons and the amino acids, which is somewhat redundant as there may be several codons associated to a single amino acid, is shown in Fig. 3.12.

By translating the RNA, the polymerase builds a chain of amino acids by concatenating them in correct order according to the current codon instructions. Once the chain of amino acid is completed and released, it undergoes a change in its conformation by folding itself in a manner specific to the chain resulting in a three dimensional molecule complex called a *protein* which finally endorses its biological functionality. The transformation from the $1D$ chain to the $3D$ structure is known as protein folding problem and is subject to intense study and huge interest but completely

beyond the scope of this work. Here we will restrict ourselves in studying the statistical properties of amino acid chains before the folding in a similar spirit to the analysis of the DNA sequences with much longer word length that would have been technically difficult to reach with the original bases chain due to the increasing size of the state space. However at the same time it is not a direct extension since only selected coding portions of the DNA sequence are translated into amino acid sequences.

In the following paragraphs we will apply the same analysis on amino acid sequences of several different *archaea* by constructing the networks in a similar manner as for the DNA sequences but this time only with a shorter word length $m = 3$, corresponding to a system of size $N = 20^3 = 8000$. These living organisms were thought for a long time to be bacterias but in fact they form a separate group among the *prokaryotes*, organisms lacking a membrane-bound nucleus [Woese et al., 1990]. The archaea share some similar characteristics with bacterias but also with *eukaryotes*, which are all the organisms (including complex animals) whose cells contain a nucleus and other organelles. These shared traits with both eukaryote and prokaryote domain of life together with some unique specificities makes the archaea a third distinct group of life which is a recent field of interest for biologists[Park et al., 2014, Gutiérrez et al., 2007].

Spectrum properties

The analysis of amino acid sequences yields results that are somewhat similar to the DNA sequences with power law behaviour of matrix elements as well as sum of ingoing matrix elements distributions. The PageRank decay rate is also quite low confirming the drastic structural differences with webpages like networks. In Fig. 3.3, the middle panel shows for qualitative comparison a part of the Google matrix for *Homo Sapiens* amino acid sequence computed with word length $m = 3$. The matrix structure resembles the one for the DNA sequence but it is less dense, indeed the sequence length here is about $L \approx 2 \cdot 10^7$ with $N_t \approx 10^6$ word transitions giving an estimate of $N_t/N^2 \approx 0.1$ transitions per matrix element.

Without discussing the statistics in depth, we will perform a qualitative systematic study of 47 different *archae* amino acid sequences of varying length[1], between $L \approx 2 \cdot 10^5$ and $\approx 5 \cdot 10^5$ letters for the organisms considered here which are listed in the following table 3.7.

1	Acidianus hospitalis	17	Methanococcus voltae	33	Pyrobaculum oguniense
2	Archeoglobus fulgidus	18	Methanoculleus bourgensis	34	Pyrobaculum sp 1860
3	Archeoglobus veneficus	19	Methanohalobium evestigatum	35	Pyrococcus sp NA2
4	Caldiarcheum subterraneum	20	Methanohalophilus mahii	36	Staphylothermus hellenicus
5	Desulfurococcus fermentans DSM 16532	21	Methanoplanus petrolearius	37	Sulfolobus solfataricus 98 2
6	Desulfurococcus fermentans DSM 2162	22	Methanosalsum zhilinae	38	Sulfolobus solfataricus P2
7	Haloarcula hispanica	23	Methanosarcina mazei Tuc01	39	Thermococcus sp 4557
8	Haloferax volcanii DS2	24	Methanospirillum hungatei	40	Thermococcus sp AM4
9	Halogeometricum borinquense DSM 11551	25	Methanothermobacter marburgensis	41	Thermofilum pendens
10	Halopiger xanaduensis SH-6	26	Methanothermococcus okinawensis	42	Thermoplasma acidophilum
11	Haloquadratum walsbyi C23	27	Methanothermus fervidus	43	Thermoplasmatales archaeon BRNA1
12	Ignisphaera aggregans	28	Methanotorris igneus	44	Thermoproteus tenax
13	Methanobacterium sp SWAN-1	29	Natrinema sp J7-2	45	Thermoproteus uzoniensis
14	Methanocaldococcus infernus	30	Nitrosoarchaeum koreensis	46	Thermosphaera aggregans
15	Methanocella conradii	31	Nitrosopumilus koreensis AR1	47	Vulcanisaeta distributa
16	Methanococcus jannaschii	32	Nitrosopumilus sp AR2	48	Homo Sapiens

Table 3.7: List of the 47 archae from which the amino acid sequences were studied, the datasets were obtained from Prof. Viktor Solovyev.

In order to compare the structural complexity of their amino acid sequences, the eigenvalues of the Google matrix (except $\lambda = 1$) of all these organisms are shown in Fig. 3.14 at $\alpha = 1$. We can observe a similarity in the global structure of the eigenvalue cloud, for all the archae we have a dense disk or radius $R < 0.4$ which are very different from the *Homo Sapiens* case where spiked structures are visible. There is also in each case a large natural gap since $|\lambda_2| < 0.5$ in each archaea.

[1]Data given by Prof. Solovyev : *http://www.molquest.kaust.edu.sa/?topic=about&no_menu=on.*

The spectra look a lot like what would be obtained from a random connectivity between the words. There is a visible difference between the radius of the disk which is due to the varying length of the different sequences considered here. A longer sequence will result in a smaller disk because there is an increased probability for every transition to possibly happen and therefore an increased similarity with a randomized sequence. Contrary to the DNA sequences case the google matrices here are not full, for a given sequence i we can say that on average there is $Q_i = N_t(i)/N$ transitions per column. Using the sparse variant of RPFM, we can compute an estimate to the radius R_i of the circle containing the eigenvalues if the sequence considered was a random chain : Supposing that there are exactly Q_i randomly placed non zero elements per column, their values being constant $1/Q_i$, the variance will be given by $\sigma^2 = 1/Q_i N$ and therefore the radius $R_i = \sigma_i\sqrt{N} = 1/\sqrt{Q_i}$. The circle of radius R_i for each archaea is plotted in green in Fig. 3.14.

Highlighting the archaea according to their classes shows that some similarities exist withing a group, for instance *Desulforococcales* have very similar eigenvalue spectrum but it is only on a statistical level and the archaea genus might be very different from each other inside a specific class. However on a finer level there are some notable differences in the outer structure of the disks. We can distinguish three main situations : organisms having a simple and clear dense eigenvalue disk and nothing else such as *Archaeoglobus fulgidus*, organisms having complex distribution of eigenvalue around the edge of the disk such as *Methanohalobium evestigatum* and organisms having some eigenvalue of large modulus on the real axis (for example *Halopiger xanaduensis*) or having a cycle such as *Thermoplasmatales archaeon* (see chapter 5 for cycles).

As discussed in the previous chapter, the eigenvalues of largest modulus give some interesting insight about the structural organization of the underlying directed network. In order to compare the complexity of the symbolic sequences of amino acid independently of the length of the sequence, we can simply estimate how far they are from their randomized counterpart. The Fig. 3.13 shows in the left panel the number of eigenvalues whose modulus is larger than the RPFM radius $N_{|\lambda|>R_i}$ versus the mean distance separating these eigenvalues from the circle $\Delta = \langle |\lambda_j| - R_i \rangle_j$ for each archaea sequence i. It seems that *Methanospirillum hungatei* (24) or *Sulfolobus solfataricus* (37) for example have a large number of eigenvalues that are on average quite distant from the RPFM circle suggesting that they have a complex structural organization.

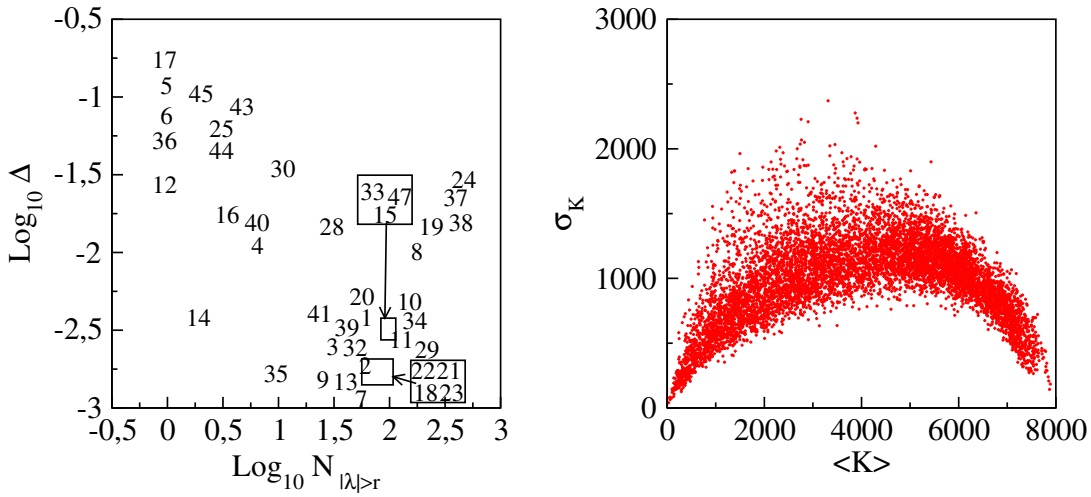

Figure 3.13: *Left panel :* Number $N_{|\lambda|>R}$ of eigenvalues outside the RPFM circle versus the mean distance Δ from the edge of the circle; the number show the ID of the archaea listed in table 3.7 and the missing sequences have no eigenvalues larger than RPFM radius R. *Right panel :* Mean PageRank index versus the standard deviation for each of the $N = 8000$ 3-letters word, the mean and variance are computed using all the 47 archaea sequences.

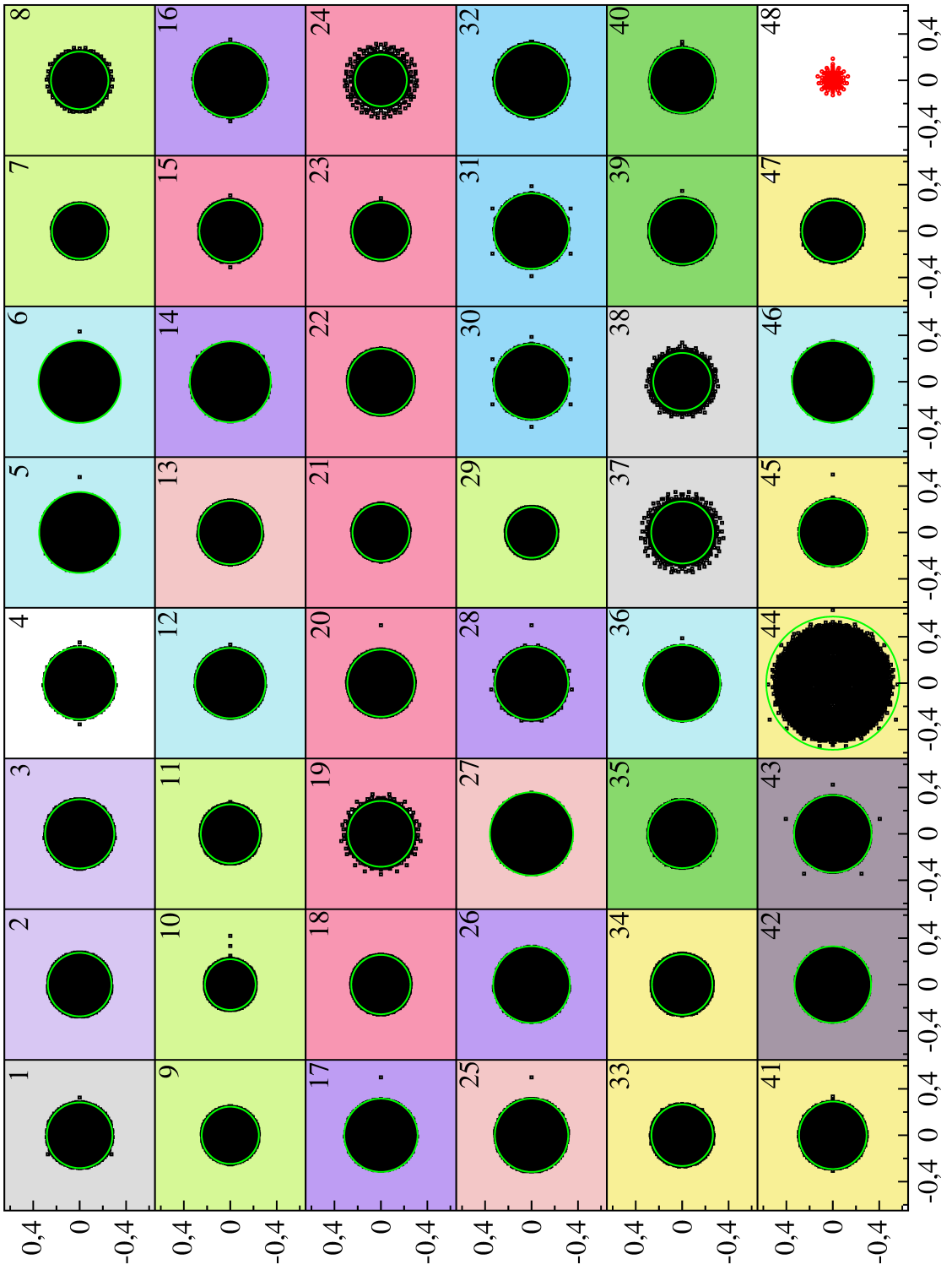

Figure 3.14: Eigenvalue clouds of the Google matrices constructed with amino acid triplets at $\alpha = 1$. Several archae (see table 3.7 for names) are shown in black and *Homo Sapiens* is shown in red. The green circle of archae i corresponds to the radius $R_i = \sigma_i\sqrt{N}$. The colors correspond to the different classes of *Archaea* domain : *Archaeoglobi* (light purple), *Halobacteria* (light olive), *Methanobacteria* (light pink), *Methanococci* (purple), *Methanomicrobia* (pink), *Thermoplasmata* (grey), *Thermococci* (green), *Nitrosopumilales* (cyan), *Thermoproteales* (yellow), *Sulfolobales* (light grey), *Desulfurococcales* (light cyan). The first eigenvalue $\lambda_1 = 1$ is never shown.

PageRank based Phylogenetic trees

In the table 3.15 the top 20 entries of the PageRank vector for the 47 different organisms are shown along with the top entries for the *Homo Sapiens* sequence. In general the latter one contains a lot of repetitive letters especially in top PageRank entries which are different from the archae case where some words are not even in the top entries of any archae sequences. We can also see some top words similarity between some groups of archae. The right panel of Fig. 3.13 shows the average rank K for each word versus its standard deviation σ_K computed from the 47 samples, we can see that the words that are highly ranked on average have low dispersion meaning that they tend to be in the top positions for all the sequences. Similar behaviour is visible for very low ranked words but there are great variations in the central region. The idea is to use that information brought by the PageRank vector to build a similarity diagram between the various archaea based on their amino acid sequences.

In biology a *phenetic* tree is a tree graph representing a classification of organisms based on their morphological characteristics or a specific trait. This classification can also be performed using a molecular sequencing dataset in order to represent schematically the closeness between two organisms taking into account the evolutionary process. The phylogenetic tree shows to which extent two species share a common origin and allows to explore their ancestry on a genetic level.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	KLE	ELK	KIE	ISI	LEL	ISL	LSL	IEL	KLK	LSI	EKI	LKK	IKL	KLI	ILL	EIK	KKI	LIG	KEI	LLL
2	ELE	KLE	ELK	EAK	VEL	EVK	LEE	LEG	EVE	LEL	EIE	AEL	ALA	ELR	AEA	EEL	KEV	IEL	AAA	LEV
3	ELE	KVE	LEL	EVE	IEL	EVK	KLE	EIE	EAK	ELK	VEL	ALA	AEL	AAA	AIA	EIK	LEG	VAL	KIE	LEV
4	AAA	LEL	AAL	LKL	LLL	ELK	GVV	VEL	LAL	ALG	ELE	LGV	LEV	LRL	LEG	AAV	LAA	ALA	EAK	VEV
5	LLL	LKL	IEL	LEL	ELK	ILL	RLE	LIL	LLI	LGL	KLE	LEG	LLG	LSL	VEL	LSI	LDL	LIG	VRL	LAL
6	VEL	RLE	LEL	ELE	LGL	LEG	EAR	LLL	VLL	VVG	VVE	ALA	VGL	VAL	GVV	LAL	ARL	GLL	LRL	VRL
7	AAA	ALA	AEA	AAL	AVA	LAA	AAG	AAE	LAL	EAE	GAA	VAA	AAD	DVA	AGA	ELE	AGV	GVA	LVA	EVD
8	AAA	ALA	AVA	AAG	VAA	AAV	LAA	AAD	AAL	EAE	AEA	GAA	VAV	DAA	LAL	EVA	EAA	ADA	ADD	GVA
9	AAA	ALA	VAA	AVA	AEA	EAE	AAG	AAL	AAV	LAA	ELE	AAE	VAV	VAL	DLE	AAD	VAG	AEL	AVV	GVA
10	AAA	EAE	ELE	AEA	ALA	AAG	EAA	LAA	AVA	EEE	AAE	LAL	AAV	GAA	EVE	DLE	AAL	VAA	DDE	AEG
11	AAA	AVA	TTT	ELE	AAV	LAA	DDD	ALA	AAG	AEA	LGL	GAA	AAL	TAA	AGA	DLE	VAA	EAA	EAE	ALG
12	III	LII	ILI	IIS	ISI	KII	IKL	ELK	ISL	IIL	LEL	SII	ILL	LIL	IIV	IKI	IRI	LLI	IRL	ELI
13	ELK	KLK	LLL	KLE	KIE	EIK	KLD	LIL	KVE	LGI	ELE	IGL	ILI	IVG	LKL	IKI	LEL	LKG	VLL	IEL
14	ELK	EIK	KLE	KIE	KLK	KIK	LKL	ELE	EIE	KKI	IEI	LLK	KLL	IKK	KKL	KLI	KKE	LEK	IEK	IKI
15	AAA	ALA	LAL	AAL	AAG	LLL	LAA	AAV	LLG	LEL	LAG	ALG	VAL	LIL	LVA	VGL	LVG	GLA	LKG	GLG
16	ELK	EIK	KIE	KLE	KIK	IKI	IKL	IEL	KLK	LEL	ELE	KKI	IKK	LKK	EIE	EKK	KKK	KKL	IEI	IEK
17	NNN	KLE	ELK	KLK	KIK	EIK	NKN	NIN	IKL	KIE	KLN	IEL	LNI	LKL	KLI	LNL	EIE	KII	NKI	KEI
18	AAA	ALA	AAL	LAL	LAA	AVA	VAL	AGA	ELR	LAG	AAG	RLE	ALG	GAA	LEL	AAV	VAA	EAA	GLA	ALE
19	KLK	IEL	EIE	EIK	LDI	ELK	LSL	EEE	DLK	ISI	LNL	LDL	ELE	EIS	INL	KLE	EAK	LEE	DEI	KLI
20	LEG	ELE	LEE	LEL	LLL	LEI	ELK	IAL	ILL	LLI	EAE	ELS	IEL	LLG	AIA	GVG	EIE	ALG	VVG	ISI
21	ELE	AAA	IAI	IEL	GIG	GIL	LEG	IGI	LEI	LAL	IGL	ILL	AAG	LLI	EIE	ILG	LAG	VAL	GIS	LEL
22	LEL	LSI	EIK	LEI	ISL	IAI	KLE	LIG	IEV	KIE	ELD	EIS	ELK	ISI	SEI	IEI	IDI	EIE	IDL	LLL
23	ELK	ELE	LLL	LEL	LSL	KLE	LKL	EEL	ELL	LEG	LEE	LAL	AEL	EIK	ELS	LLG	EAK	IEL	KLL	LGL
24	LLG	LLL	IAL	LIG	LDI	IGI	ISL	IEL	AEL	LEL	AAA	ALA	LGI	LEI	LLI	ELR	ELS	LSL	GLL	AAL
25	ELE	EIE	AAA	IEL	RLE	LEL	LEE	ELR	GVG	RAE	RIE	EVE	EAE	KLE	AEL	LVG	EEE	VIG	LAA	LAL
26	KIK	ELK	EIK	IKL	KKI	KLE	KIN	KIE	LKI	IKI	KLK	NKI	KII	KKK	ILI	IEL	KLI	EIE	IGI	LKL
27	KLK	KIK	KKK	KKI	IKK	KIE	EIK	KLE	ELK	IKL	KAK	LKL	LKK	KGK	KVK	IKI	LKG	KGI	KEI	ELE
28	EIK	ELK	KLE	KLK	IKI	KIK	KIE	LKL	EIE	IEI	KKI	IKL	IEL	ELE	EKK	KKL	KKK	LKI	LEL	IKK
29	AAA	AAG	ALA	LAA	AAL	AEA	AAV	AVA	VAA	EAE	VAL	LAL	AGA	EVE	ADA	GAA	ARA	VAV	EAA	AAE
30	KIK	KLE	KIE	IKI	KLK	KLL	IKL	LKL	EIK	ELK	KLD	SLE	KKS	KIL	IDL	LKK	IEI	SLS	IGI	KKL
31	KIK	KLE	KIE	IKI	KLK	KLL	IKL	LKL	EIK	ELK	KLD	SLE	KKS	KIL	IDL	LKK	IEI	SLS	IGI	KKL
32	KIK	ELK	KLK	KIE	IKL	KLE	IKI	EIK	IEI	LKS	ISI	KKI	LKK	EIE	LKG	KKK	LSI	SEI	LKI	KIS
33	AAA	LAA	AAL	ALA	AAV	VAV	LAL	ALL	AVA	AAG	ELK	AGA	EAR	LLL	LVA	LKL	LAV	LEL	VAL	VAA
34	AAA	AAL	ALA	LAL	LAA	ALL	ARA	AAG	VAA	LRL	ELR	AVA	LEL	LAV	LLL	EAR	AEA	VVG	ALG	LAG
35	ELE	ELK	IEL	KLE	EIE	VEL	LLL	LEL	EIK	ILL	EAK	LEI	LEG	ELR	LLE	KIE	EEK	LKL	RIE	EVK
36	KLE	IEI	LKL	ELK	KLK	LEI	EIK	IKL	LEL	LIL	LKI	ILI	ELE	IEL	ISI	IAI	LSI	LSL	VEI	LIG
37	KLE	LKL	LLL	LSL	LIL	ELK	IKL	ISL	KIE	ILL	LKG	KLI	ISI	LEL	KLK	KLL	LEK	KVE	ILI	LGI
38	KLE	LLL	KLK	LIL	ELK	KIE	LKL	LSL	LEL	IKL	ISL	ILL	LKG	VLL	IEL	LSI	LEI	IEI	KVE	ISI
39	ELE	LEL	LLL	ELK	LAL	LGL	RLE	LLG	ELR	KLE	LLV	LEG	ALA	VEL	VAL	LLA	ALL	GLL	ALG	EEL
40	LEL	ELE	ELK	KLE	LLL	LAL	ELR	RLE	LEG	LGL	ALL	AEL	EAE	LKL	LRL	LLG	GVV	EVE	EEL	AAA
41	ELE	LEL	LLL	ELK	ELR	KLE	LEG	LRL	LAL	RLE	LLA	LGL	ALG	EIE	VAL	EAK	EVE	VEL	LVL	LKG
42	IEL	ISI	ISL	SIS	LSI	IDI	IAL	ILI	IVS	IAI	LIA	RLE	IGS	IGL	GIG	VSL	LAA	GIV	ASA	AII
43	AAA	AAV	AIA	AEA	GIA	LLA	DLD	IAA	AAG	AVG	VGV	AAL	AGA	LAG	VVG	LAL	GLG	VGG	ELK	VLG
44	AAA	AAG	LLG	ALA	LAA	LAL	ELR	AEA	AAL	EAR	LEG	RLE	ALL	LPL	LAG	GVA	AVA	LLL	ELK	LGL
45	AAA	ALA	AAL	LAA	LAL	LLA	LAG	AAG	ELR	AAV	AVA	AGA	VAA	ALL	VAL	GLA	AGL	EAA	ALG	AAR
46	LEL	LKL	ELK	KLE	ELE	LVG	VEL	LSL	LLL	VEV	GVV	EVK	LRL	LEV	IEL	LLV	RLE	ILL	ELL	GVL
47	LLL	ILI	VGV	LLI	LRL	ALI	AAA	LKL	LSL	LIL	LNL	LEL	LLG	AAL	ALL	LAL	IEL	ILL	GVV	ELR
48	SSS	LLL	EEE	PPP	LSL	LEL	SLS	AAA	LLS	LAL	GGG	SLL	ELK	LLA	KLE	LKL	ALL	LRL	LLG	LSS

Figure 3.15: Top 20 PageRank entries of the 47 archaea. The top 20 entries from the *Homo Sapiens* sequence are shown in the 48th row.

There are several possible strategies to build a phylogenetic tree : statistics based methods (maximum parsimony, maximum likelihood,...) or distance based methods (UPGMA, Neighbour joining, ...). Here the distance based methods are the most suited and simple option to begin with.

To generate such a tree for N organisms we need to obtain a square $N \times N$ pairwise distance

matrix quantifying the degree of closeness for each pair of organisms. A good candidate would be the proximity measure ζ introduced in eq. 3.1. Indeed the proximity measure satisfies naturally the following properties required for a distance metric, for all s_1, s_2 and s_3 in the set of considered organisms : **1)** non-negativity $\zeta(s_1, s_2) \geq 0$, **2)** equality $\zeta(s_1, s_2) = 0$ if and only if $s_1 = s_2$, **3)** symmetry $\zeta(s_1, s_2) = \zeta(s_2, s_1)$ and **4)** triangle inequality $\zeta(s_1, s_2) \leq \zeta(s_1, s_3) + \zeta(s_3, s_2)$.

A distance based algorithm typically tries to aggregate nearest nodes or group of nodes into a higher level cluster and compute the distance between the new cluster and the other nodes. This process is iterated until reaching the biggest cluster containing all the initial nodes of the system[Sokal and Michener, 1958]. The simplest and earliest algorithm is known as UPGMA, unfortunately this method is proven to get the wrong tree topology in many cases [Holland, 2006]. In fact the UPGMA method is inconsistent for datasets that are not ultrametric for which the metric property **4)** is replaced by the stronger version **4)** strong triangle $\zeta(s_1, s_2) \leq max\{\zeta(s_1, s_3), \zeta(s_3, s_2)\}$.

This ultrametric property being not satisfied here we use the neighbour joining method which is consistent in any case and is still algorithmically simple and fast [Saitou and Nei, 1987]. The result is shown in Fig. 3.16 where the tree naturally captures to some extent the various classes of archaea.

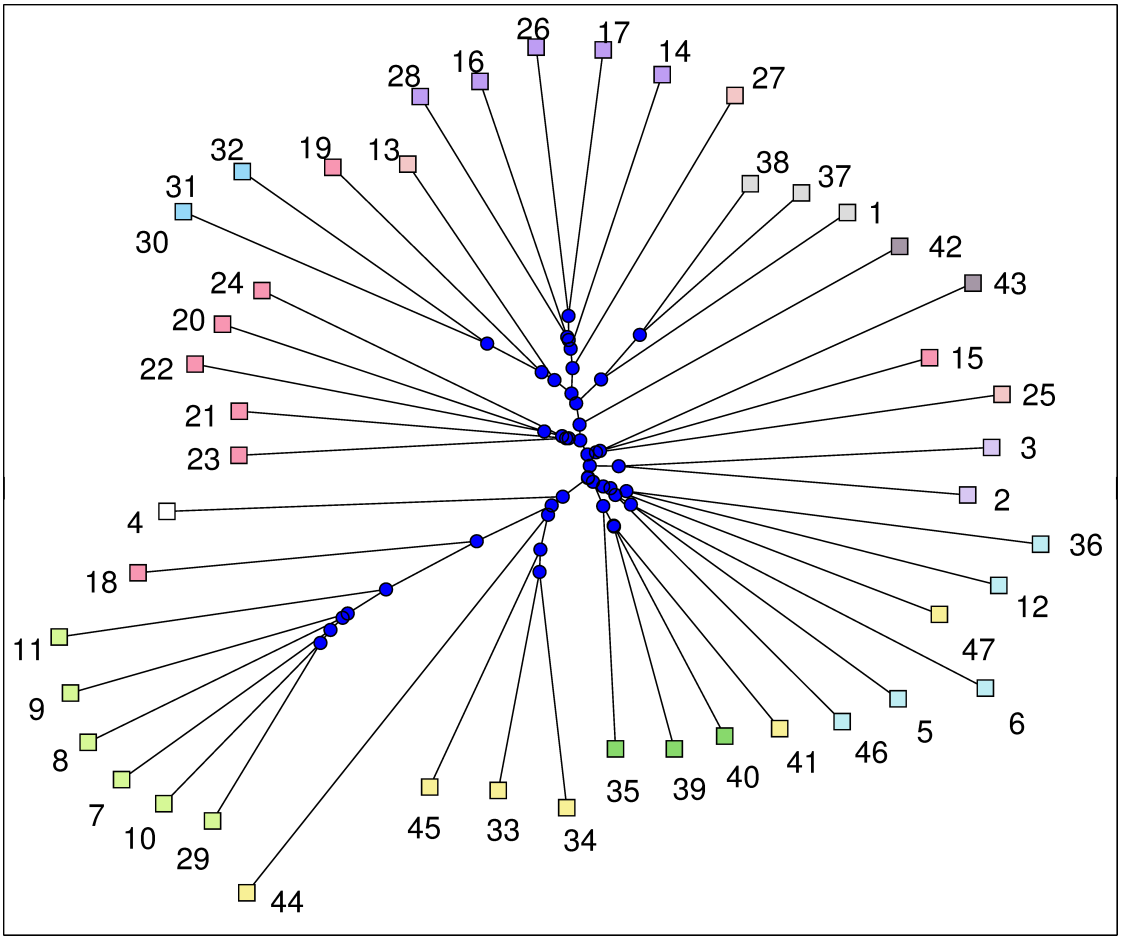

Figure 3.16: Phylogenetic tree of 47 bacteria generated with neighbour joining method. The distance matrix used is based on rank differences of amino acid triplets between different archae whose ID are given in table 3.7. The colors correspond to the different classes of *Archaea* domain : *Archaeoglobi* (light purple), *Halobacteria* (light olive), *Methanobacteria* (light pink), *Methanococci* (purple), *Methanomicrobia* (pink), *Thermoplasmata* (grey), *Thermococci* (green), *Nitrosopumilales* (cyan), *Thermoproteales* (yellow), *Sulfolobales* (light grey), *Desulfurococcales* (light cyan).

3.5 Conclusion

Through this chapter we have seen how the Google matrix method is applicable in a different context than for the Internet network by analysing DNA sequences of several species. We have shown thanks to the statistics of matrix elements, the spectrum and the dominant eigenvector analysis that the structural differences between the usual webpages networks and the DNA sequences networks are quite striking. We have established that the distribution of matrix elements of the DNA sequences are similar to the outgoing links distribution in WWW networks but in contrast the sum of ingoing matrix elements in the DNA case, which is similar to the distribution of ingoing links, shows a significantly faster decay than its Internet counterpart leading to a slow decay of the PageRank vector. We have observed that the DNA sequences networks have a natural spectral gap that is not present in WWW spectrum and we have seen that the eigenvalue cloud varies drastically between mammalian and non mammalian species. Despite the fact that the PageRank entries are close to the frequency distribution of the words, we have suggested that it could be used as a measure of closeness between species in the directed network framework. We have also briefly mentioned the possible extension to amino acid chains and the use of the proximity measure given by the PageRank in order to build a phylogenetic tree.
Despite its usefulness and its efficiency in probing the large scale-free networks it is also important to recall the limitations of the Google matrix method and in general the directed network approach to some problems. Here for instance the problems involving the detection of motifs and rare words for example may require the use of several different word lengths at once or the length of a word might be unknown. The detection of mutations would also be a difficult task nevertheless the hope is that further advances using the directed network point of view could be useful in providing new insights about the problems.
Most of the work is numerical computation because analytical results on complex networks are hard to provide. However we have seen that the specificities in the DNA sequences are such that a simple random matrix model cannot reproduce the spectrum features and the PageRank probability decay found in the real datasets. Therefore the challenge of developing a simple random matrix model that can reproduce the behaviour of the real sequences still remains.

Even in the definition of the nodes and edges in DNA sequence we have explored only the simplest case by assuming adjacent words to be linked, it is in principle possible to extend this idea to more complicated relationship between the nodes such as introducing a spacing between the considered words and studying the effects induced by the variations of the spacing may also bring some interesting insights.

Regarding the amino acid sequences a more advanced comparative work would help in understanding the specificities highlighted by the Google matrix and PageRank proximity correlation when constructing the phylogenetic tree and plenty of possibilities remain unexplored.

Chapter 4

The network of *C.elegans* neurons

4.1 Generalities on Neurons and the *C.elegans* worm

Continuing our ascension on the scale of systems that can be approached from a complex networks point of view we will now consider objects at the cellular level.

One of the biggest mystery of modern day science is the consciousness : despite huge advances in understanding the machinery of Life both at the molecular and the biological level, there are still many open questions about the origin of the consciousness and the process of thought. It is clear that the proper functioning of complex organisms are controlled by their brain via the nervous system but this monitoring process in evolved animals such as human beings are far from being trivial. Parts of automated functional activities of the brain such as responses to particular stimulus, treatment of visual information, subconscious reflexes for example are on the way of being understood but a large part of higher functionalities are difficult to grasp. It is generally suggested that phenomenon as complex as the consciousness might arise from the collective behaviour and countless interactions between a very large number of neurons forming the brain. Maybe the awareness emerges from complex relationship and information flow between the numerous individual cells, which are nothing more than a biological computational unit when taken separately, meaning that the essence of higher functions partly arises from the network structure of the neurons.

Understandably there is a lot of interest put into the research on the human brain and large scale projects aiming at simulating a whole brain with all its individual neurons are emerging [BBP, 2014, Izhikevich, 2007]. In order to successfully implement and study the brain it is crucial to understand structural properties of the neural network, therefore the complex network approach can prove useful in several contexts related to the neural systems[Eguíluz et al., 2005].

The brain is an organ composed of particular cells, the neurons, that are made of the main body, called *soma*, containing the nucleus and most of the common compounds usually found in other types of cells. In addition those cells have a long extension, called *axon*, whose extremity ends with several branches connecting to other neurons through *synapses*. In order to receive information from other cells the soma is surrounded or extended in receptor filled endings called *dendrites*. A schematic view of a typical neuron is shown in Fig. 4.1 along with a fluorescence image of actual neurons of mouse cerebral cortex taken from [Lee et al., 2005]. The neurons are so designed to receive electrical excitations from other neurons, process the signals and send an electrical response to the neurons it is connected to. In fact due to a peculiar chemical balance maintained by ion channels and ion pumps there is a potential difference across cell membrane so that in resting state the cell is polarized. The neurons have the properties to be excited in certain area up to a certain threshold value of electric potential above which it responds with a massive depolarization. This change of potential, called the *action potential* (cf. Fig. 4.1), travels down the axon towards the *synaptic connections* in order to be propagated further away.

In our work it is useful to distinguish three types of neurons : The *sensory neurons* whose dendrites receptors respond to mechanical constraints, light exposure or temperature variations

for example making the neuron sensitive to the environmental conditions. These neurons can therefore be excited by external stimulus thereby providing information about the surroundings. The *motor neurons* have their axon connected to muscular fibers so that they can direct movements of the organism. The *interneurons* are the intermediate cells receiving signals from input cells and processing them before exciting other neurons, they serve as modulators and signal relays.

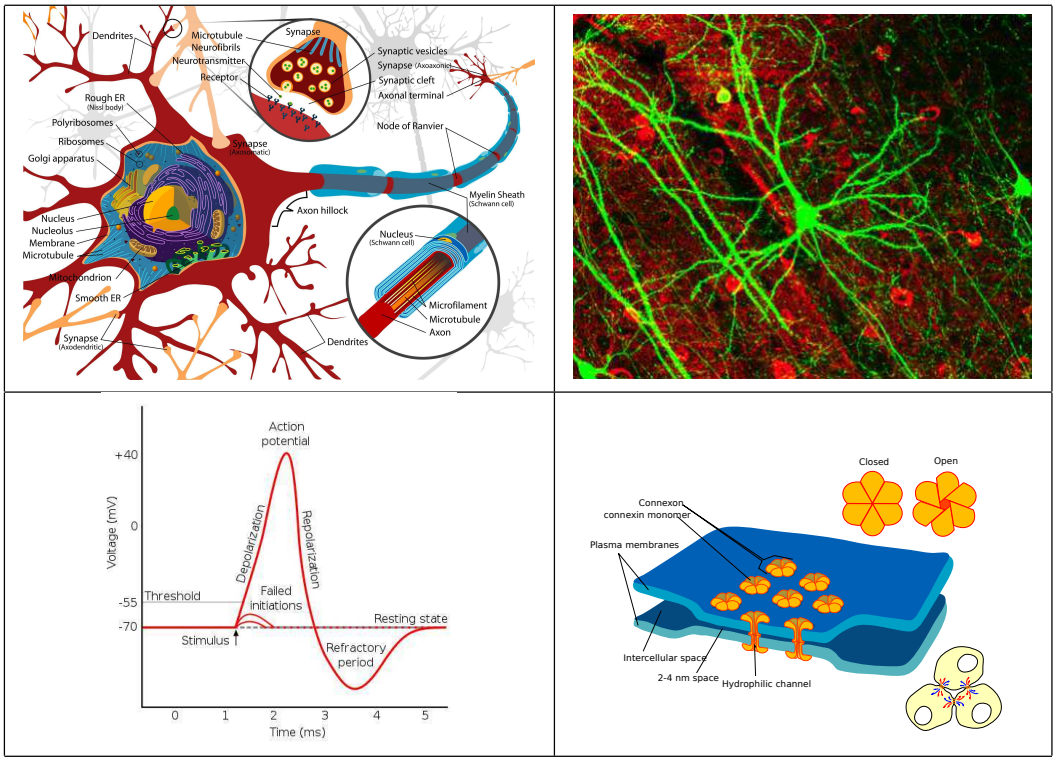

Figure 4.1: *Top left :* Schematic view of a typical neuron cell configuration. *Top right :* Protein fluorescence image of neurons in mouse cerebral cortex. *Bottom left :* Description of the action potential. *Bottom right :* Illustration of a gap junction. (*Pictures from Wikipedia Commons*).

It is very difficult to probe the neuron connectivity without any kind of invasive tool, also the large number of neurons in the human brain makes it hard to draw a precise cartography of neuron relationships. Only for small animals the process becomes possible and currently the one fully known nervous system is the one from a tiny earth worm of about a millimeter long, the *Caenorhabditis elegans* whose image is shown in Fig. 4.2 along with a schematic description of its anatomy below. This transparent worm typically lives in the ground and serves widely as a model organism to study various cellular mechanisms becoming here again a Nobel prize offering field. Moreover it is also simple enough to conduct a comprehensive analysis of neuron functionality and connectivity so that people have made a complete database at [Wormatlas, 2013].

Motivation : Although a complex network approach to the neuron network is obvious, the directed graph point of view is not so common. A similar study where the considered network was a simulation has been done in [Shepelyansky and Zhirov, 2010], here we suggest analysing the only real dataset of neuron connectivity and compare its topological properties with other known networks. Despite the very small size of the system it is still interesting to see what aspects are captured by our method with the hope that progresses on small systems will help with the understanding of the human brain.

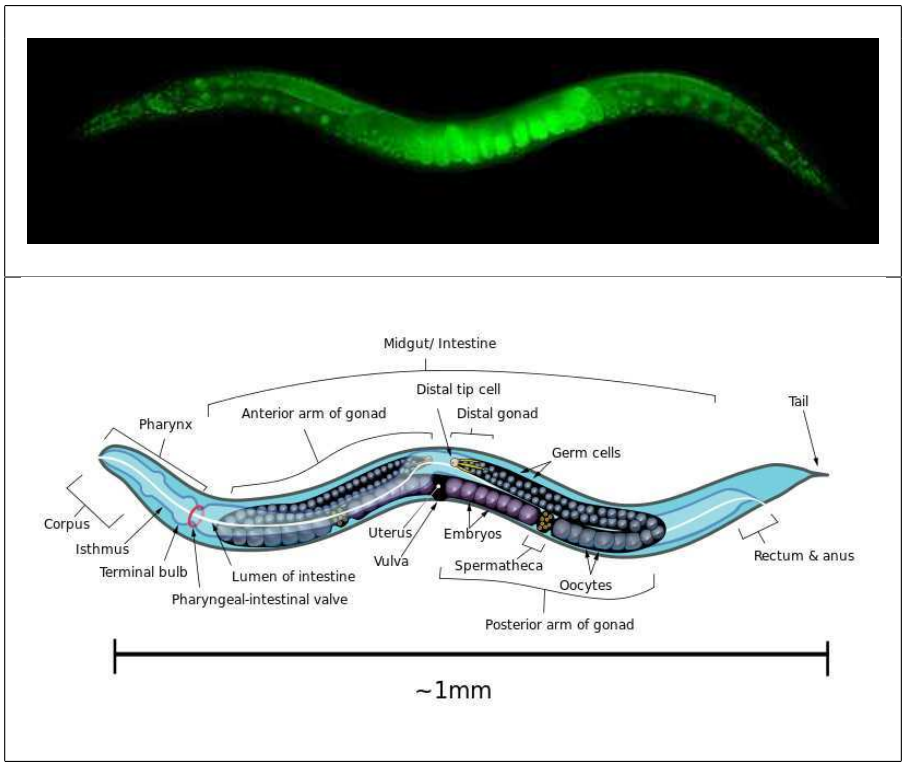

Figure 4.2: Fluorescence image of the *C.elegans* worm (top) with a schematic view of its anatomy (bottom). (*Pictures from Wikipedia Commons*).

4.2 The Network of Neurons

There are 302 neurons in total in the *C.elegans* worm out of which a few are pharyngeal, having no direct connections via synaptic links to the larger part constituting the nervous system. We will only consider the larger part made of $N = 279$ neurons which are the nodes of our network. To define the links we will consider the two following situations :

First, an impending action potential along the axon activates the release of the *neurotransmitters* which are special chemical compounds that induce the excitation of a neuron by falling on the receptors at the dendrites extremities. Whether or not the receiving neuron will fire depends on several factors such as the quantity of neurotransmitters received and the frequency of the signals but it is interesting to note that the entire chemical process of signal conduction and transmission is fully understood and a minimal model reproducing the exact behaviour of a single neuron is possible [Izhikevich, 2007].

Here the directed part comes from the synaptic connexions described by an asymmetric matrix of size 279×279 whose elements $S_{syn,ij} = 1$ if the axon of the neuron j connects to dendrites of neuron i representing the direction of signal propagation and $S_{syn,ij} = 0$ otherwise. We do not consider the multiplicity of the synapses connecting to multiple dendrites therefore S_{syn} is a binary matrix.

Second, in addition to the synaptic links the neurons might be in direct contact with neighbouring neuron cells through dendrites or even the soma in which case a communication between them also exists thanks to the *gap junctions*, illustrated in Fig. 4.1, which are channels embedded in the cell membrane so that adjacent cells can exchange compounds without piercing the membrane.

We describe the gap junctions by a symmetric matrix of size 279×279 whose elements $S_{gap,ij} = S_{gap,ji} = 1$ if neurons i and j are adjacent and communicating through membrane channels, the flow of compounds are bilateral.

We define our connectivity matrix by $S = S_{syn} + S_{gap}$ and following the instructions to build the Google matrix we end up with :

$$G_{ij} = \frac{\alpha S_{syn,ij} + \alpha S_{gap,ij}}{\sum_i (S_{syn,ij} + S_{gap,ij})} + \frac{\alpha}{N} d(j) + \frac{1-\alpha}{N} \tag{4.1}$$

with $\alpha = 0.85$ and $\mathbf{d}$ being the vector indicating the dangling nodes.

4.3 G and G^* : the network and the inverted network

So far we have mainly considered the network as it is without any kind of modifications once it is properly defined. Let us now introduce a new concept that will prove to be extremely useful for various situations : the inverted network. The idea is to simply reverse the direction of all the links as illustrated in the small example in Fig. 4.3.

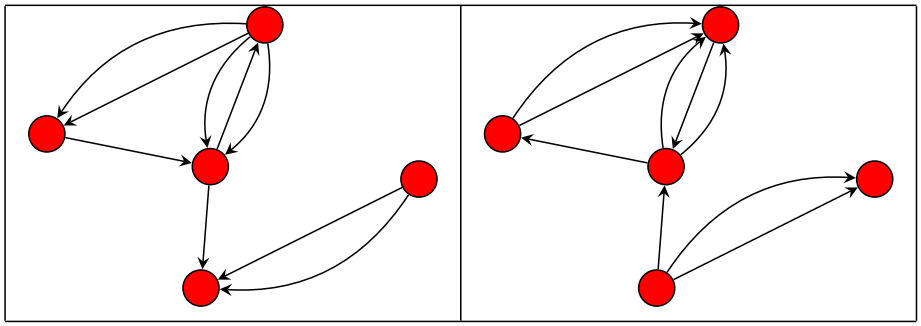

Figure 4.3: Example of a directed network represented by a google matrix G and its inverse directed network that would be represented by the google matrix G^*.

By doing so we have an additional network for which we build the Google matrix following the standard procedure, as explained in chapter 2, which will be referred to as G^*. In fact it is mathematically equivalent to construct G^* by using the transposed connectivity matrix S^T before the column normalization and the replacement of the dangling nodes.

The dominant eigenvector of G^* corresponding to $\lambda = 1$ is called the *CheiRank* vector P^* to differentiate from the PageRank vector P defined for G in the original network and the ranking index K is denoted by K^* in the case of CheiRank. The PageRank usually highlights the most influential nodes, those that are authoritative in a sense, in a complementary way the CheiRank highlights more communicative nodes. Therefore CheiRank is not merely the tail part of the PageRank but gives a different classification of nodes that is structurally informative.

This idea was beautifully illustrated in an application to wikipedia articles in [Zhirov et al., 2010] where for instance the rankings of personalities articles yielded interesting results demonstrating the complementarity of both rankings. Indeed while the PageRank top entries were mostly highlighting well known politicians and powerful rulers : **1)** Napoleon I of France, **2)** George W. Bush, **3)** Elizabeth II of the United Kingdom, **4)** William Shakespeare, **5)** Carl Linnaeus, **6)** Adolf Hitler, **7)** Aristotle, **8)** Bill Clinton, **9)** Franklin D. Roosevelt and **10)** Ronald Reagan, the CheiRank on the contrary gave the following entries : **1)** Kasey S. Pipes, **2)** Roger Calmel, **3)** Yury G. Chernavsky, **4)** Josh Billings (pitcher), **5)** George Lyell, **6)** Landon Donovan, **7)** Marilyn C. Solvay, **8)** Matt Kelley, **9)** Johann Georg Hagen and **10)** Chikage Oogi, showing its capacity to highlight artists, scientists and sportsmen and interestingly those personalities are lesser known to the common people. In this ranking framework, a deeper analysis of the cultural aspect of human knowledge was done in[Eom et al., 2014].

Spectrum and Eigenvectors

The global matrix structure is asymmetric both in G and G^* leading to a complex spectrum of eigenvalues as shown in top panel of Fig. 4.4. The imaginary part of the eigenvalues is distributed in a range $-0.2 < Im\lambda < 0.2$ which is narrower than for the networks of Wikipedia and UK universities. The flattening effect towards the real axis is related to a significant number of symmetric links which are mostly coming from the gap junctions, indeed a real symmetric matrix has only real eigenvalues. On the other hand the networks of Le Monde or Python have comparable width for $Im\lambda$ [Ermann et al., 2013].

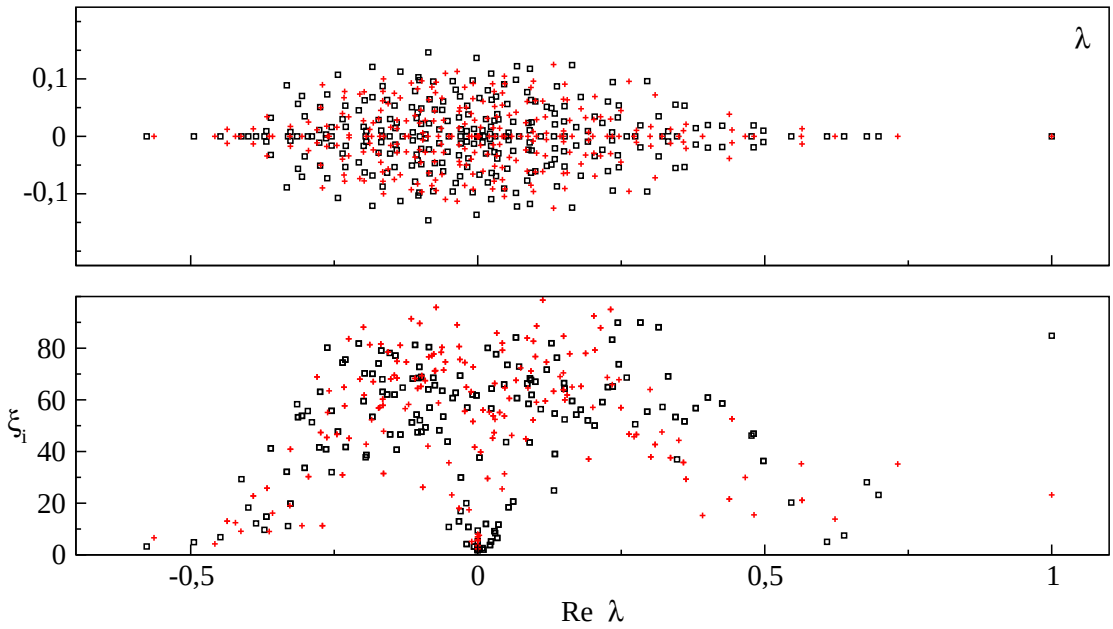

Figure 4.4: Top panel: spectrum of eigenvalues λ for the Google matrices G and G^* at $\alpha = 0.85$ (black and red symbols). Bottom panel: IPR ξ of eigenvectors as a function of corresponding $Re\lambda$ (same colors).

Considering the physicist's point of view can be helpful in finding different representations. Drawing analogies with concepts from other areas such as condensed matter physics might provide a useful insight in network science. Here are two of those concepts that can be interesting to note in our case :

Relaxation time

When a system is tossed out of a stable state by some perturbation it tends to go back to the equilibrium. This return is called the relaxation and is not instantaneous, generally the relaxation time τ is modeled by an exponential law $e^{-\tau}$. In chapter 2 we have seen that the convergence rate for a Markov chain in general is given by the ratio of the two largest eigenvalues by modulus which therefore means that $e^{-t/\tau} = |\lambda_2/\lambda_1|^t$ and $\tau = -1/ln\lambda_2$.

Inverse Participation Ratio

In solid state physics localization phenomenon of particles and quasiparticles in disordered medium are widely studied, since the physical objects are often described by eigenstates ψ_i of some matrix the inverse participation ratio (IPR) $\xi_i = \left(\sum_j |\psi_i(j)|^2\right)^2 / \sum_j |\psi_i(j)|^4$ provides a normalization independent measure characterizing the localization length of the considered objects. In our context of directed network it represents an approximate number of nodes over which the eigenstate lies or so to say the number of nodes where a significant part of the probability is located. For a vector uniformly spread over P vertices it would be equal to P, a random vector thus has an IPR proportional to the size of the system. On a side note, it is

argued in [Giraud et al., 2009] that there are two regimes for the PageRank vector where in one case it is localized within a finite number of sites when the system grows and the other case where increasing the network size N results in an increase of the localization length. The IPR would therefore be a useful quantity to probe a potential transition from localized to delocalized regime of the PageRank vector which would manifest as a flat probability decay hence a questionable ranking.

In our *C.elegans* network we find that the second by modulus eigenvalues are $\lambda_2 = 0.8214$ for G and $\lambda_2 = 0.8608$ for G^* corresponding to an approximate relaxation time of $\tau \approx 5$ and $\tau \approx 6.7$ iterations of G and G^* respectively.

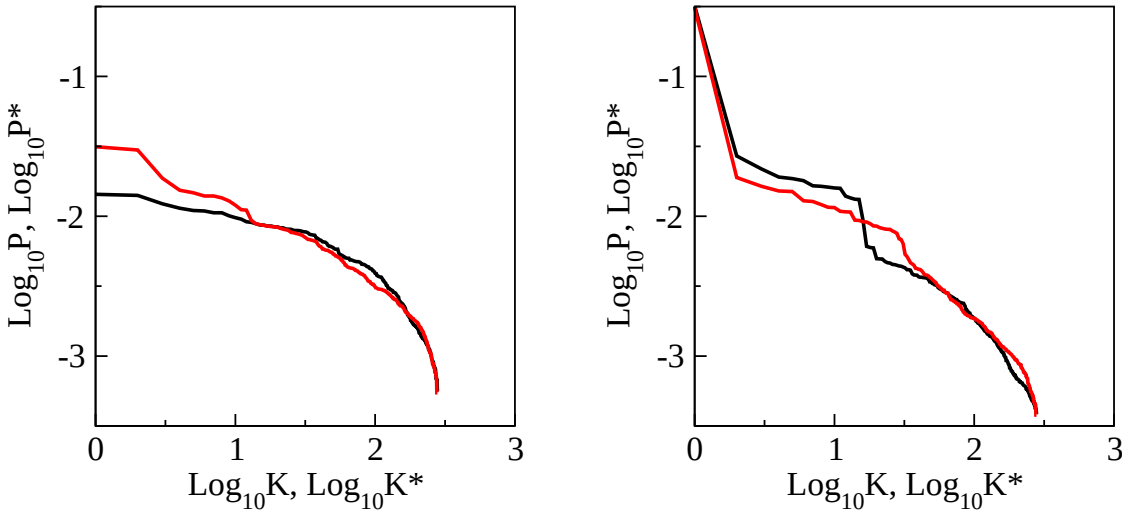

Figure 4.5: *Left panel:* dependence of PageRank (CheiRank) probability $P(K)$ ($P^*(K^*)$) on its index K (K^*) shown by black (red) curve. *Right panel:* dependence of ImpactRank probability P (P^*) on its index K (K^*), obtained via propagator of G (G^*) at $\alpha = 0.85$ and $\gamma = 0.7$ for the initial probability located on neuron RMGL (see text).

The dependence of PageRank and CheiRank probability vectors on their indexes K and K^* is shown in Fig. 4.5. A formal fit for a power law dependence $P \propto 1/K^\nu, P^* \propto 1/K^{*\nu}$ in the range $1 \leq K, K^* \leq 200$ gives $\nu = 0.33 \pm 0.03$ for PageRank and $\nu = 0.50 \pm 0.03$ for CheiRank. Of course we should be careful with those values, the number of nodes is very small compared to the WWW or Wikipedia networks but roughly we can say that a power law provides a satisfactory description of data tendency. We note that the values of ν are notably smaller than the usual exponent value $\nu \approx 0.9$ (in K), 0.6 (in K^*) found for the WWW or Wikipedia networks. Also in our neural network we find that the exponent in K is smaller than in K^* while usually one finds the opposite situation. Moreover we have that the IPR $\xi \approx 85$ for P and $\xi \approx 23$ for P^* so that comparatively the PageRank is distributed over a larger number of neurons. It is possible that such an inversion is related to a significant importance of outgoing links in neural systems: in a sense such links transfer orders, while ingoing links bring instructions to a given neuron from others. We note that somewhat similar situation appears for networks of Business Process Management (BMP) where *Principals* of a company are located at the top CheiRank position while the top PageRank positions belong to company *Contacts* [Abel and Shepelyansky, 2011].

PageRank-CheiRank correlation

It is very useful to consider the correlation between both PageRank and CheiRank distributions, as explained in [Ermann et al., 2012] to quantify the dependence we need to consider the joint probability $P(\rho, \rho^*)$ of finding a node i in an area around $(\rho(i), \rho^*(i))$ and define the correlator κ :

$$\kappa = N \sum_i P(i) P^*(i) - 1 \tag{4.2}$$

When the probability distributions P and P^* are independent the correlator is zero $\kappa = 0$. The following table gives a few examples of values of correlator κ taken from [Ermann et al., 2012] :

Networks	N	κ
E.Coli gene transcription	423	-0.0645
Linux Kernel V2.6	285510	0.022
C.elegans	279	0.125
Business Process Management	175	0.164
Cambridge webpages 2006	212710	1.71
Wikipedia English articles 2009	3282257	4.08

Table 4.1: A few examples of different types of networks with their size N and their κ correlator value.

For *C.elegans* neuron network the value of correlator is relatively small compared to those found for Wikipedia and WWW of UK universities indicating that in a sense the situation is more similar to the networks of Linux Kernel and BMP, thus the *C.elegans* network has practically no correlations between ingoing and outgoing links. It is argued in [Chepelianskii, 2010] that such a network structure allows to perform a control of information flow in a more efficient way, in other words it allows to reduce the propagation of errors in software codes and it is also suggested that networks that are most likely used for storage purposes have a high correlation whereas networks that are purely functional are nearly uncorrelated. It seems that the neural networks also adopt such a structure.

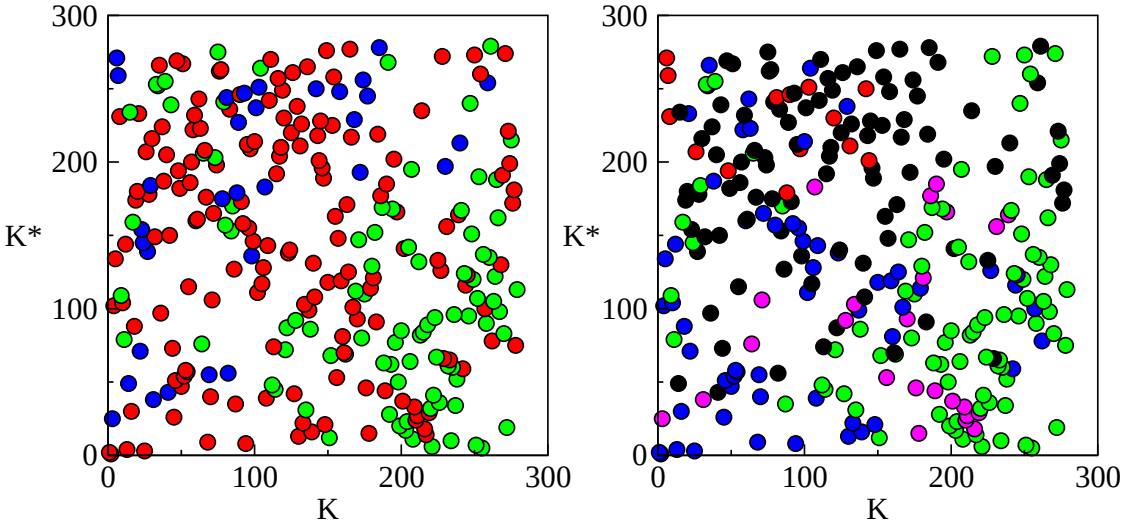

Figure 4.6: PageRank - CheiRank plane (K, K^*) showing distribution of neurons according to their ranking. *Left panel :* soma region coloration - head (red), middle (green), tail (blue). *Right panel :* neuron type coloration - sensory (red), motor (green), interneuron (blue), polymodal (purple) and unspecified (black). The classifications and colors are given according to [Wormatlas, 2013].

2D Plane Representation

Computing the dominant eigenvector of G and G^* on the same system with same node definition gives us two probability distributions and therefore two complementary rankings K_i and K_i^* for each node i. Similarly each neuron i belongs to two ranks K_i and K_i^* and it is convenient to represent the distribution of neurons on the two-dimensional plane (2D) of PageRank-CheiRank indexes (K, K^*) shown in Fig. 4.6.

This kind of plot gives a global view of the dependence between both rankings and greatly helps identifying key features of the directed network. It is easier to determine whether some nodes are highlighted by both rankings or avoided by both of them and also those who are avoided by one and highlighted by the other. We note that if both rankings are strictly the same the points would align up on the diagonal.

In our case the plot confirms that there are little correlations between both rankings since the points are scattered over the whole plane. Neurons ranked at top K positions of PageRank have their soma located mainly in both extremities of the worm (head and tail) showing that neurons in those regions have important connections coming from many other neurons which control head and tail movements. This tendency is even more visible for neurons at top K^* positions of CheiRank but with a preference for head and middle regions. In general neurons that have their soma in the middle region of the worm are quite highly ranked in CheiRank but not in PageRank. The neurons located at the head region have top positions in CheiRank and also PageRank, while the middle region has some top CheiRank indexes but rather large indexes of PageRank (Fig. 4.6 left panel). The neuron type coloration (Fig. 4.6 right panel) also reveals that sensory neurons are at top PageRank positions but at rather large CheiRank indexes, whereas in general motor neurons are in the opposite situation similar to those neurons having their soma in the middle part of the worm.

	PR	CR	2DR	EOPR	EOCR	IMPR	IMCR
1	AVAR	AVAL	AVAL	PHAL	AS07	RMGL	RMGL
2	AVAL	AVAR	AVAR	PHAR	VA10	URXL	AVAL
3	PVCR	AVBR	AVBL	VC04	AS08	ADEL	ASHL
4	RIH	AVBL	AVBR	FLPL	AS10	AIAL	AVBR
5	AIAL	DD02	PVCR	ASKL	DB06	IL2L	URXL
6	PHAL	VD02	AVKL	ASKR	DB05	ADLL	AVEL
7	PHAR	DD01	PVCL	AVFL	AS01	PVQL	RIBL
8	ADEL	RIBL	PVPR	AVG	VA02	ALML	RMDR
9	HSNR	RIBR	RIGL	PVPL	DA07	ASKL	RMDL
10	RMGR	VD04	PVPL	RIFR	VA03	CEPDL	RMDVL
11	VC03	VD03	RIS	PQR	VD03	ASHL	AVAR
12	AIAR	VD01	AVDR	VC05	AS09	AWBL	SIBVR
13	AVBL	AVER	RIGR	AVJL	VA06	SAADR	AIBR
14	PVPL	RMEV	AVDL	PVQR	VA03	RMHR	ADAL
15	AVM	RMDVR	AVKR	RIFL	VD02	RMHL	RMHL
16	AVKL	AVEL	RIBR	ASHR	DA06	RIH	AVBL
17	HSNL	VD05	DVC	VD13	VA05	OLQVL	SIBVL
18	RMGL	SMDDR	AIBL	AIMR	AS04	AIML	ASKL
19	AVHR	DD03	DVA	AVHR	AS06	HSNL	RID
20	AVFL	VA02	AVJL	PVPR	DD01	SDQR	SMBVL

Figure 4.7: Top twenty neurons of PageRank (PR), CheiRank (CR); 2D Rank (2DR); Equal Opportunity PageRank (EOPR) and CheiRank (EOCR); ImpactRank of G (IMPR) and G^* (IMCR) at initial state RMGL at $\gamma = 0.7$; following [Wormatlas, 2013], the colors mark: interneurons (blue), motor neurons (green), sensory neurons (red), polymodal neurons (purple).

The top 20 neurons of PageRank and CheiRank vectors are given in the first two columns of Fig. 4.7. We note that both rankings favor important signal relaying neurons such as AVA and AVB that integrate signals from crucial nodes and in turn pilot other crucial nodes. Neurons $AVAL, AVAR$, $AVBL, AVBR$ and $AVEL, AVER$ are considered to belong to the rich club analyzed in [Towlson et al., 2013].

4.4 2DRank, EqOpRank and ImpactRank

PageRank and CheiRank vectors provide in our context the two most straightforward classifications of nodes, in principle we can combine them or define new rankings depending on the specificities we are looking for. Without thorough investigation we propose the following possible rankings :

2D Rank : To capture the nodes that are both very influential and very communicative at the same time we can simply combine both rankings by using the 2DRank index K_2, explained in the Fig. 4.8 bellow taken from [Zhirov et al., 2010], which counts nodes in order of their appearance on ribs of squares in (K, K^*) plane with the square size growing from $K = 1$ to $K = N$. The top neurons in K_2 are AVAL, AVAR, AVBL, AVBR, PVCR. Thus at the top K_2 values we find dominance of interneurons as shown in the $2DR$ column of Fig. 4.7.

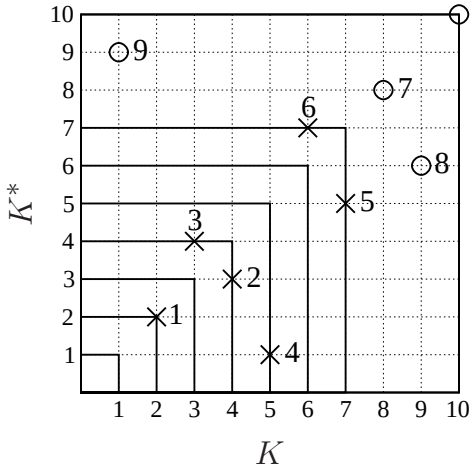

Figure 4.8: A toy example illustrating the functioning of 2DRank algorithm of node ranking in (K, K^*) plane: square size $k \times k$ is regularly increased in size $k \to k + 1$ (the current depicted iteration is $k = 7$), nodes appearing on edges of this square at each step are included in the listing K_2 of 2DRank (crosses), first on right edge, then on top edge; nodes outside of the square (circles) are included in the listing K_2 at later stage. Numbers near symbols give K_2 values of 2DRank. *(Figure from [Zhirov et al., 2010]).*

Equal Opportunity Rank : It may be also useful to consider renormalized equal opportunity rank recently discussed in [Bánky et al., 2013]. In this approach PageRank probability of node i is replaced by $P(i)/d(i)$ where $d(i)$ is in-degree of node i. For the Google matrix this recipe should be replaced by $P(i) \to P(i)/\sum_j G_{ij}$ and respectively for CheiRank by $P^*(i) \to P^*(i)/\sum_j G^*_{ij}$. The corresponding rank indexes K and K^* rank the neurons in the decreasing order of these renormalized probabilities. The distribution of nodes in the plane (K, K^*) is shown in Fig. 4.9. In this ranking the top K nodes correspond to important sensory neurons rather than information relaying centers, whereas the top nodes of K^* are composed mainly by motor neurons as shown in $EOPR$ and $EOCR$ columns of Fig. 4.7. Thus such an approach allows to highlight additional features of *C.elegans* network, being complementary to PageRank and CheiRank properties discussed above, by substracting the effect of hub nodes.

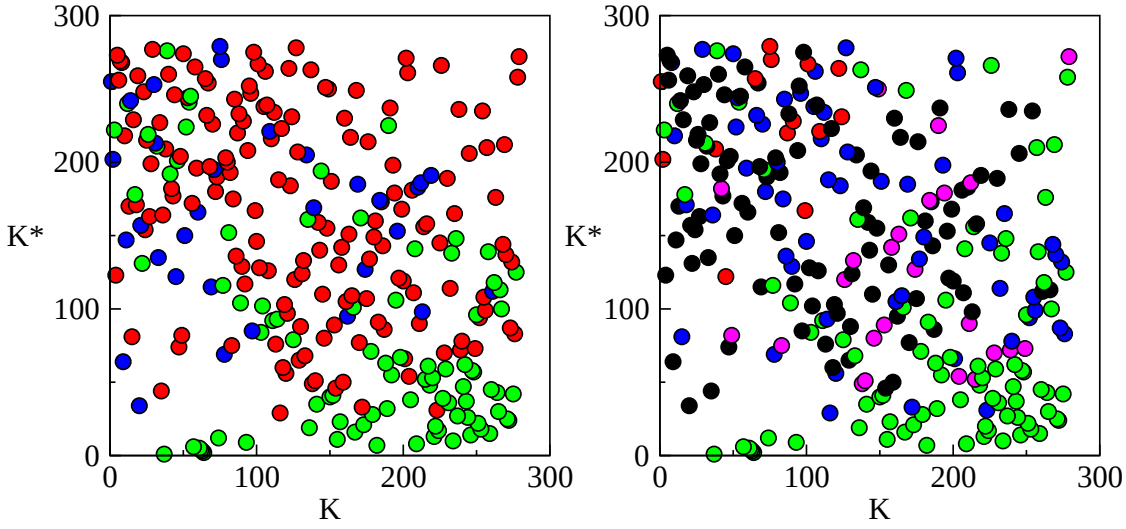

Figure 4.9: Distribution of neurons in the plane (K, K^*) of equal opportunity ranks. *Left panel :* soma region coloration - head (red), middle (green), tail (blue). *Right panel :* neuron type coloration - sensory (red), motor (green), interneuron (blue), polymodal (purple) and unspecified (black). The classifications and colors are given according to [Wormatlas, 2013].

Impact Rank : In certain cases it is useful to determine the influence or impact of a given neuron on other neurons and how it propagates through the network. A recent proposal of ImpactRank, described in [Frahm et al., 2014], is based on the probability distribution of a vector $\mathbf{v_f}$ (or $\mathbf{v_f^*}$ for its inverted network counterpart) :

$$\mathbf{v_f} = \frac{1-\gamma}{1-\gamma G}\mathbf{v_0} \tag{4.3}$$

$$\mathbf{v_f^*} = \frac{1-\gamma}{1-\gamma G^*}\mathbf{v_0} \tag{4.4}$$

where $\mathbf{v_0}$ describes the state with the initially populated neuron and γ being the impact damping factor typically chosen in the range $\gamma \approx 0.5 - 0.9$. The vector $\mathbf{v_f}$ can be viewed as a Green function propagator acting on the state $\mathbf{v_0}$, whose computation is obtained numerically by a summation of geometrical expansion series as $1 + \gamma G + \gamma^2 G^2 + ...$ which are convergent within approximately first 200 terms at $\gamma \sim 0.7$ (see also [Frahm et al., 2014]). The introduction of the damping factor $\gamma < 1$ is necessary to make the expansion convergent. The vector $\mathbf{v_f}$ is normalized to unity $\sum_i v_f(i) = 1$ and corresponds to the eigenvector of eigenvalue $\lambda = 1$ of the effective google matrix[1] $G_{eff} = \gamma G + (1-\gamma)\mathbf{v_0 e^T}$, in that sense it is the stationary solution of a process driven by the google matrix but reinitiated from time to time, with probability $1-\gamma$, to the initial vector $\mathbf{v_0}$. It represents the nodes that are influenced by the nodes populating $\mathbf{v_0}$ and similarly $\mathbf{v_f^*}$ describes the nodes who influence $\mathbf{v_0}$.

The distributions of probabilities of ImpactRank $P(i) = v_f(i)$, $P^*(i) = v_f^*(i)$ versus the corresponding ImpactRank indexes K, K^* are shown in Fig. 4.5 (right panel) for the initial state neuron $RMGL$. The corresponding top 20 ImpactRank neurons influenced by (and who influence) $RMGL$ are given in columns $IMPR$, $(IMCR)$ of Fig. 4.7. The analysis of neurons linked to $RMGL$ shows that indeed, ImpactRank correctly selects neurons influenced by $RMGL$. The neurons in the top list of $P(i)$ are those pointed by outgoing links of $RMGL$ while those in the top list of $P^*(i)$ are those that have ingoing links towards $RMGL$. Such a method can be easily applied to other initial neuron states of interest showing a contamination propagation over the neural network.

[1] In fact it can also be viewed as a google matrix with a personalized teleportation matrix $\mathbf{v_p e^T}$ instead of $\mathbf{e e^T}$.

4.5 Conclusion

Throughout this chapter we have analysed the structural properties of the *C.elegans* worm neural network, this small system is the only real data of neuron connectivity known for now and understanding its structural properties might help in studying larger neural systems. We have compared the properties of the probability distributions computed from the original network and its inverted configuration and presented a way of using complementary information in order to classify the nodes, here the neurons, in a two dimensional plane and discussed their correlations. Finally we have suggested other possible ranking measures stemming from slight modifications of the Google matrix or a combination of PageRank and CheiRank vectors.

It is clear that several problems arise when we deal with the network of neurons, first of course is the problem of how to define properly the links and relationships between the different cells. Indeed the way we chose is not unique and other considerations are possible and might highlight different topological features. Another crucial issue is the static nature of the Google matrix method, we have only analysed the connectivity structure between the neurons but this does not represent the actual flow of signal and information throughout the neural system as we don't consider the dynamics of the neural network.

It would be interesting to carry this analysis further on larger systems despite the fact that dynamical phenomenons like neuron rewiring for instance are not taken into account by the Google matrix framework as presented here, nevertheless we can bring additional information on the topology of the network by studying the other eigenstates and find out if they help in detecting functional parts in the brain.

Chapter 5

The game of Go from a complex network perspective

5.1 The Ancient Game of Go

This time we will use the complex network approach on a system at the human scale by exploring a very famous and ancient game : the *Game of Go*. This game is played by two opponents on a traditional wooden board, called the *Goban*, where 19×19 intersections are drawn as shown in Fig. 5.1. Both players have stones of the same color, black for one and white for the other, and at each turn they place one of their stones in one of the available intersections among the 361 possible crossings. Once the stones are placed they cannot be moved, the aim is to build territories encompassing the largest area possible and to defeat the opponent by surrounding their stones. A consecutive sequence of stones of the same color is called a *chain* and its possible ways to extend are called *liberties*, an illustration is shown in Fig. 5.1 where the number of free intersections allowing the black chain to grow corresponds to the number of its liberties. When a chain is almost entirely surrounded and has only one liberty left it is in *atari* status and closing the last liberty results in the death of the chain and the removal of the stones from the board. At the end of the game, the territories and the captured stones both give the score and determine the winner.

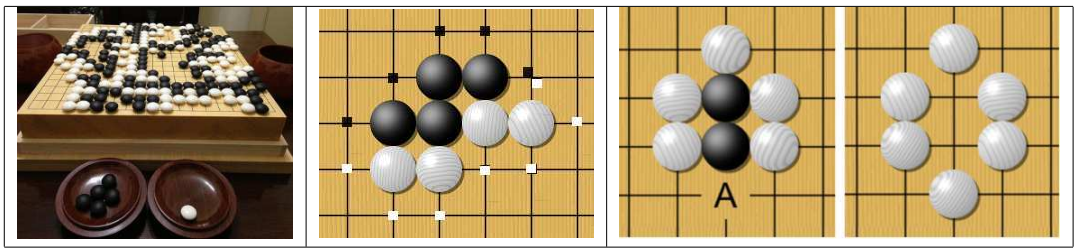

Figure 5.1: *Left :* Traditional Goban with black and white stones. *Middle :* A black chain with its liberties indicated by black dots. Similarly the white liberties are indicated by white dots. *Right :* White player places a stone in A thereby closing the remaining liberty of the black stones which were in atari status, they are then removed from the board. *(Pictures from Wikipedia Commons).*

The exact circumstances of the game's invention are unknown but it originated several thousands of years ago in China where it greatly gained in popularity and quickly became an art alongside the painting, the music and the calligraphy. It later diffused to other Asian countries such as Korea and Japan where it slowly became an important part of the local culture. In Japan, the rules of the game also knew slight modifications and players started to explore the theory of the game through problems and exercises which lead to the development of standard opening moves (*fuseki*) and sequences of moves (*joseki*).

A system of hierarchical classification of players has also been introduced similar to the martial arts where the beginner's level is 30 kyu gradually increasing to 1 kyu and goes above that from 1 dan to 9 dan for the highest possible rank. The players having a dan grade are considered masters with strong skills such that a player of a given level is systematically stronger than the one of a lower level. In order to compensate for this difference players from different levels can fight against each other provided that the strongest opponent has some *handicap* which consists of starting the game with already strategically placed stones of the opposite color. The greater the difference in level the higher number of stones are placed before the game starts.

It is only during the late 1990s that the game knew a wider expansion when people from outside the Asian countries started to reach professional ranking and participate in prestigious tournaments. In general games are an important part of human activities and a better understanding of gaming may provide some insight in the human decision making processes, it is therefore expected that with the growing popularity of the game of go scientists also started to tackle it from a computational perspective with the aim of creating a computer program capable of beating a human player.

Despite the computational power available today there are two major obstacles to reaching this goal. First, the number of allowed position is huge with about 10^{171} configurations (about 10^{50} for chess for example [Tromp and Farnebäck, 2007]) which prevents a systematic exploration of all the possible states of the game. Second, contrary to chess, it is very hard for a computer to decide whether a move is advantageous during the game and for a given situation the comparison between several moves as the best option is also difficult.

The standard approach to this day consists of evaluating a move for a given state of the goban. One can for example use databases of recorded games and expert knowledge to learn and predict the value of a move [Schraudolph et al., 1994]. It turns out that a better way is provided by the recently introduced Monte-Carlo go algorithm which assigns a value to a move from a given state by playing randomly, but according to the rules, until the end of the game. Typically thousands of games are played from that state for a given move and the final value corresponds to the number of times it leads to a winning configuration [Chaslot et al., 2006, Browne et al., 2012].

Crazy Stone [Coulom, 2007b] or MoGo [Wang and Gelly, 2007] figure among the most promising programs currently available, indeed they include many improvements and tricks of Monte-Carlo go algorithm to explore more efficiently and fast the tree of moves possibilities. They incorporate biases towards certain specific moves (for example capture moves) and explore more carefully the most promising moves while at the same time keeping an incentive to explore rarely used moves whose values have a large uncertainty[Gelly and Silver, 2011],[Gelly et al., 2012]. With all these developments, the computer programs are able to beat an average player or a master player with a strong handicap. Further improvement leading to a program capable of beating a highly ranked master player or a professional player without handicap remains a challenge.

Motivation : Although global features, such as chain connections, or the influence of stones over domains of the goban, are crucial in the game of go, local features can be used at many places in the algorithms of computer go, for instance to improve the initialization of the value of each move, or to get a faster estimate of the exact value [Bouzy and Chaslot, 2005, Coulom, 2007a]. There is therefore a clear interest in having a better understanding of local features in the game of go. Moreover the complex network approach has been rarely used to study games and we hope that the new point of view of directed networks will help improving the already existing traditional methods. In [Georgeot and Giraud, 2012], the authors introduced a small network based on local positional patterns and showed that it can be used to extract information on the tactical sequences used in real games. However, the small size of the plaquettes made it difficult to disambiguate many strategically different moves. Here we construct three networks based on positional patterns of different sizes and study their properties, out of them the largest one enables to specify more precise features that were difficult to disambiguate in the previous work.

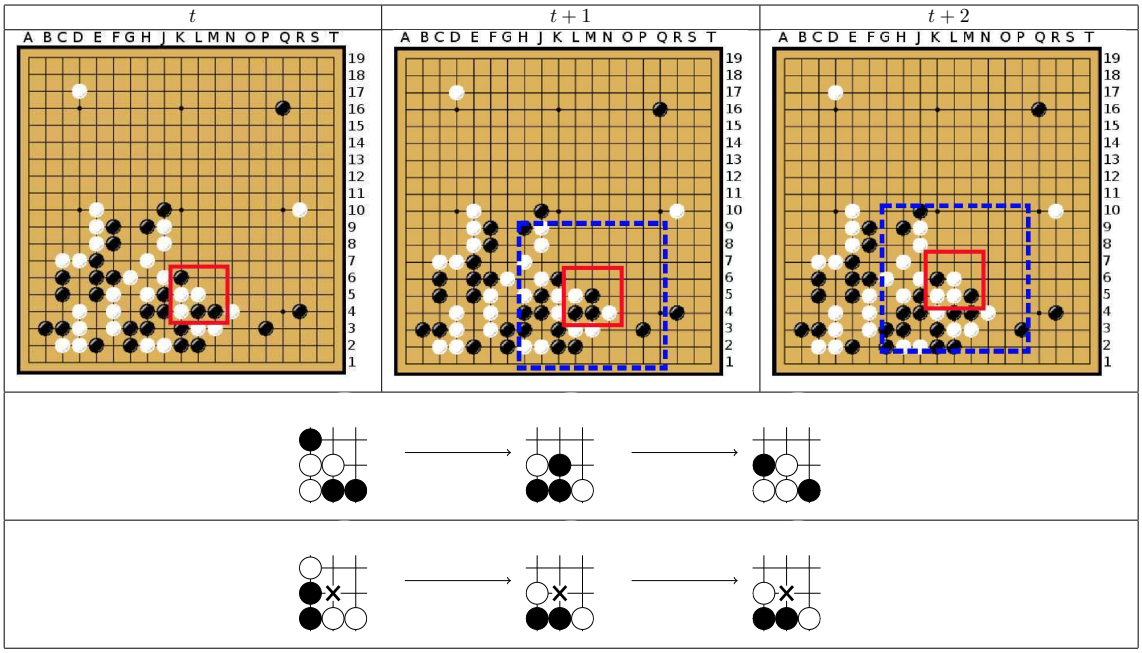

Figure 5.2: Illustrative example of how to build the directed network : At each time step we identify the surroundings of the position where a stone is about to be placed, then the plaquettes are symmetrized and finally connected in chronological order. *Top :* Goban during a game phase : white player has just placed his stone at time t in $(L, 5)$ then black plays at $t+1$ in $(M, 5)$ and finally white responds at $t+2$ in $(L, 6)$. Red square denote a plaquette and the blue square shows the area inside which two moves can be considered related. *Middle :* Raw move patterns extracted in the form of 3×3 square plaquettes. *Bottom :* Symmetrized move patterns as if black is playing in each case. Those are the nodes of the network. *(Pictures from Wikipedia Commons).*

5.2 The Network of Moves

Network definition

Due to the uncommon approach of the game as a complex network it is non trivial to define properly the nodes and the edges in our context. The first natural idea that comes to mind is to consider the nodes as the states of the goban and the links would be naturally the sequence in time bringing the goban from one state to the next one, that is the sequences played in chronological order. This method is completely beyond any computational power and is clearly not feasible. Instead we assume that good players follow general strategies through a series of local tactical fights. We construct the networks representing the game by connecting local moves played in the same neighbourhood (note the similarity with some language networks [Cancho and Solé, 2001] also based on local features) which are described by identifying the empty intersection (h, v) (with $1 \leq h, v \leq 19$) where the new stone is placed.

The vertices are based on what we call "plaquettes", i.e. a part of the goban with a given shape and size which characterizes our network. Each plaquette corresponds to a certain pattern of white and black stones with an empty intersection at its center, on which the player is about to put a stone. An illustrative three time step example is shown in Fig. 5.2 where at time t white player places his stone, then at $t+1$ the black player makes his move and again the white responds at time $t+2$. At each step we identify the environment of the goban around the position where the stone is about to be placed, the shape of the plaquettes in this example is a square (highlighted in red) encompassing 3×3 intersections and the links are simply the edges connecting those moves

from t towards $t+1$. The blue square indicates the area inside which two moves can be considered belonging to the same strategy therefore we do not connect moves further apart as they may simply mean that several unrelated local fights are ongoing in two opposite areas of the goban at the same time.

If both players have the same level it is hard to tell whether an opponent is going to win or lose the game based on a local fight. The idea is that the moves from both players are important but the network is constructed solely thanks to the environment features of a move, therefore every configuration is translated into a configuration where the black player is about to place his stone. This means that we symmetrize all the plaquettes by color swapping, additionally we also identify similar patterns independently of the orientation or the square symmetry.

The three networks that we study here are given by different definitions of the plaquette's shape and specificities :

Network I : made as in [Georgeot and Giraud, 2012] by taking as plaquettes squares of 3×3 intersections, which are subparts of the goban of the form $\{(h+r, v+s), -1 \leq r, s \leq 1\}$ (edges and corners of the board can be accounted for by imagining additional dummy lines outside the board). Once borders and symmetries are taken into account, we obtain as vertices a total of 1107 nonequivalent plaquettes (with empty centers).

Network II : made by also taking squares of 3×3 intersections and identifying plaquettes related by symmetry, but we also include the atari status of the four nearest-neighbour points from the center. Atari status assesses if the chain of stones to which a given stone belongs has only one liberty (one empty intersection connected to it). Removing the last liberty of a chain in atari entails the capture of the whole group. In this case, many seemingly possible configurations are not legal since they would contradict the atari status. This leaves 2051 legal nonequivalent plaquettes with empty centers (the same figure was found in [Huang et al., 2011]).

Network III : based on diamond shaped plaquettes, made of the 3×3 plaquettes discussed above plus the four at distance two from the center in the four directions left, right, top, down. We still identify plaquettes related by symmetry, but do not take into account the atari status. This gives us 193995 nonequivalent plaquettes with empty centers constituting the vertices of the network (96771 are so rare that they are actually never used in our database of games).

To define links of our three networks, we connect vertices corresponding to moves a and b played at (h_a, v_a) and (h_b, v_b) on the board if b follows a in a game of the database and $\max\{|h_b - h_a|, |v_b - v_a|\} \leq d$ where d is some distance. Here contrary to [Georgeot and Giraud, 2012] we put a link only between a an the first move following a in the specified zone. Each integer d corresponds to a different network. It specifies the distance beyond which two moves are considered unrelated. In [Georgeot and Giraud, 2012], different values of d were considered and it was shown that the value $d = 4$ was the most relevant, allowing a correct hierarchization of moves: related local fights are kept while far away tactical moves are not taken into account. In the following we will thus retain this value $d = 4$. Two vertices are connected by a number of directed links given by the number of times the two corresponding moves follow each other in the same neighbourhood of the goban in the game files of the database.

The code in itself was a very big piece of work, one should find a representation for the patterns, generate them, perform symmetries on non trivial shapes (for border and corner patterns for instance), extract information from game sgf files (which are not always properly formatted), play the entire game according to the rules (checking if putting a stone will result in the capture of an enemy's chain), and all these steps should be efficient and as generic as possible in order to allow modifications of pattern shapes (going from square to diamond) or pattern specificities (adding atari status). However a very simplified skeleton of a portion of the code for the network I is shown in the Fig. 5.3 along with an example of an actual game record in .sgf format which is a simple text file.

```
int main(){
  vector<vector<int>> goban(19,vector<int>(19,0)),moves;
  //Declare moves as list of an integer sequences
  generate_central_patterns(moves);//3^8 sequences
  generate_border_patterns(moves); //3^5 sequences
  generate_corner_patterns(moves); //3^3 sequences
  //Generate all the possible sequences of integers
  //representing a move with states coded as
  //0 for empty, 1 for black stone, 2 for white stone.
  remove_colorswap(moves);
  remove_rotation(moves,90); remove_rotation(moves,180);
       remove_rotation(moves,270);
  remove_mirror(moves,1,0); remove_mirror(moves,0,1);
      remove_mirror(moves,1,1);
  //discrad sequences with 1 <-> 2 exchange, rotation
  //symmetry and square mirror image along x,y,diagonal
  //axis : moves now contain 1107 nonequivalent patterns
  vector<string> played_seq; vector<vector<int>>
      node_list;
  vector<int> current_pattern;
  played_seq=init_goban("game.sgf"); //if necessary
  //place handicaps on goban. Extract sequence of moves.
    for(int t=0;t<played_seq.size();t++){
    current_pattern=play(goban,played_seq[t]); //get the
  //surrounding pattern, place the stone. If it closes
  //a last liberty, remove the enemy atari stones.
    index=symmetrize_and_find(current_pattern,moves);
  //find the symmetrized pattern in the list of moves.
    add(node_list,seq[t],index);
  /*record (x,y) positions and ID of moves.*/}
    build_network(node_list,4); // create the matrix in
         j->i format for a network with d=4.
  return 0;}
void build_network(vector<vector<int>> node,int d){
  /* ... code ... */
  for(int s=1;s<node.size();s++){
   for(int p=s-1;p>=0;p--){
if(max(abs(node[s][0]-node[p][0]),
       abs(node[s][1]-node[p][1]))<=d){vertex_j=node[p];
            vertex_i=node[s]; break;} }}
  /* ... more code ... */ }
```

```
(;GM[1]
FF[4]
SZ[19]
PW[xxxstar]
WR[8d]
PB[daikon]
BR[4d]
DT[2009-03-19]
PC[The KGS Go Server at http://www.gokgs.com/]
KM[0.50]
RE[W+Resign]
RU[Japanese]
AB[dd]
[pd]
[dp]
[pp]
CA[UTF-8]
ST[2]
AP[CGoban:3]
TM[1200]
HA[4]
;W[jp];B[jd];W[jj];B[pj];W[cf];B[dj];W[cn];B[en];W[fc];B
    [ee];W[fq];B[el];W[cj];B[ci];W[ck];B[di];W[cp];B[cq
    ];W[do];B[eo];W[dq];B[ep];W[cr];B[eq];W[bq];B[dr];W[
    cc];B[dc];W[db];B[cd];W[cb];B[bc];W[bb];B[bd];W[gd];
    B[fr];W[nq];B[pn];W[nc];B[oc];W[nd];B[pf];W[nf];B[jg
    ];W[ff];B[dg];W[kf];B[jf];W[je];B[ie];W[ke];B[id];W[
    hf];B[if];W[lh];B[kd];W[pg];B[qg];W[qc];B[ob];W[qf];
    B[qe];W[rf];B[re];W[of];B[pe];W[qh];B[rg];W[rh];B[sf
    ];W[ph];B[hg];W[pl];B[lg];W[qq];B[pq];W[pr];B[or];W[
    qr];B[oq];W[nr];B[ns];W[ro];B[np];W[qm];B[lq];W[kg];
    B[kh];W[ki];B[jh];W[mg];B[li];W[mh];B[lj];W[gg];B[hh
    ];W[hj];B[ik];W[jk];B[jl];W[kl];B[kk];W[mp];B[mq];W[
    ij];B[ll];W[km];B[lm];W[kn];B[ln];W[im];B[ko];W[jo];
    B[kp];W[mk];B[lk];W[jq];B[nj];W[qj];B[gb];W[fb];B[gc
    ];W[fe];B[ec];W[eb];B[hd];W[ib];B[hb];W[df];B[ef];W[
    eg];B[eh];W[fg];B[cg];W[bf];B[bg];W[ab];B[af];W[nb];
    B[na];W[ma];B[oa];W[lb];B[ld];W[me];B[cq];W[jb];B[gi
    ];W[dq];B[ed];W[fd];B[cq];W[br];B[cm];W[bm];B[dn];W[
    co];B[cl];W[bl];B[ho];W[hn];B[gn];W[hk];B[jn];W[jm])
```

Figure 5.3: *Left :* Very simplified scheme portion of the code for building the network I from one single game file. *Right :* An example of an sgf game file where WR and BR indicate the rank of white and black players (here 8 dan and 4 dan respectively), HA indicates the presence of handicaps (here 4 stones in favor of black), AB give the coordinates where the handicap stones should be placed and the bottom series of strings is the course of the game. Each coordinates where a player puts a stone is given by letters, the first and second letters correspond to capital letters and numbers as shown in Fig. 5.2.

Network statistics

We have identified the occurrence of these different plaquettes in games from a database available at [U-go, 2013] where each game is entirely registered in .sgf format using specific sequences of strings to denote moves from both players. This database contains the sequence of moves of 135663 different games corresponding to players of diverse levels. The games recorded have been played online, and the dan rankings have been mutually assessed according to the results of these plays. The frequency of the different plaquettes is shown in Fig. 5.4. It can be compared to Zipf's law, an empirical law seen in many natural distributions (word frequency, city sizes, chess openings...) [Henmon and Zipf, 1936, Gabaix, 1999, Okuyama et al., 1999, Blasius and Tönjes, 2009]. For items ranked according to their frequency, it corresponds to a power-law decay of the frequency versus the rank. The data presented in Fig. 5.4 show that the three different network choices all give rise to a distribution following Zipf's law, although the slope varies from ≈ -1 (networks I and II) to a slightly slower decay for the largest network (network III).

We display in Fig. 5.5 the top 30 moves in order of decreasing frequency of occurrences for network III. The most common moves correspond to few stones on the plaquettes, which is natural since these ones are present at the beginning of almost all local fights, while the subsequent moves differ from games to games.

The total number of links including multiplicity is 26116006 links. The numbers without link multiplicity are respectively 558190 (network I), 852578 (network II) and 7405395 (network III).

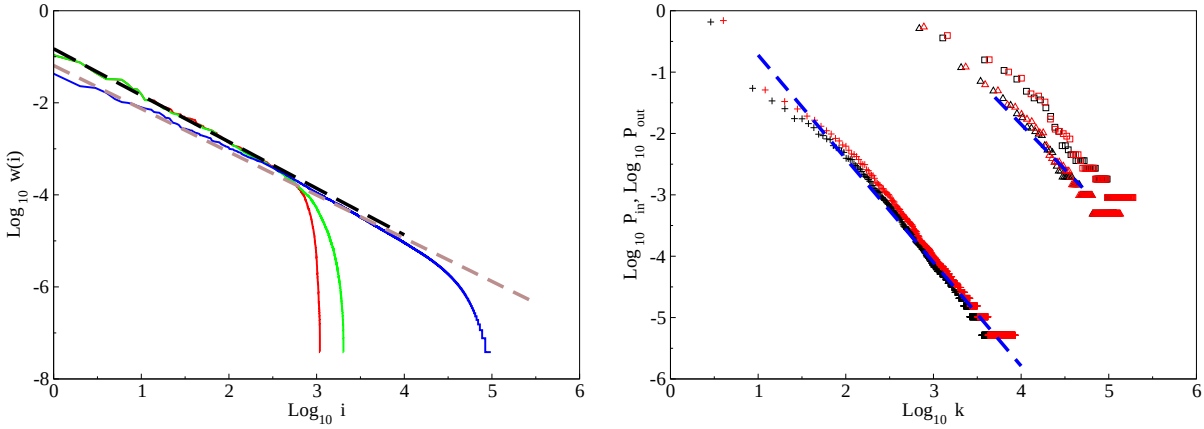

Figure 5.4: *Left panel :* Distribution of frequency of occurrences $w(i)$ of different plaquettes for the three different networks (full lines), from left to right at the bottom: red: square plaquettes (network I), green: square plaquettes with atari status (network II), blue: diamond plaquettes (network III) (data from networks I and II are indistinguishable over parts of the curves). The dashed straight lines are power law fits with slopes -1.02 (black upper line, fit of network II) and -0.94 (brown lower line, fit of network III). *Right panel :* Distribution of incoming links $P_{\rm in}$ (black) and outgoing links $P_{\rm out}$ (red/grey) for the three different networks; square plaquettes (network I) (squares), square plaquettes with atari (network II) (triangles), diamond plaquettes (network III) (crosses). The dashed lines are power law fits with slopes -1.47 (right) and -1.69 (left).

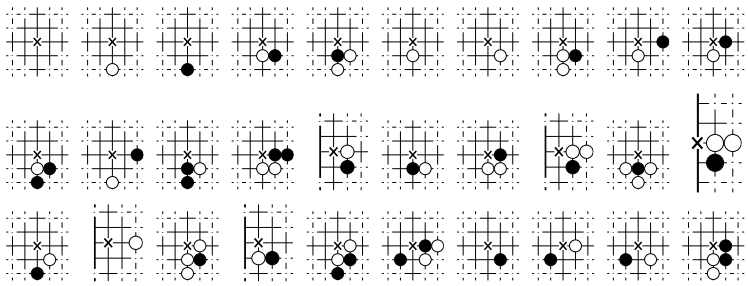

Figure 5.5: Top 30 plaquettes in frequency of occurrences for the network III (diamond plaquettes). Black plays at the black cross. Dotted intersections are outside the diamond plaquette and their status is unknown.

The link distributions are shown in Fig. 5.4: it is close to a power-law, implying that the networks present the scale-free property. We can notice a symmetry between ingoing and outgoing links, which is a peculiarity of this problem, not seen in the World Wide web for instance, where the exponent for $P_{\rm out}$ (≈ -2.7) is different from the one for $P_{\rm in}$ (≈ -2.1) [Donato et al., 2004]. Here exponents are similar and close to 1.5, intermediate between these two values. Our results indicate the presence of a symmetry (at least at a statistical level) between moves that follow many different others and moves which have many possible followers. This symmetry is natural, since in many cases (i.e. in the course of a local fight) the occurrence of a plaquette in the database implies the presence of both an ingoing and an outgoing link.

5.3 Spectrum and Ranking vectors

We have presented up to now the construction of our networks for the game of go, and their global statistical properties. To get more insight into the organization of the game, we will compute the PageRank and CheiRank vectors from G and G^* following the usual procedure. The stochastic connectivity matrix was obtained from the definition of the network as explained above and we performed the computations at $\alpha = 1$.

In Fig. 5.6 the distributions of PageRank and CheiRank are plotted for the three networks showing that ranking vectors follow an algebraic law with a slightly different exponent for the largest network. Similarly as for the link distribution, we notice a symmetry between distributions of ranking vectors based on ingoing links and outgoing links, again an original feature which can be related to the statistical symmetry between ingoing and outgoing links.

In order to check to what extent this symmetry affects the ranking vectors, we also plot in Fig. 5.6 the CheiRank K^* as a function of the PageRank K. It indeed shows that the two quantities are not independent and strong correlations between PageRank and CheiRank do exist. This symmetry is not visible in general for other networks (see e. g. [Ermann and Shepelyansky, 2011] where similar plots are shown in the context of world trade, displaying much less correlation). Nevertheless, the symmetry is clearly not exact, especially for the largest network (a perfect correlation will produce points only on the diagonal), in fact the plots are not even symmetric with respect to the diagonal. Thus PageRank and CheiRank produce genuinely different information on the network.

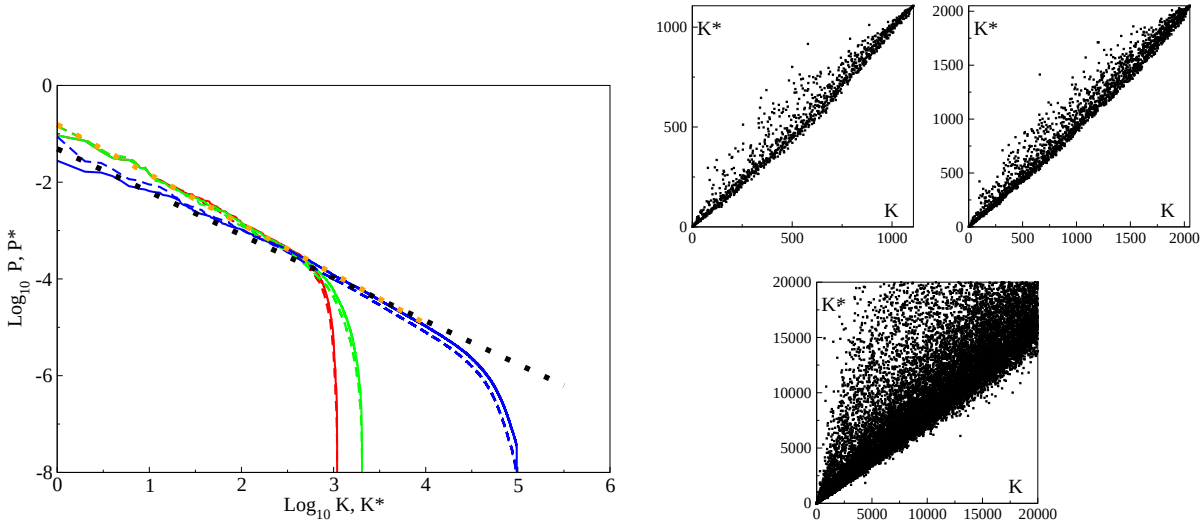

Figure 5.6: *Left panel :* Distribution of ranking vectors (normalized by $\sum_K P(K) = \sum_{K^*} P^*(K^*) = 1$) for the three different networks: PageRank $P(K)$ (solid lines) and CheiRank $P^*(K^*)$ (dashed lines), same color code for the networks as in Fig. 5.4 (data from networks I and II are indistinguishable over parts of the curves). The dotted lines are power law fits with slopes -1.03 (orange upper line, fit of network II) and -0.89 (black lower line, fit of network III). *Right panel :* PageRank-CheiRank correlation plot of the three different networks : square plaquettes (network I)(top left), square plaquettes with atari status (network II)(top right) and diamond plaquettes (network III)(bottom). PageRank K is given in x-axis and CheiRank K^* in y-axis, the plot of network III is a zoom on the top 20000 moves in both K and K^*.

Fig. 5.7 right panel shows the first 30 plaquettes in decreasing importance in the PageRank and CheiRank vectors. The correlation between the two sequences is clearly visible, although it is again not perfect. We note that these sequences are also very similar to the one obtained by just counting the move frequency (as in Zipf's law): most frequent moves tend to dominate the ranking vectors.

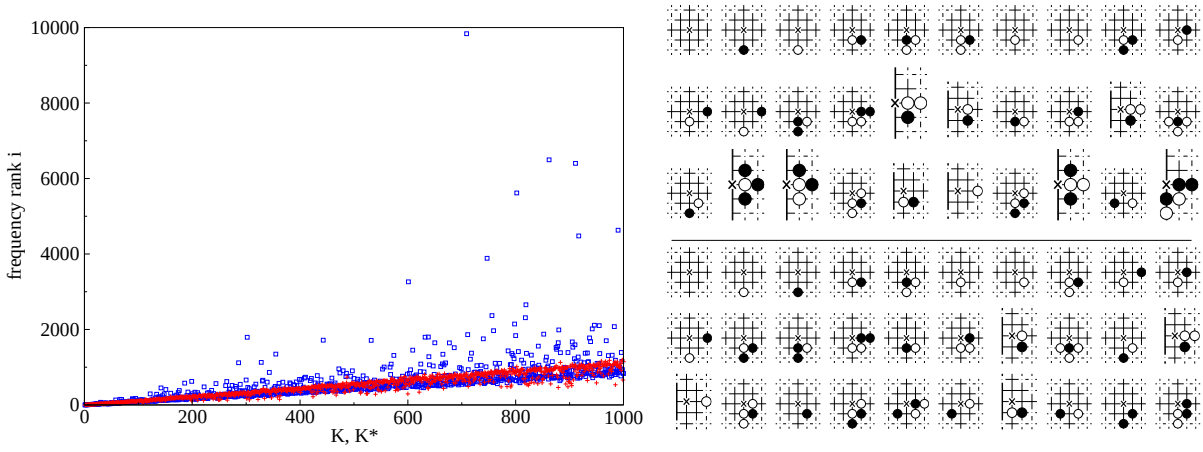

Figure 5.7: *Left panel :* Correlation plot of PageRank-CheiRank vs frequency of moves for network III (diamond plaquettes) (only first 1000 moves in K are shown); blue squares: PageRank K, red crosses: CheiRank K^*. *Right panel :* Top 30 plaquettes for first eigenvector of G (PageRank)(top) and G^* (CheiRank)(bottom) of the network III.

However, as Fig. 5.7 left panel shows, the correlation between ranking vectors and frequency ordering is far from perfect, especially for the PageRank which can be extremely different from the rank obtained by frequency. This shows that the ranking vectors present an information obtained from the network construction, which differs from the mere frequency count of moves in the database. Indeed, as explained above the frequency count is related to the link distribution due to the construction process of the network. It is known in general that the PageRank has some relation with the distribution of ingoing links, but with the significant difference that it highlights nodes whose ingoing links come from (recursively defined) other important nodes. In our case this means that highlighted moves correspond to plaquettes with ingoing links coming from other important plaquettes. Thus the PageRank underlines moves to which converge many well-trodden paths of history in the different games of the database. The CheiRank does the same in the reverse direction, highlighting moves which open many such paths.

The ranking vectors discussed above are just one eigenvector of the matrices associated with a given network. However, other eigenvalues and their associated eigenvectors also contain information about the network. We have computed the spectrum of the Google matrix for the three networks which are shown in Fig. 5.8. For square plaquettes (network I) and square plaquettes plus atari status (network II) all eigenvalues are computed. In the case of the largest network, standard diagonalization techniques could not be used and therefore we applied an Arnoldi-type algorithm to compute the largest few thousands eigenvalues in the complex plane. For the G matrix of the diamond network (network III), about 1000 eigenvectors were computed. For G^* matrix of diamond, about 500 eigenvectors were computed.

Arnoldi method

The Arnoldi method is an algorithm proposed in 1951 [Arnoldi, 1951] which is useful to compute eigenvalues of large sparse asymmetric matrices when the complete diagonalization is not possible for computational reasons. This method is based on the subspace spanned by 0 to $n-1$th powers of the matrix multiplied by an initial vector. The resulting vectors can be transformed into an orthogonal basis of this subspace thereby providing a good approximation of n eigenvectors corresponding to the n largest (in modulus) eigenvalues of our considered matrix. In our case the size of the network III is off limits for a direct diagonalization so that we used the standardised code added for LAPACK, known as ARPACK, to compute a few hundreds of eigenvectors corresponding to the largest eigenvalues.

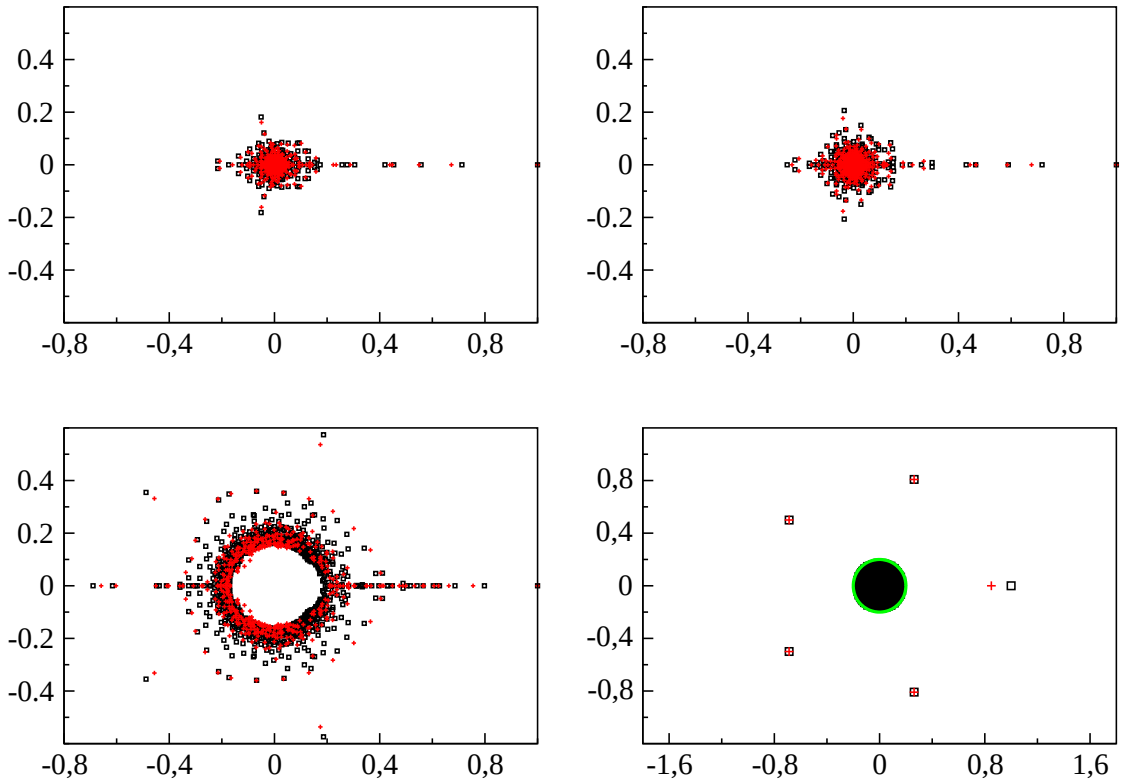

Figure 5.8: Spectrum in the complex plane of G (black squares) and G^* (red/grey crosses) for the three different networks : I (top left), II (top right) and III (bottom). *Bottom right :* Spectrum of G_M at $\alpha = 0.85$ generated with $N = 2000$, $L = 50000$ and $r = 5$ (black squares). The average number of non zero element per column is $Q = L/N = 25$ and the green circle has a radius of $R = 1/\sqrt{25} = 0.2$. The red crosses are the solutions of $\lambda^5 - 0.85^5 = 0$.

For networks I and II we observe a huge gap between the first and the other eigenvalues. For the third network, there is still a gap between the first eigenvalue and next ones, but it is smaller. While the distribution of the ranking vectors shown in Fig. 5.6 reflects the distribution of links, the gap in the spectrum is related to the connectivity of the network and the presence of large isolated communities [Georgeot et al., 2010].

For the network III there are also some eigenvalues of large modulus placed at regular angles $2\pi/5$ indicating the presence of some cycles of order 5 in the network of moves. To understand the meaning of those eigenvalues we can use a very simple model, construct the google matrix G_M and derive an approximation to those eigenvalues. Let's consider a system of N nodes linked as a directed chain of length $L >> N$ and suppose that the set of $S = \{N_1, ..., N_N\}$ nodes is partitioned into r disjoint subsets S_r such that $\cup_r S_r = S$. Let's assume now that each subset points to only one other subset, the chain is constructed by randomly linking nodes picked inside a subset with probability $p = r/N$ thereby creating a path between group of nodes and generating the cyclic eigenvalues. Because of the partitioning the stochastic matrix S_M has a block format and the approximation consists of considering a matrix M of size $r \times r$ equivalent to S_M where the blocks are replaced by 1. In that case the determinant is given by :

$$det(\alpha M - Id\lambda) = \lambda^r - \alpha^r = (\lambda - \alpha) \sum_{q=0}^{r-1} \alpha^{r-1-q} \lambda^q \tag{5.1}$$

The cycle eigenvalues are approximated by the zeros of the power series. An example with $r = 5$ is shown in the bottom right panel of Fig. 5.8 where the solutions of the polynomial (red) matches the $2\pi/5$ eigenvalue cycle of G_M visible in black. Thus the eigenvalues placed at regular

angles are produced by connections between several group of nodes inside which the nodes are only weakly connected between themselves. One can improve the node partition model and play around to generate more complicated eigenvalue spectrum for the google matrix G_M.

The presence of a large gap indicates a large connectivity, which is reasonable for the smaller networks. The presence of a smaller gap for network III indicates that there is more structure in the networks with larger plaquettes which disambiguate the different game paths and makes the communities of moves more visible. However, the gap being still present shows that even at the level of diamond-shaped plaquettes, the moves can belong to many different communities: this underlines one of the specificities of the game of go, which makes a given position part of many different strategic processes, and makes it so difficult to simulate by a computer.

5.4 Eigenvectors and Communities

Until now we have mainly concentrated and discussed in details the spectrum and the dominant eigenvector in all our work, here we will focus on the other eigenvectors as they carry important informations that can be significantly different from the ranking vectors and we will illustrate our analysis on a few arbitrarily chosen such eigenvectors.

In Fig. 5.9 we display the intensities of the first 200 eigenvectors of the three different networks. It is clear that these eigenvectors have specific features, not being spread out uniformly or localized around a single specific location. Correlations are also clearly visible between different eigenvectors, materialized by the vertical lines where several eigenvectors have similar intensities on the same node. Correlations are less visible on the largest network, but it is also due to the much larger size of the vectors which decreases the individual projections on each node. It is interesting to note that these correlations are not necessarily related to the PageRank values or the frequency of moves: vertical lines tend to be more visible on the left of the figure corresponding to high PageRank, but they are present all over the interval meaning that certain sequences of eigenvectors have correlated peaks at locations with relatively low PageRank.

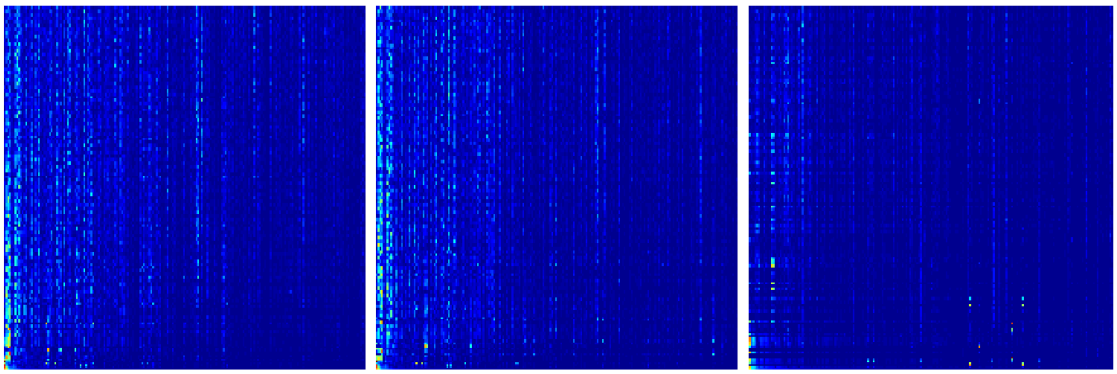

Figure 5.9: Eigenvector correlation map of the matrix G for the three different networks : I (left), II (middle) and III (right). Top 200 eigenvectors in order of decreasing eigenvalue modulus are plotted horizontally from bottom to top. Only the first 200 components are shown in the PageRank basis. The colors are proportional to the modulus of components (the normalization of an eigenstate ψ is $\sum_i |\psi_i|^2 = 1$), from blue/dark grey (minimal) to red/light grey (maximal).

In order to quantify these effects, we first look at the spreading of the eigenvectors. For a given vector, how many sites have significant projections ? This can be measured for a vector ψ through the Inverse Participation Ratio (IPR) as discussed in chapter 4 : $\sum_i |\psi_i|^4 / (\sum_i |\psi_i|^2)^2$. The data of Fig. 5.10 for the eigenvectors corresponding to the largest eigenvalues show that these vectors are not random or uniformly spread. On the contrary their IPR is quite small even for the largest network: in this case only a few dozen sites contribute to a given eigenvector, among almost 200000 possible nodes. We also find that there is a relatively large dispersion of the IPR around the mean

value. Qualitatively the features are quite similar for both G and G^* distributions but there is both a lower mean value and a lower dispersion for G^*, indicating that the statistical symmetry found previously between incoming and outgoing links is indeed only approximate.

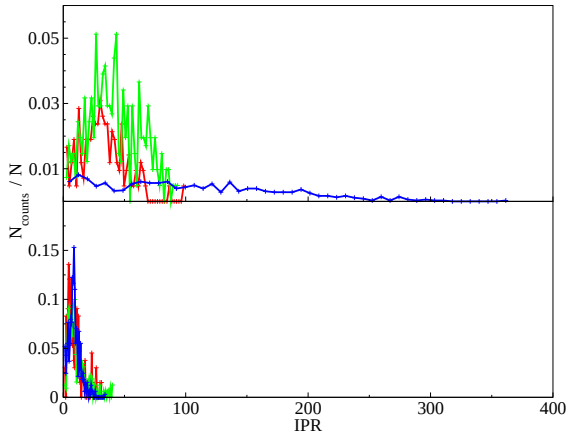

Figure 5.10: Histogram of IPR values for Network I (red/dark grey), Network II (green/light grey) and Network III (blue/black). Top panel shows the values computed for eigenvectors of G and bottom panel shows the same for G^*. Data correspond to the top 221 eigenvalues (network I), top 410 eigenvalues (network II) and top 999 eigenvalues (network III).

What is the meaning of these eigenvectors ? If we interpret the Google matrix as describing a random walk among the nodes of the network as in the original paper [Page et al., 1999], eigenvectors of G correspond to parts of the network where the random surfer gets stopped for some time before going elsewhere in the network. In other words, they are localized on sets of moves which are more linked together than with the rest of the network. This corresponds to so-called communities of nodes which share certain common properties. In social network, the importance of communities has been stressed several times and they are the subject of a large number of studies (see e.g. the review [Fortunato, 2010]). The use of the eigenvectors of G to extract the communities is one of the many available methods, which has been used already in the different context of the World Wide Web [Ermann et al., 2013]. As already mentioned, eigenvectors with largest eigenvalues tend to be localized on groups of nodes where the probability is trapped for some time. This approach will thus detect communities of nodes from where it is difficult to escape, i.e. with few links leading to the outside. In parallel, the eigenvectors of G^* tend to be localized on groups of nodes with few incoming links from the outside. Fig. 5.10 shows that this latter type of community, obtained from G^*, tends to be smaller on average for the go game than the former type, obtained from G. These different communities should reflect different strategic groupings of moves during the course of the game.

The concept of community being intrinsically ambiguous, one can assign a subjective meaning to the definition of the community related to a chosen method. In our case, it is a difficult task to establish clear characteristics regarding what moves should be considered belonging to which community, however in the spirit of "moves that are more played together" or "similar moves" we can observe that a single eigenvector may contain a mixing of several communities. This could explain why in Fig. 5.9 one can see similar patterns appearing in different eigenvectors. These considerations are confirmed by the figures Fig. 5.11 to Fig. 5.14 where the first 30 moves of representative eigenvectors of G and G^* are displayed, ranked by decreasing component modulus. While some common features appear, one gets the impression that groups of moves corresponding to different strategic processes are mixed and should be disentangled : for instance the fourth example in the left part of Fig. 5.11 seems to mix moves where black captures a white stone and moves where black connects a chain.

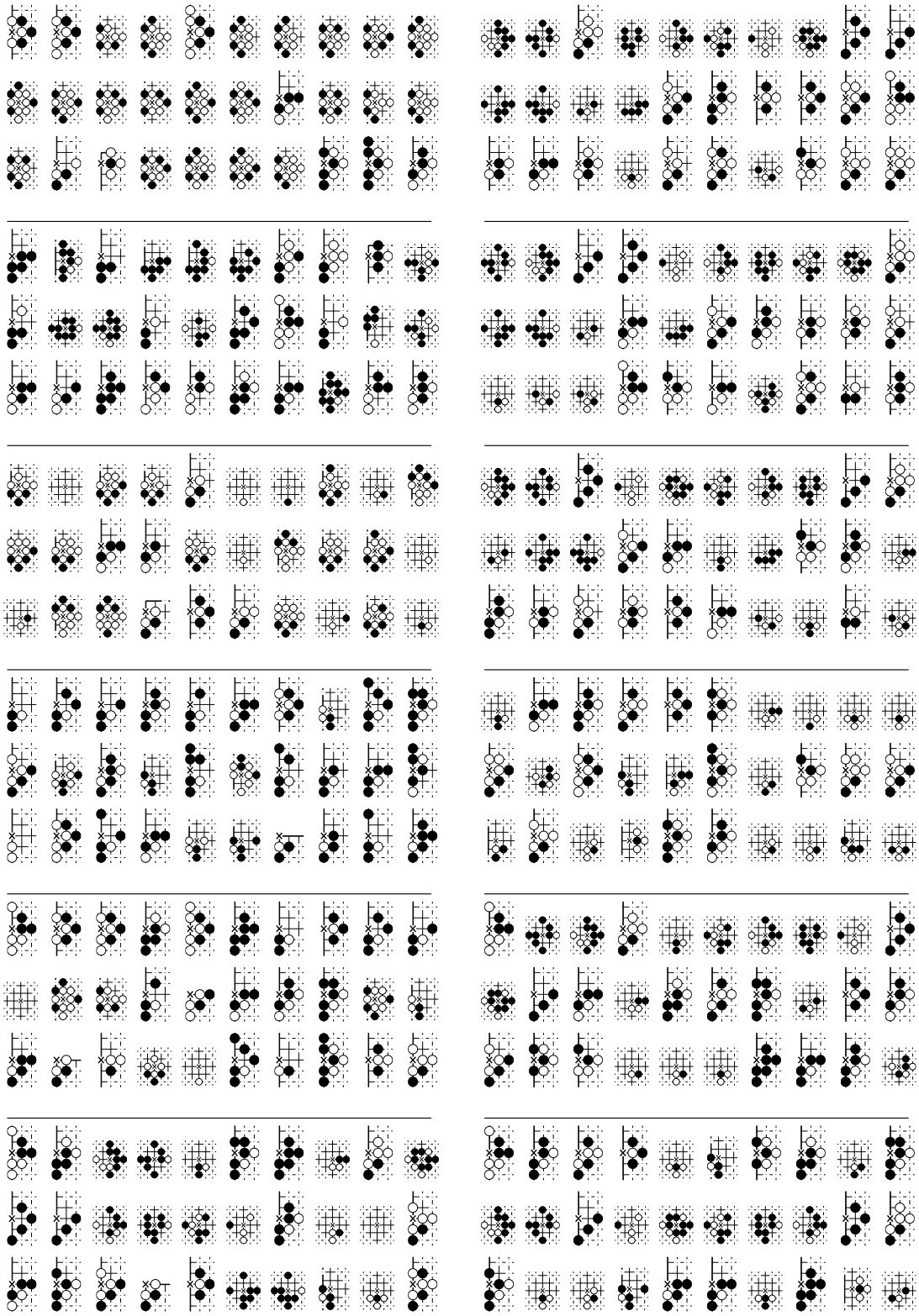

Figure 5.11: Examples of the top 30 nodes where eigenvectors of G localize themselves for diamond network. *From top to bottom left and top to bottom right :* $\lambda_7 = -0.6158$, $\lambda_{11} = 0.1865 - 0.5739i$, $\lambda_{13} = 0.5651$, $\lambda_{21} = -0.4380$, $\lambda_{22} = 0.4294 + 0.0006481i$, $\lambda_{32} = 0.3847 + 0.04677i$, $\lambda_{34} = 0.3412 + 0.1430i$, $\lambda_{37} = -0.06799 + 0.3593i$, $\lambda_{51} = -0.2929 + 0.1753i$, $\lambda_{128} = -0.2673$, $\lambda_{88} = -0.284 + 0.0732i$ and $\lambda_{95} = -0.2188 + 0.1804i$.

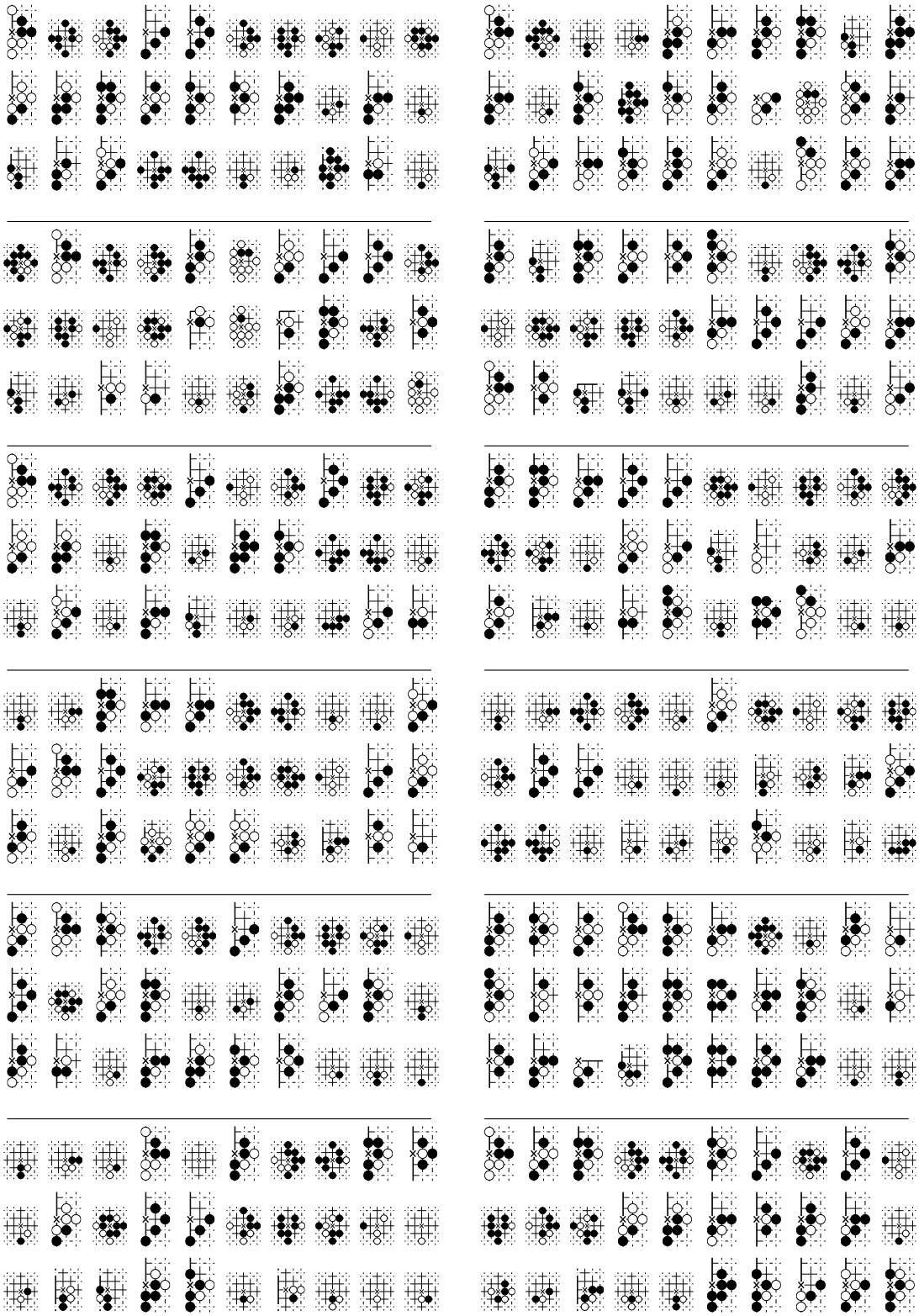

Figure 5.12: More examples of the top 30 nodes where eigenvectors of G localize themselves for diamond network. *From top to bottom left and top to bottom right :* $\lambda_{84} = -0.03434 + 0.2949i$, $\lambda_{74} = -0.1613 + 0.2660i$, $\lambda_{63} = 0.1135 + 0.3005i$, $\lambda_{103} = 0.2819$, $\lambda_{118} = 0.05446 + 0.2717i$, $\lambda_{48} = 0.3439$, $\lambda_{78} = -0.3037$, $\lambda_{135} = 0.2297 + 0.1242i$,$\lambda_{156} = -0.1196 + 0.2195i$,$\lambda_{45} = -0.3515$,$\lambda_{211} = -0.05076 + 0.2223i$ and $\lambda_{99} = -0.1924 + 0.2069i$.

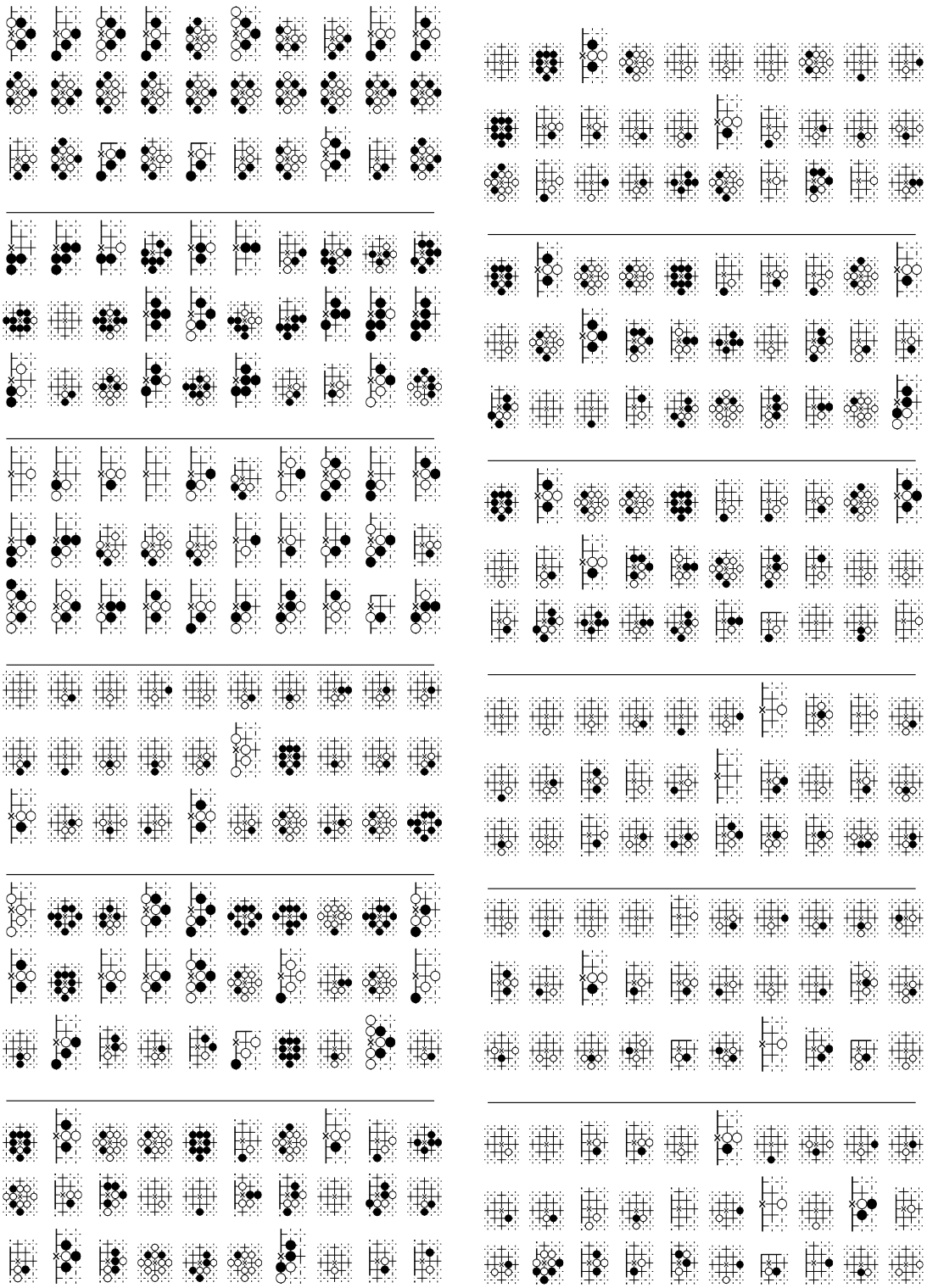

Figure 5.13: Examples of the top 30 nodes where eigenvectors of G^* localize themselves for diamond network. *From top to bottom left and top to bottom right :* $\lambda_7 = -0.6023$, $\lambda_{11} = 0.1743 - 0.5365i$, $\lambda_{18} = -0.4511$, $\lambda_{21} = -0.4021$, $\lambda_{22} = 0.4145$, $\lambda_{32} = 0.3646 - 0.1359i$, $\lambda_{34} = 0.3018 - 0.2175i$, $\lambda_{37} = -0.0683 + 0.3599i$, $\lambda_{51} = 0.3515$, $\lambda_{128} = -0.1938 - 0.0234i$, $\lambda_{88} = -0.1770 - 0.1498i$ and $\lambda_{95} = -0.2124 + 0.0761i$.

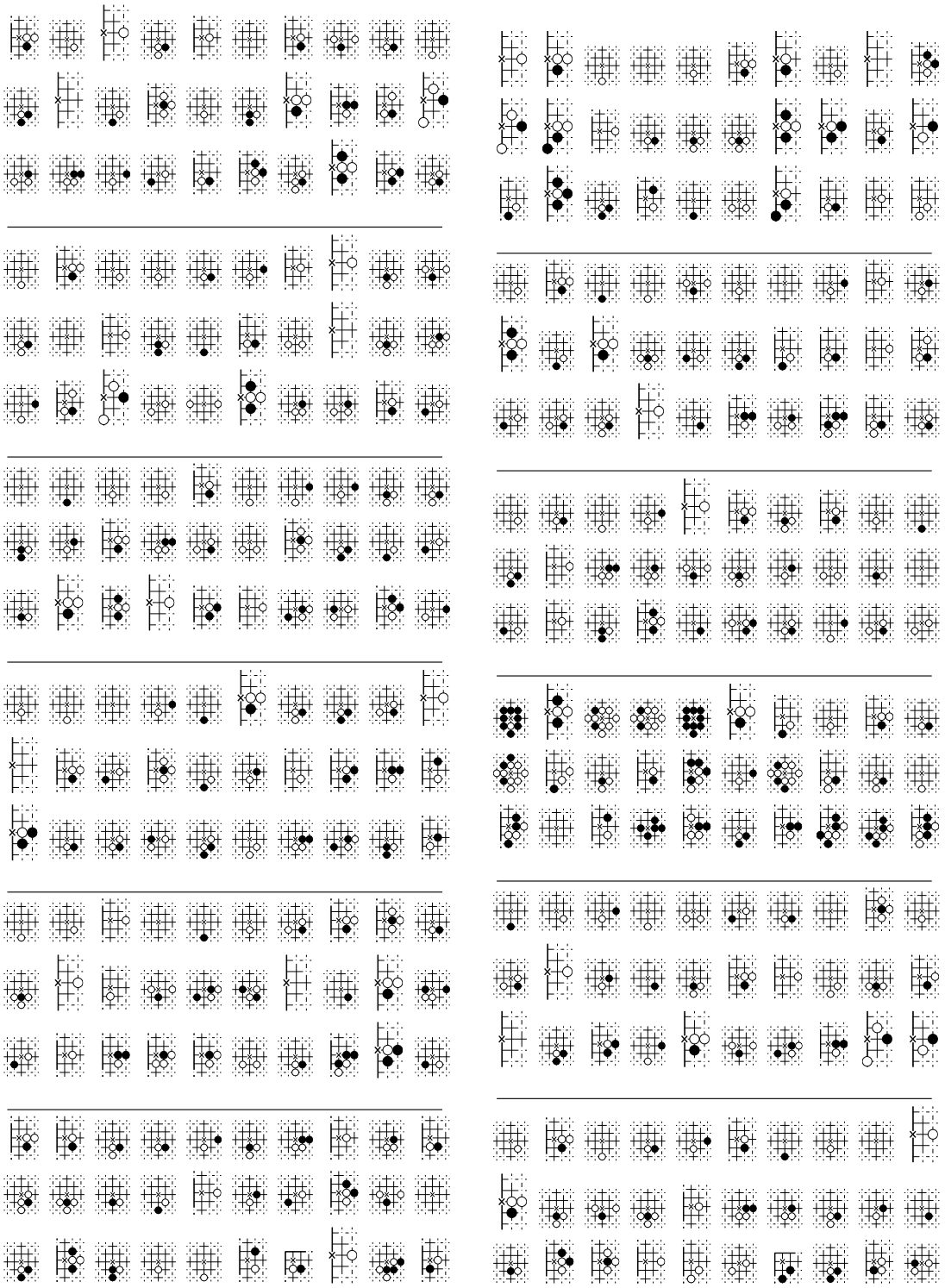

Figure 5.14: More examples of the top 30 nodes where eigenvectors of G^* localize themselves for diamond network. *From top to bottom left and top to bottom right :* $\lambda_{84} = -0.1194 - 0.2064i$, $\lambda_{74} = -0.2546 - 0.0146i$, $\lambda_{63} = 0.2770 - 0.0684i$, $\lambda_{103} = -0.2116 + 0.0607i$, $\lambda_{118} = -0.1617 - 0.1308i$, $\lambda_{48} = -0.3002 - 0.1724i$, $\lambda_{78} = -0.1884 - 0.1466i$, $\lambda_{135} = 0.2629 - 0.0433i$, $\lambda_{156} = 0.1988 - 0.1276i$, $\lambda_{45} = 0.1309 + 0.3310i$, $\lambda_{211} = -0.1462 + 0.1205i$ and $\lambda_{99} = -0.0858 + 0.2137$.

Community extraction methods

In principle one could use correlations as the ones shown in Fig. 5.9 directly to identify communities, but we chose a different strategy. We propose here different basic methods that can be a first step into separating the communities within a given eigenvector. The simplest and most straightforward method consists in filtering out the effects of the most common and important moves by removing the top moves given by PageRank and CheiRank vectors. Examples are shown in figures Fig. 5.15 to Fig. 5.18 where the remaining moves in the given eigenvectors of figures Fig. 5.11 to Fig. 5.14 correspond to a specific set of moves. Very common moves (such as empty or almost empty plaquettes) have been deleted, leaving more focused groups of moves. For example, the third eigenvector in left part of Fig. 5.15 is much more focused on various moves containing situations of *Ko* or of imminent capture (*Ko* or "eternity" is a famous type of fights with alternate captures of opponent's stones).

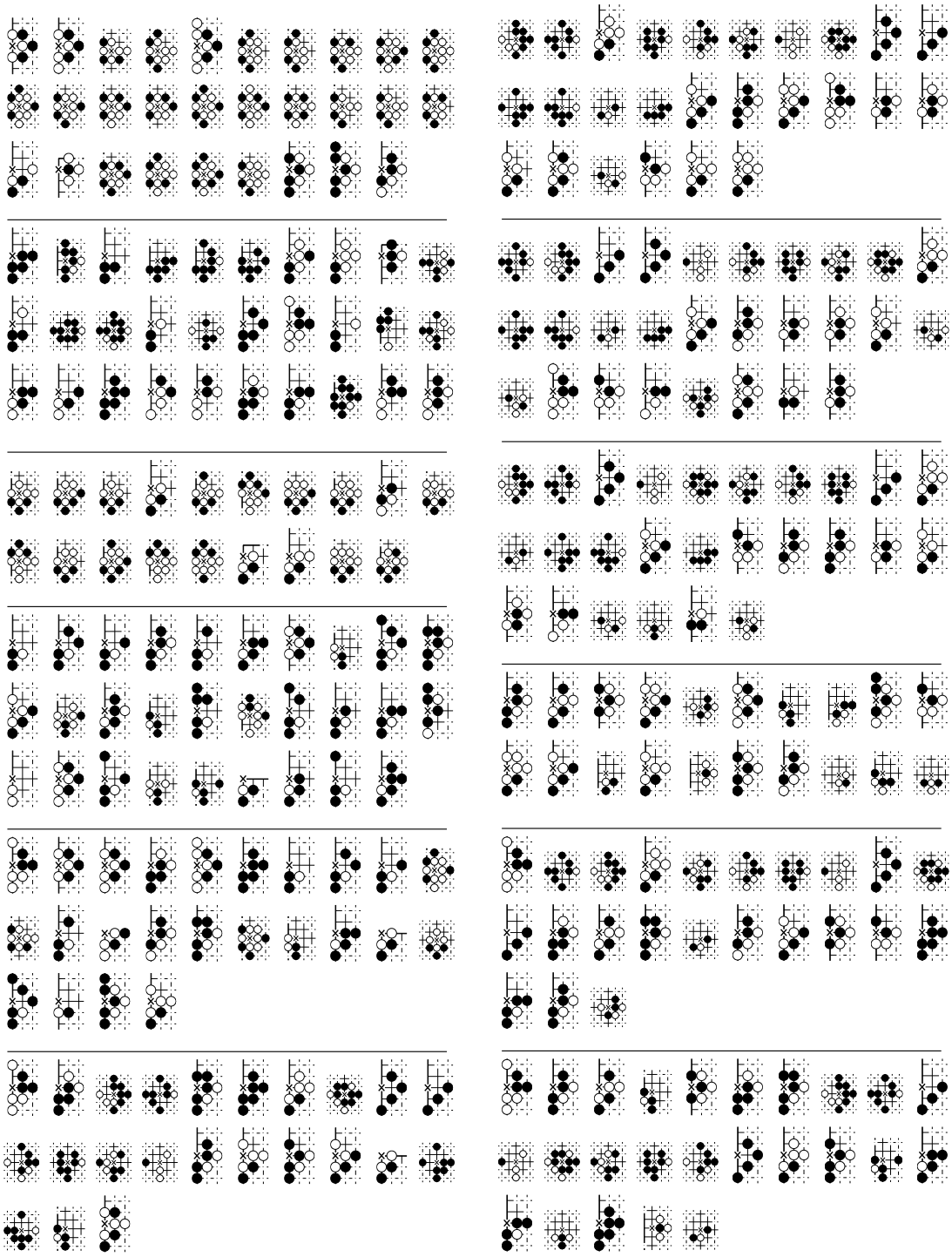

Figure 5.15: Same eigenvectors as in Fig. 5.11 treated by filtering out the top 30 PageRank moves.

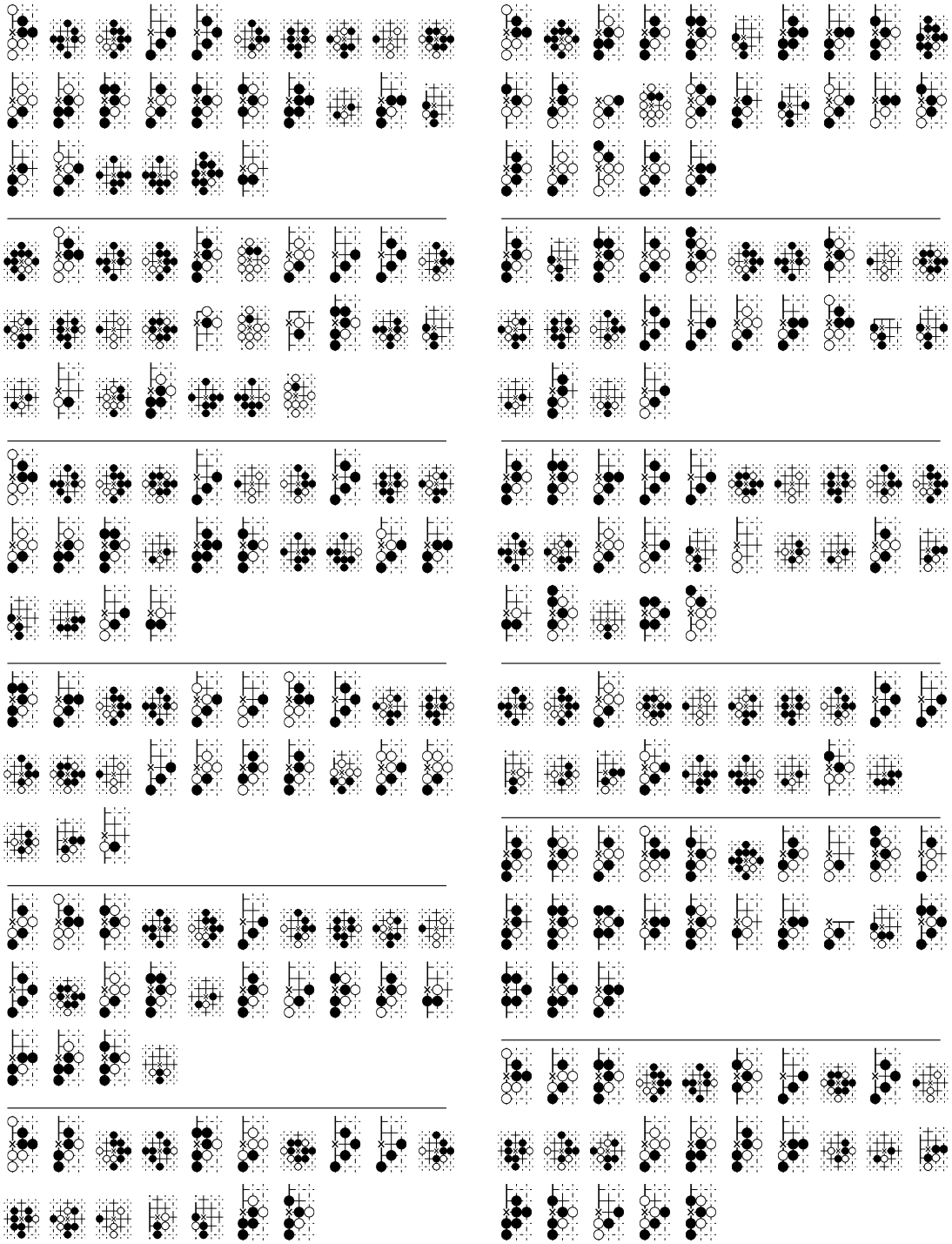

Figure 5.16: Same eigenvectors as in Fig. 5.12 treated by filtering out the top 30 PageRank moves.

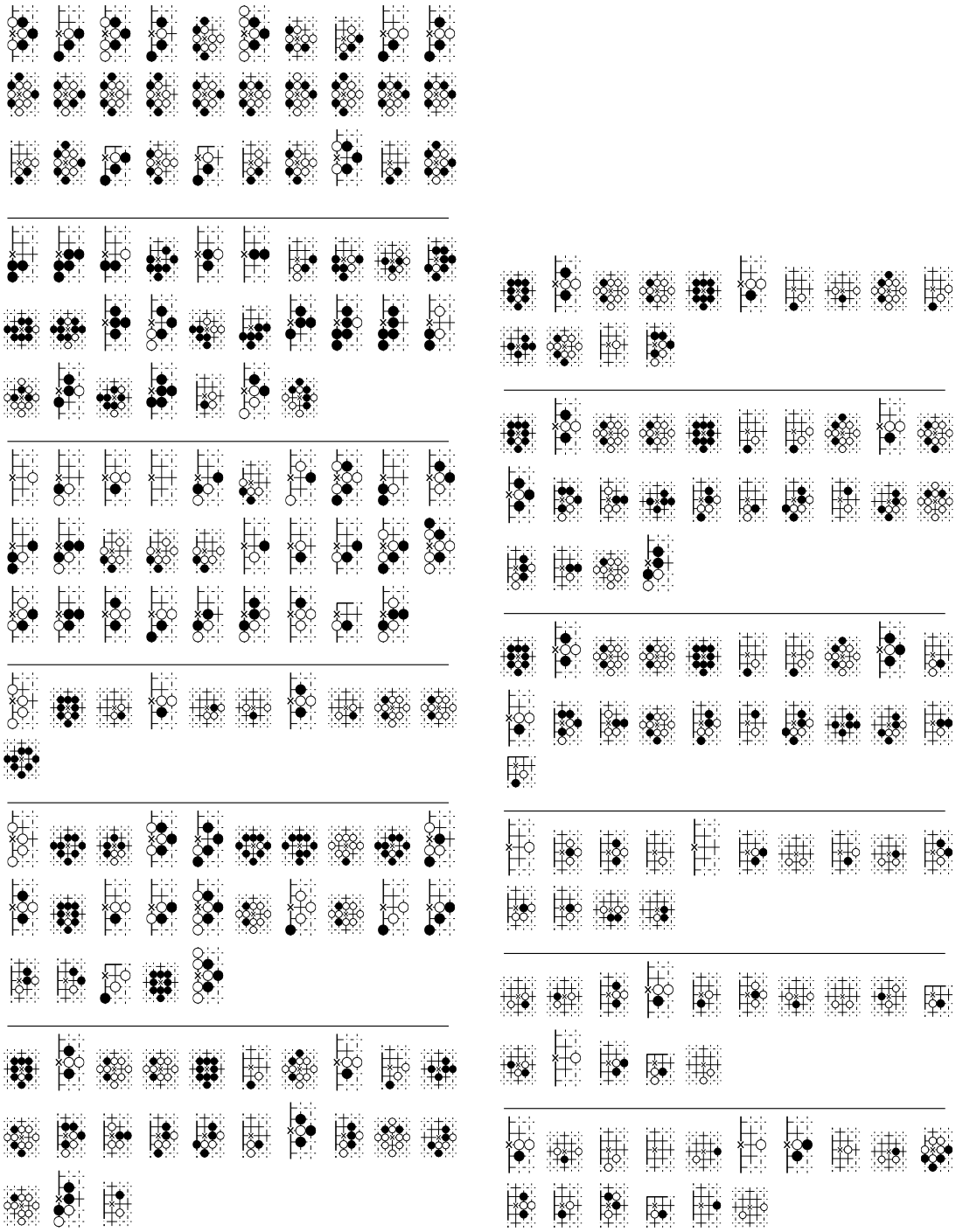

Figure 5.17: Same eigenvectors as in Fig. 5.13 treated by filtering out the top 30 CheiRank moves.

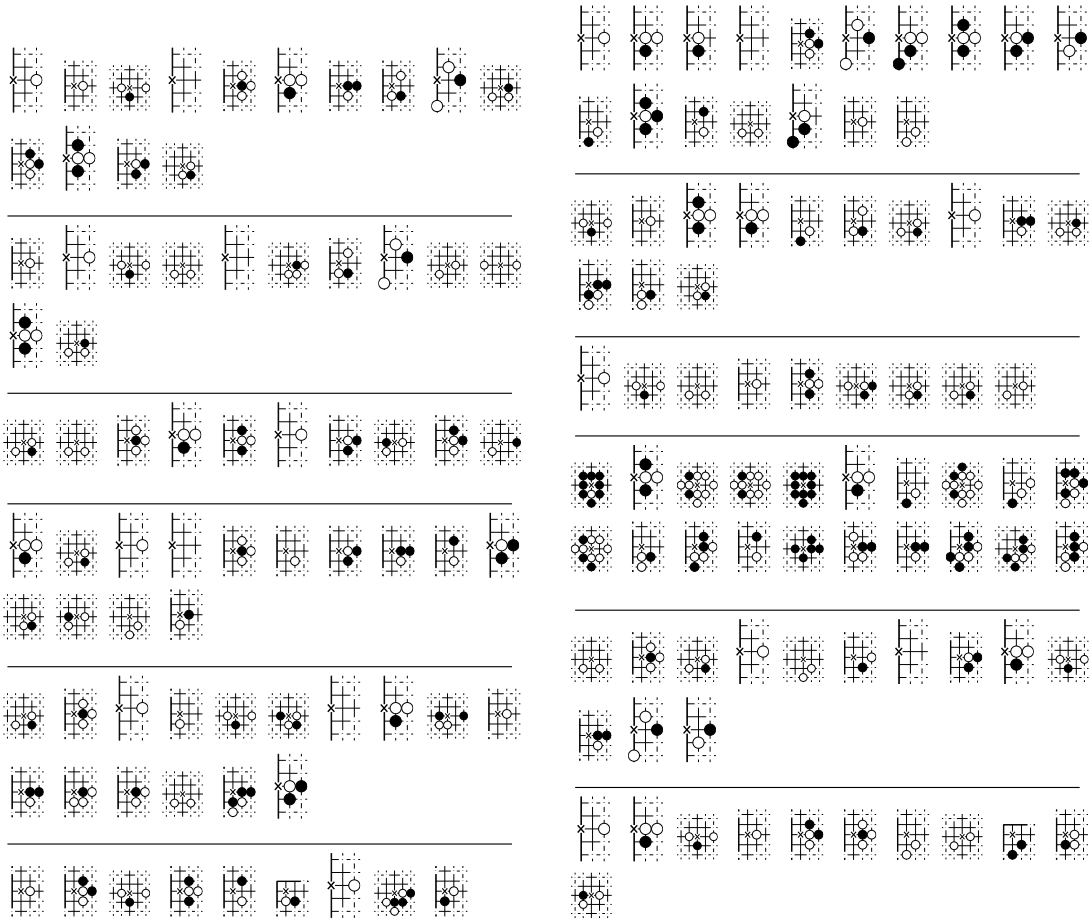

Figure 5.18: Same eigenvectors as in Fig. 5.14 treated by filtering out the top 30 CheiRank moves.

A more systematic method that we propose is to consider the ancestors of each move and determine if they share a significant number of preceding moves. As the Google matrix describes a Markovian transition model it would be natural to look for incoming flows of two moves to decide whether they belong to the same community. We implement it as follows: We choose two moves m_1 and m_2, with respectively N_1 and N_2 incoming links. We denote the origin of these incoming links pointing to m_1 and m_2 as sets of moves S_1 and S_2. If both moves share at least a certain fraction ϵ of common ancestors, that is if $\epsilon \min(N_1, N_2) < \mathrm{card}(S_1 \cap S_2)$, we assign both moves to the same community. This process is iterated until no more new moves are added to this community. This extracting process is of course empirical, but helps us nevertheless to sort out some subgroups of moves that are different from those extracted with previous methods, provided that the parameter ϵ is carefully tuned. Indeed a too low value of ϵ does not help much in extracting a group as in most cases moves share naturally a certain amount of preceding moves but a too high value of ϵ will not capture anything for a sparse matrix. In our Network III we thus used the range of values $0.3 < \epsilon < 0.7$. Unfortunately there is no typical behaviour of how the size of a community varies with respect to ϵ: this size depends highly on the initial move and on the number of components of an eigenvector on which one is allowed to explore the ancestries.

We have applied this extracting process on our eigenvectors. We thus identify communities in two steps, the first being to select eigenvectors corresponding to the largest eigenvalues of G or G^*, and the second step to follow this ancestry technique. As mentioned earlier an eigenvector corresponding to a large eigenvalue modulus is more likely to be localized on a small number of nodes, therefore one can truncate a given eigenvector to retain its top nodes and apply this method by choosing one of the top nodes as the starting move and constructing the community by successively exploring this subset. Starting from different nodes will allow to identify the different communities. Fig. 5.19 shows that the method is able to extract moves which have common features, much more so that just looking at largest components of the vectors or removing the ranking vectors. Small subsets of moves are disambiguated from the larger groups of the preceding figures, showing sequences which seem to go together with situations of Ko with different black dispositions (first and third eigenvector of Fig. 5.19 left part), black connecting on the side of the board (fourth eigenvector of Fig. 5.19 left part), and so on. Similarly the first line of Fig. 5.19 right part can be associated to attempts by black to take over an opponent's chain on the rim of the board. These examples show that the method is effective to regroup moves according to reasonably defined affinities.

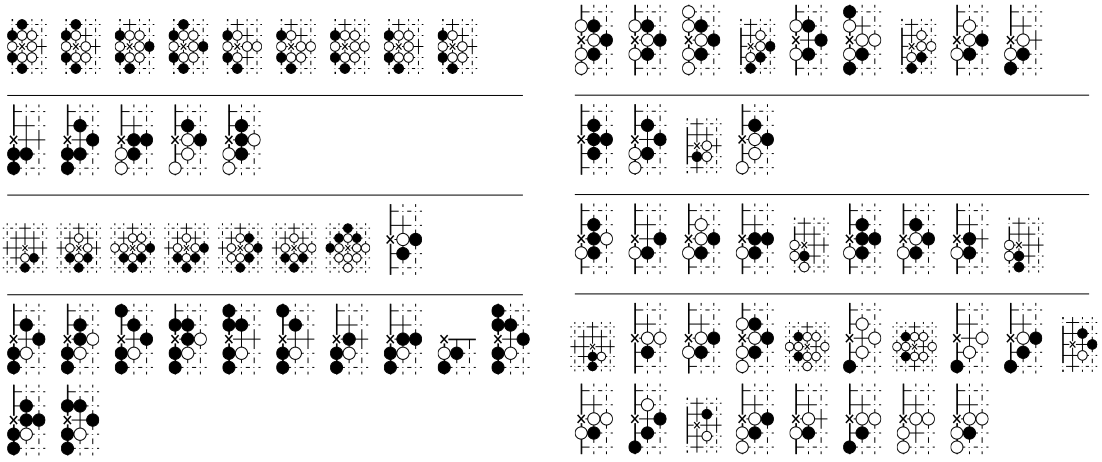

Figure 5.19: *Left panel :* Example of set of moves extracted from eigenvectors of Fig. 5.11 by considering common ancestry of moves with threshold level $\epsilon = 0.3$ applied to λ_7, λ_{11} and λ_{21}, and threshold level $\epsilon = 0.5$ applied to λ_{13}. *Right panel :* Example of set of moves extracted from data of Fig. 5.13 by considering common ancestry of moves with threshold level $\epsilon = 0.3$ applied to λ_7, λ_{11}, λ_{18} and λ_{21}.

We mention an alternative method which gives good results in some instances. It consists in analyzing the angles of an eigenvector components when plotted in a complex plane. This method is not systematic as there exist several real valued eigenvectors but for the complex ones we can observe interesting patterns. Either the plots show a meaningless cloud of points or they can reveal a tendency of a subset of components to be aligned. As shown in an example in Fig. 5.20 there can be one or several directions within the same eigenvector, indicating that maybe the phases of the components can characterize moves sharing common properties. Qualitatively speaking the spatial configuration of these subgroups of moves look similar but there are also similarities between moves having different angles, and a formal understanding of the meaning of phases is still lacking. We note that for undirected networks the sign of components of eigenvectors of the adjacency matrix has been used to detect communities [Krzakala et al., 2013].

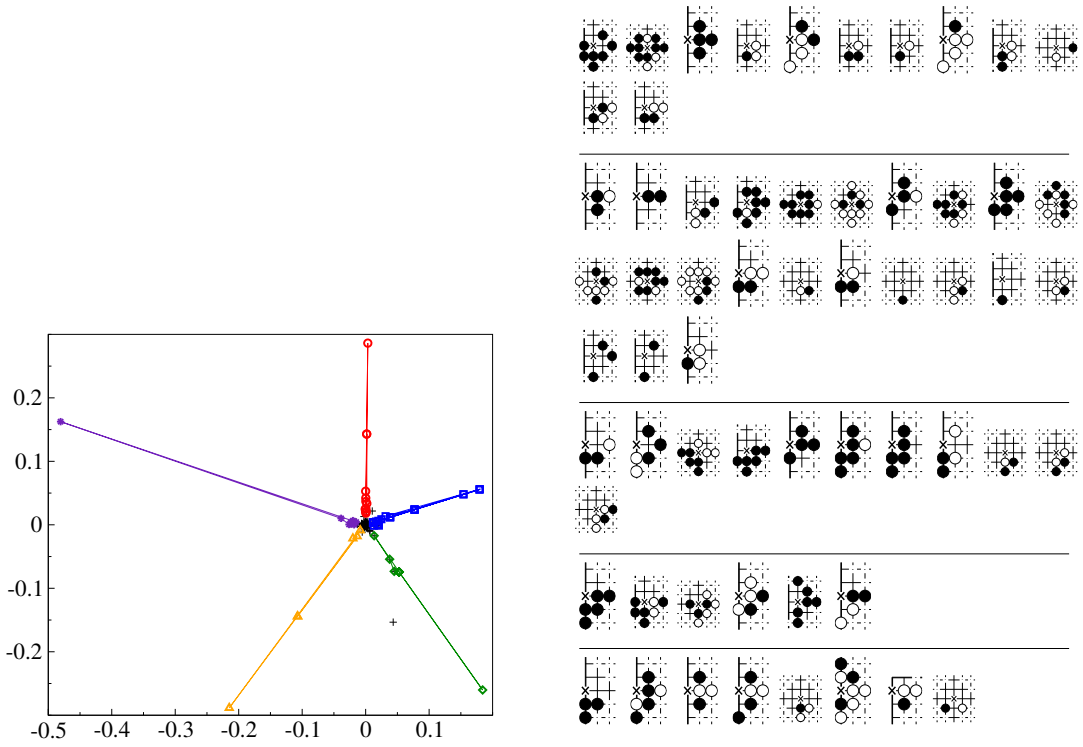

Figure 5.20: Example of community extraction through phase analysis applied on the eigenvector ψ of G^* corresponding to λ_{13}. *Left panel :* eigenvector components in the complex plane. *Right panel :* groups of plaquettes, from top to bottom, correspond to respective symbols red circles, blue squares, green diamonds, oranges triangles and purple stars.

It is in principle not excluded that one should look into combinations of eigenvectors but even though we considered single vectors, the results show that it is possible to extract community of moves which share some common properties with these methods. The combination of methods outlined in this section, namely isolating top moves in eigenvectors associated to large eigenvalues, and disambiguating them through search for common ancestries, seems to yield meaningful groups of moves. We stress again that they do not merely correspond to most played moves or sequences of moves, nor to the best ranked in the PageRank or CheiRank, but give a different information related to the network structure around these moves. It is possible to play with the parameters of the method (threshold ϵ, number of eigenvectors, starting point of the common ancestry) in order to find different sets of communities, which should be analyzed in relation with the strategy of the game, and then could help organize the Monte Carlo go search by running it into specific communities.

5.5 Extension to more generalized networks

We can refine the analysis further by disaggregating the datasets in several ways by constructing different networks from the same database. The number of nodes is still the same, but links are now selected according to some specific criterion and may give rise to different properties.

An important aspect of the games, especially in view of applications to computer go, is to select moves which are more susceptible of winning the game. It is possible to separate the players between winners and losers, but the presence of handicaps makes this process ambiguous. Indeed, it is possible to place up to nine stones before the beginning of the game at strategic locations, giving an advantage to a weaker player which may allow him to play against a better opponent with a fair chance of winning. Another possibility we thus investigated was to separate the players

by their levels according to their dan ranking. In the database [U-go, 2013] the number of dans of the players is known, and it is therefore possible to separate games played at different levels. To explore these differences, we constructed the diamond network from games played by 1d versus 1d, the one from 9d versus 9d, and the one from 6d versus 6d. The left panel of Fig. 5.21 shows the quantity $r_j = \sum_{i \leftarrow j} |k_i - k'_i| / \sum_i k_i$ defined for a pair of networks, where k_i (resp. k'_i) is the number of links from a fixed node j to node i for one network (resp. for the second network). For each node, r_j thus quantifies the difference in outgoing links between two networks. We plot the distribution of this quantity highlighting the difference between the network 1d/1d and the network 9d/9d. We see that they are indeed different, with a mean $\langle r_j \rangle \approx 1.33$. Nevertheless, in the same panel we add for comparison the difference between two networks of 6d/6d, showing that one can also find differences between networks built from players of the same level. In view of this, to see if the difference between 1d/1d and 9d/9d is statistically significant, the right panel shows the average $r = \langle r_j \rangle$ for different choices of samples of 6d versus 6d games and the value for the networks constructed from the games of 1d players and 9d players, with the average taken on top 1500 moves of the PageRank and we see that the difference between 1d players an 9d players has some statistical significance. The quantity r is a simple way of quantifying the structural differences in the networks at the level of outgoing flows which is in our case an indication that 9d players might have an overall structurally different style of play than 1d players, even though the difference is relatively small.

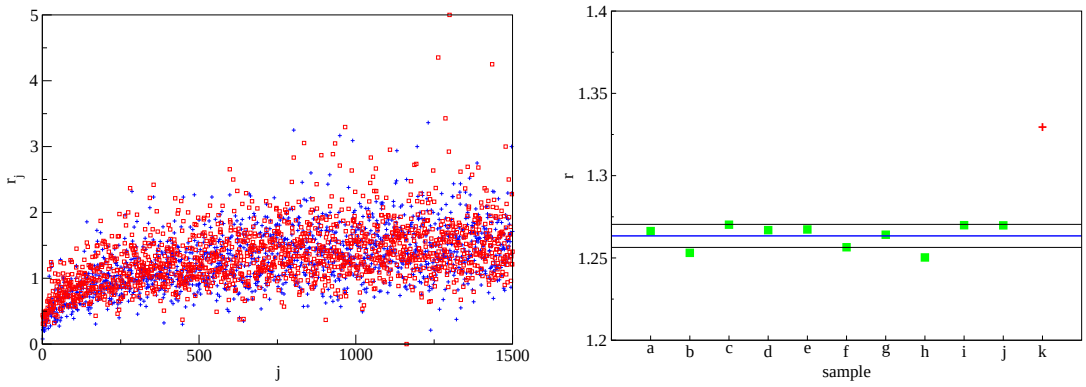

Figure 5.21: *Left panel :* Fluctuation difference $r_j = \sum_{i \leftarrow j} |k_i - k'_i| / \sum_i k_i$ of outgoing links versus move indices for top 1500 moves of diamond patterns in PageRank order (network III). An example of difference is shown between two networks built from games between 6d players (blue crosses) and two networks built respectively from games between 1d players and games between 9d players (red squares). The number of games in each case is 2731, corresponding to the number of 1d/1d games in the database [U-go, 2013]. *Right panel :* Difference r between the networks built from games of 1d players and of 9d players (red cross) together with several examples of r for pairs of networks constructed from different samples of games of 6d players (green squares). The three horizontal lines mark the mean and the variance of the 6d values and the number of games in each sample is 2731.

An other interesting possibility which might also be useful for applications is to create separate networks for different phases of the game. For instance, one can take into account when using the database of real games only the first 50 moves, the middle 50, or the final 50. Again, this does not modify the nodes of the networks, but changes the links, creating three different networks corresponding to respectively beginning, middle, and ending phases of the game. The number of links is now 6155936 for the beginning phase, 6460771 for the middle phase, and 5947467 for the ending phase, instead of 26116006 for the whole game (the numbers without degeneracies for diamond plaquettes are respectively 613953, 2070305 and 3182771). The spectra of the three networks for the diamond plaquettes are shown in Fig. 5.22 (again, only the largest eigenvalues

are calculated). It is clear that the spectra are quite different, indicating that the structure of the network is not equivalent for the different phases of the game. It is visible that the eigenvalue cloud is larger for the ending phase indicating that near the final stage of the game the random surfer gets trapped more easily in specific patterns, which should correspond to typical endgames. Similarly, the gap is smaller for the beginning phase, indicating that one strongly knit community exists with an eigenvalue close to the PageRank value.

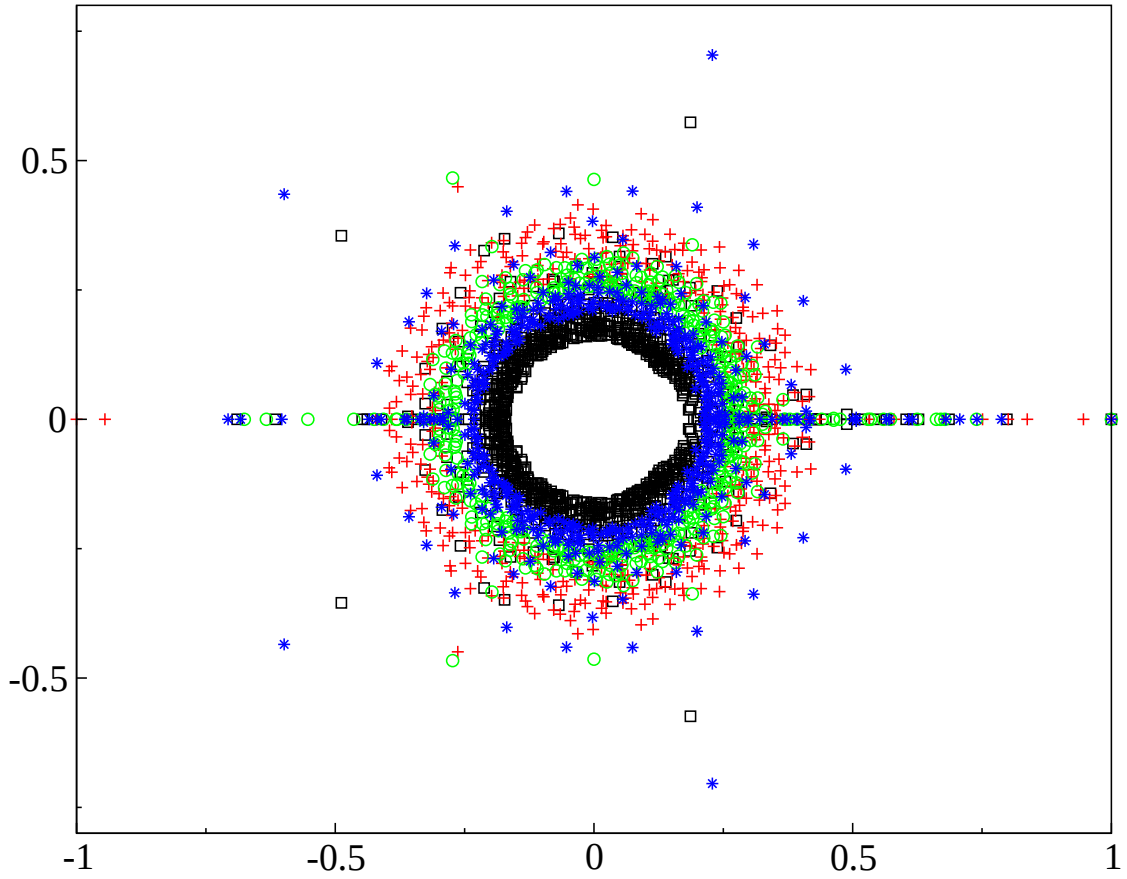

Figure 5.22: Spectrum of G for diamond networks of different game phases : first 50 moves (red crosses), middle 50 moves (green circles) and last 50 moves (blue stars). The black squares correspond to the spectrum of the network when the whole game is taken into account, shown for reference.

The eigenvectors shown in figures Fig. 5.23 to Fig. 5.25 highlight different sets of moves as might be expected since strategy should differ in those phases. Obviously, eigenvectors for opening moves are much more biased towards relatively empty plaquettes, indicating the start of local fights. In the middle and end of the games, communities are biased towards moves corresponding to more and more filled plaquettes, indicating ongoing fights or fight endings. We stress the fact that those sets of moves are not just the most played moves in the respective phases. Running the community detection process discussed above on such eigenvectors should select communities specific to these different phases of the game.

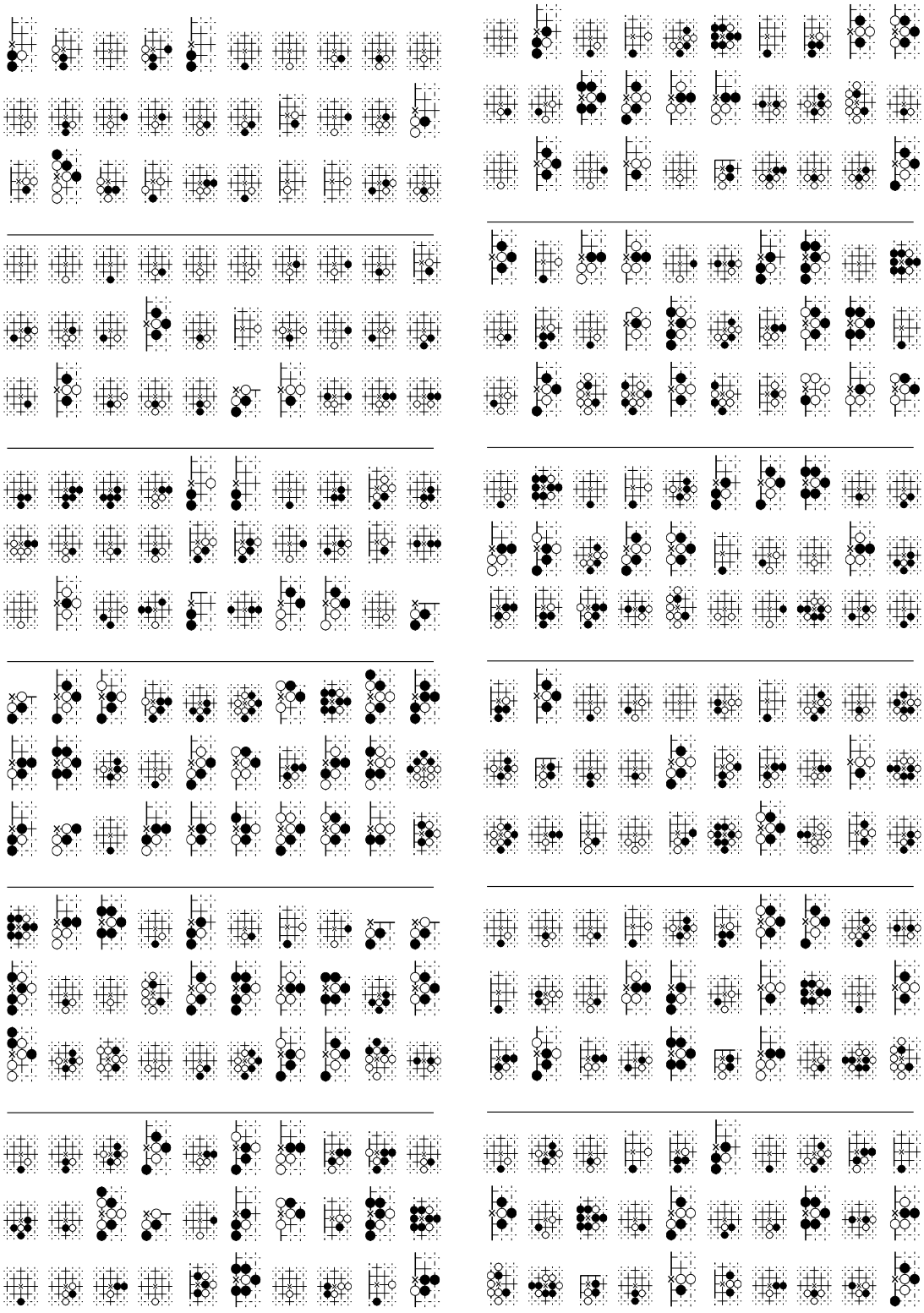

Figure 5.23: Examples of set of top 30 moves where eigenvectors of G localize themselves, those examples are computed for diamond network in starting game phase. *From top to bottom left and top to bottom right :* $\lambda_4 = 0.9460$, $\lambda_9 = 0.6780$, $\lambda_{13} = -0.2632 - 0.4494i$, $\lambda_{17} = 0.4163 + 0.0438i$, $\lambda_{21} = 0.4150$, $\lambda_{32} = 0.2426 - 0.3330i$, $\lambda_{44} = 0.3694 + 0.1626i$, $\lambda_{51} = 0.2799 + 0.2878i$, $\lambda_{61} = -0.1978 + 0.3411i$, $\lambda_{72} = 0.3691 - 0.1291i$, $\lambda_{80} = 0.1471 + 0.3577i$ and $\lambda_{95} = 0.1738 - 0.3401i$.

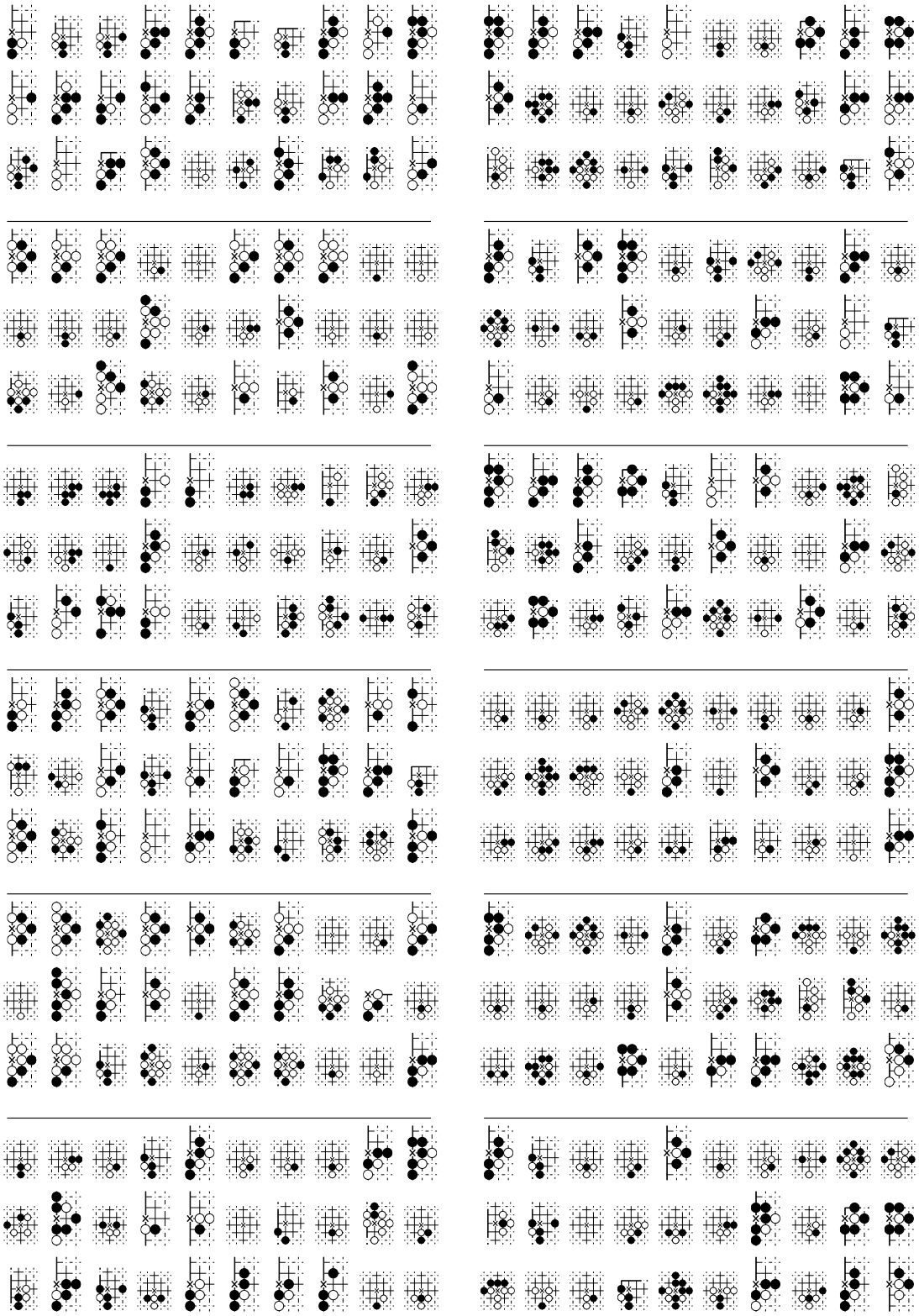

Figure 5.24: Examples of set of top 30 moves where eigenvectors of G localize themselves, those examples are computed for diamond network in middle game phase. *From top to bottom left and top to bottom right :* $\lambda_4 = -0.6757$, $\lambda_9 = 0.5730$, $\lambda_{13} = -0.2733 - 0.4663i$, $\lambda_{17} = -0.4641$, $\lambda_{21} = 0.4633 - 0.0018i$, $\lambda_{32} = -0.3801$, $\lambda_{44} = 0.2451 + 0.2246i$, $\lambda_{51} = 0.1983 - 0.2622i$, $\lambda_{61} = 0.2306 - 0.2262i$, $\lambda_{72} = 0.1573 + 0.2769i$, $\lambda_{80} = 0.0250 + 0.3144i$ and $\lambda_{95} = -0.2368 + 0.2020i$.

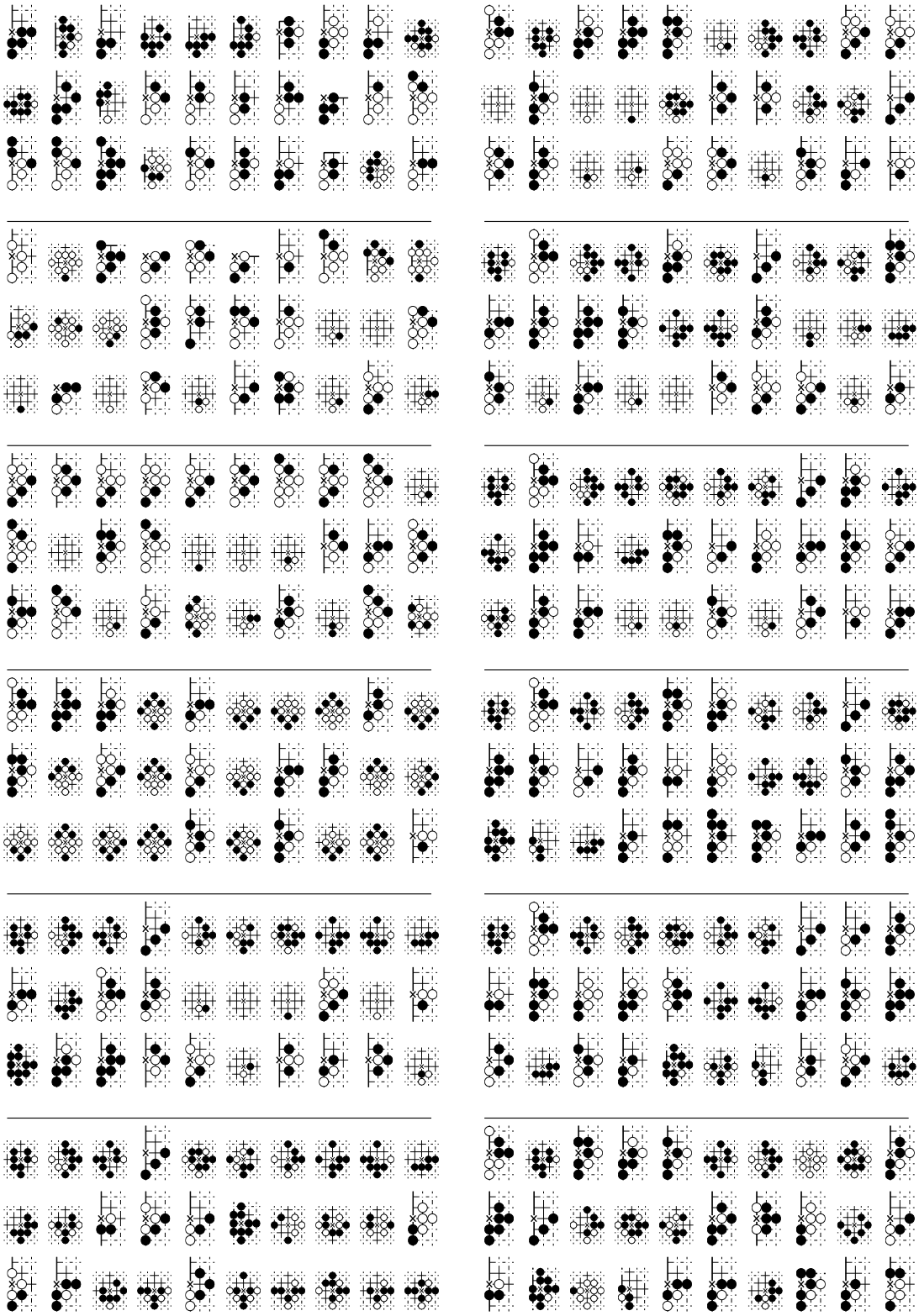

Figure 5.25: Examples of set of top 30 moves where eigenvectors of G localize themselves, those examples are computed for diamond network in ending game phase. *From top to bottom left and top to bottom right :* $\lambda_4 = -0.5989 - 0.4351i$, $\lambda_9 = 0.7071$, $\lambda_{13} = 0.6084$, $\lambda_{17} = 0.5067 + 0.0032i$, $\lambda_{21} = 0.4868 - 0.0963i$, $\lambda_{32} = -0.1686 + 0.4019i$, $\lambda_{44} = 0.4101 - 0.0157i$, $\lambda_{51} = 0.3812 + 0.0669i$, $\lambda_{61} = 0.3296 + 0.1454i$, $\lambda_{72} = 0.1592 - 0.2952i$, $\lambda_{80} = -0.2714 - 0.1837i$ and $\lambda_{95} = -0.1229 - 0.2744i$.

5.6 Conclusion

We have shown that it is possible to construct networks which describe the game of go, in a spirit similar to the ones already used for languages. We have extended the results presented in [Georgeot and Giraud, 2012] by comparing three networks of different sizes according to the size of the plaquettes which serve as nodes of the network. The three networks share structural similarities, such as a statistical correlation (but not an exact symmetry) between incoming and outgoing links. However, the largest network, besides necessitating more refined numerical tools in order to obtain the largest eigenvalues and associated eigenvectors, is also much less connected and disambiguates much better the different moves. We have also shown that specific subnetworks can be constructed, selecting links in the databases according to levels of the players or phases of the game. In general the next to leading eigenvectors in the Google matrix represent a different information from the list of most common moves going beyond the mere frequency count of plaquettes appearance. In fact, these eigenvectors can even sometimes be highly sensitive to rare links, indeed during our analysis one impossible move was highlighted in one of the top eigenvectors. This move had only two links among the several millions, leading us to find a fake gamefile in the dataset. This shows that the network approach can detect specificities that a mere statistical analysis of the datasets will miss. We have proposed various community detection processes, and the knowledge of these communities could be used for instance to initialize the value of moves according to the local pattern, at a value given by the value of its ancestors. It could also be used to propagate the value of a move to similar moves in the context of the existing Go game algorithm and therefore it could help improving the efficiency of Monte Carlo Go.

However there are numerous obstacles that need to be removed before one can apply these findings for a concrete implementation. There is still the need to understand precisely what information are given by which eigenvectors and how to identify them in a systematic and automated manner. There is also the question of quantifying the concept of best moves similarly to the values given to the plaquettes if we want to assess what are the best moves to be used in a specific context.

A possibly useful extension to this Google matrix method would be the consideration of the personalization vector in the teleportation matrix $\mathbf{ve^T}$ such that the arbitrary probability vector suits specific needs for a bias in our network. It will also be fascinating to see if other games such as chess could be modeled this way, and how different the results will be. Besides its applicability to the simulations of go on computers, we also believe that such studies enable to get insight on the way the human brain participates in such game activities. In this direction, an interesting extension of this work could be to compare the networks built from games played by human beings and computers, and determine how different they are.

Chapter 6

The use of PageRank in opinion formation models

6.1 A brief introduction to Sociophysics

In this final section we will get away from the whole Google matrix framework as a tool to analyze directed network topology and we will propose a different use to the information brought by the PageRank vector in a totally different context.

During the last decades various communication technologies drastically changed the social interactions among the individuals in our society. People now share political ideas and form virtual groups using the Internet which easily breaks the barriers of geographical limitations and allow people from different cultures and background to interact with each other. In fact this tendency has become so strong that it is now possible to monitor the spreading of news and rumors by scanning the threads of social medias such as Twitter or Facebook. Some large scale companies also try to implement tools of social network analysis in order to anticipate the arising problems and act effectively[1].

In the same time physicists started to become more and more interested in the various social phenomena where the tools developed for the fundamental problems of statistical physics can be applied to some extent. Of course the complexity of the human being cannot be approximated by a particle like object therefore straight generalization of these kinds are senseless. However there exist some regularities that arise from large group of people and it is precisely these collective behaviours that are investigated from a point of view where the analogies with widely studied physical systems might be useful [Galam, 1986],[Galam, 2005],[Galam, 2008].

For instance among the diverse aspects of sociological problems the opinion formation and its large scale dynamics have recently drawn a lot of interests. Indeed one can see the opinions or the votes as spin states that can be up or down and compare their interactions to the influence of people on each other. Questions related to the propagation of an opinion or the possibility of reaching a consensus state are considered and many more extensions are possible such as looking at cultural and propaganda effects as an external field acting on the system or even associating the individual decision fluctuations to thermal noises.

Some important steps in the analysis of opinion formation have been done with the development of various voter models described in a great details in [Galam, 1986], [Liggett, 1999], [Galam, 2005], [Watts and Dodds, 2007], [Galam, 2008], [Castellano et al., 2009], [Krapivsky et al., 2010].

Here we propose to study the opinion dynamics by including the features of a social network : First we will describe a simple model (PROF) which determines the opinion of an individual based on the PageRank values of its neighbours and second (PROF-Sznajd) we will consider the effects of group of individuals still using the PageRank values as a main component.

[1] See for example the early warning unit of Nestlé *http://www.nestle.com/randd/quality-safety.*

Motivation : From a social network point of view our motivation is twofold. First, the voter models most often consider individuals as agents sitting on regular lattices which is somewhat misleading because in reality the regular grid does not exist and even more so the people are not necessarily influenced by those geographically close to them. On the contrary the structure of the network of acquaintances is complex and the network perspective considering the individuals as nodes allows to catch those complex relationships. Second, we assume that people are more likely to be influenced or follow opinions of their friends that have a high social status, the PageRank values here plays the role of ranking people, or nodes, according to their social importance and therefore allows to implement a system where an individual is mainly looking at his highly ranked friends.

6.2 PageRank Model of Opinion Formation

Model

We propose the simplest model of opinion formation (PROF model) that will be used on real datasets such as Cambridge and Oxford universities webpages networks already discussed in previous chapters so that the scale-free feature and complex connectivity structure are taken into account. The first step consists of computing the PageRank vector with the usual value of $\alpha = 0.85$ in our case.

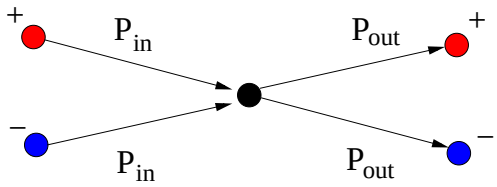

Figure 6.1: A node i (here in black) chooses its opinion by considering its friends which are the nodes directly connected to him.

In addition to that a network node i is characterized by an Ising spin variable σ_i which can take values $+1$ or -1 coded also by red or blue color respectively for clarity. The sign of a node i is determined by its direct neighbors j which have the PageRank probabilities P_j. For that we compute the sum Σ_i over all directly linked neighbors j of node i as shown in Fig. 6.1:

$$\Sigma_i = a \sum_j P^+_{j,in} + b \sum_j P^+_{j,out} - a \sum_j P^-_{j,in} - b \sum_j P^-_{j,out} \tag{6.1}$$

with $a + b = 1$ and where $P_{j,in}$ and $P_{j,out}$ denote the PageRank probability P_j of a node j pointing to node i (incoming link) and a node j to which node i points to (outgoing link) respectively. Here, the two parameters a and b are used to tune the importance of incoming and outgoing links with the imposed relation $a + b = 1$ ($0 \leq a, b \leq 1$). The values P^+ and P^- correspond to red and blue nodes respectively. The value of spin σ_i takes the value 1 or -1 respectively for $\Sigma_i > 0$ or $\Sigma_i < 0$. In a certain sense we can say that a large value of parameter b corresponds to a conformist society where an elector i takes an opinion of other electors to which he points to (nodes with many incoming links are on average at the top positions of PageRank). On the opposite side a large value of a corresponds to a tenacious society where an elector i takes mainly an opinion of those electors who point to him.

The condition on spin inversion can be written via the effective Ising Hamiltonian H of the whole system of interacting spins: $H = -\sum_{i,j} J_{ij}\sigma_i\sigma_j = -\sum_i B_i\sigma_i = \sum_i \epsilon_i$ where the spin-spin interaction J_{ij} determines the local magnetic field B_i on a given node i with $B_i = \sum_j (aP_{j,in} + bP_{j,out})\sigma_j$ which gives the local spin energy $\epsilon_i = -B_i\sigma_i$.

According to these relations the interaction between a selected spin i and its neighbors j is given by the PageRank probability: $J_{ij} = aP_{j,in} + bP_{j,out}$. Thus from a physical view point the whole system can be seen as a disordered ferromagnet [Galam, 2008, Krapivsky et al., 2010]. In this way the spin flip condition corresponds to a local energy ϵ_i minimization done at zero temperature. We note that such an analogy with spin systems is well known for opinion formation models on regular lattices [Galam, 2008],[Castellano et al., 2009],[Krapivsky et al., 2010]. However, it should be noted that generally we have asymmetric couplings $J_{ij} \neq J_{ji}$ that is unusual for physical problems [Galam and Walliser, 2010].

Implementation

The numerical implementation goes as following : Using a standard random number generator we assign an opinion either $+1$ or -1 to each node i such that a fraction f_i of the nodes have, let's say red opinion and $1 - f_i$ have blue opinion. This will define our initial state where each individual has an initial opinion chosen randomly and we will let the system relax according to the opinion flip rule until a stable state is reached. To do it we chose a random visiting order and start picking one of the nodes i of the system and proceed to compute σ_i in order to update its opinion if necessary. Then we pick the second one and repeat the procedure, we do this until the N nodes have been visited once which determine one iteration of the algorithm. We then proceed to do t such iterations.

The reason for choosing a random order is quite straightforward when thinking about the society, indeed there are no particular reason in starting to influence highly ranked members of the society before the common people, rather one gets a better chance of converting a popular individual if the opinion is already shared by a significant amount of people.

During the spin flip condition checking we use the serial update procedure meaning that the opinions of the same time step t are used to convert a node rather than the opinions of time step $t - 1$. This approach is quite unusual in statistical physics but makes sense in the case of opinion dynamics because in reality people who try to influence someone will not use old information as the main argument but they will rather highlight the fact that, at the time of their interaction, other people have already changed their mind giving more weight to their actions.

To construct the density plot we do the averaging over $N_r \leq 10^4$ such random generations of initial states to obtain statistically stable results for final opinion distributions.

Results on Cambridge and Oxford Webpages

Here we present the results of our PROF model considered on the Cambridge and Oxford universities webpages networks discussed in previous chapters, reminding that they have $N = 212710$ and $N = 200823$ nodes respectively with $N_l = 2015265$ and $N_l = 1831542$ links respectively. Both networks have scale-free features and a usual decay rate of PageRank probability as $P(K) \propto 1/K^{\beta}$ with $\beta \approx 0.9$.

The results are presented in terms of fraction of red nodes since by definition all other nodes are blue.

The typical examples of time evolution of the fraction of red nodes $f(t)$ with the number of time iterations t are shown in Fig. 6.2. We see the presence of bistability in the opinion formation: two random states with the same initial fraction of red nodes $f_i = f(t = 0)$ evolve to two different final fractions of red nodes f_f. The process gives an impression of convergence to a fixed state approximately after $t_c \approx 10$ iterations.

We also checked that all node colors become fixed after this convergence time t_c to a fixed state, which is similar to those found for opinion formation on regular lattices where $t_c = O(1)$ [Castellano et al., 2009, Krapivsky et al., 2010, Sood and Redner, 2005].

The results of Fig. 6.2 show that for a random initial distribution of colors we may have different final states with ± 0.2 variation compared to the initial $f_i = 0.5$. However, if we consider that N_{top}

nodes with the top K index values (from 1 to N_{top}) have the same opinion (for instance red) then we find that even a small fraction of the total number of nodes N (e.g. N_{top} of about 0.5% or 1% of N) can impose its opinion for a significant fraction of nodes of about $f_f \approx 0.4$. This shows that in the framework of PROF model the society elite, corresponding to top K nodes, can significantly influence the opinion of the whole society under the condition that the elite members have a fixed opinion between themselves.

We also considered the case when the red nodes are placed on $N_{top} = 2000$ top nodes of CheiRank index K^*, reminding that CheiRank is the stationary probability distribution of the inverted network thus on average $P^*(K^*)$ being proportional to the number of outgoing links, we find that the top nodes with a small f_i values are not able to impose their opinion and the final fraction becomes blue. We attribute this effect to the fact that the opinion flip condition in the PROF model is determined by the PageRank probability $P(K)$ and that the correlations between CheiRank and PageRank are not very strong ([Zhirov et al., 2010, Ermann et al., 2012]).

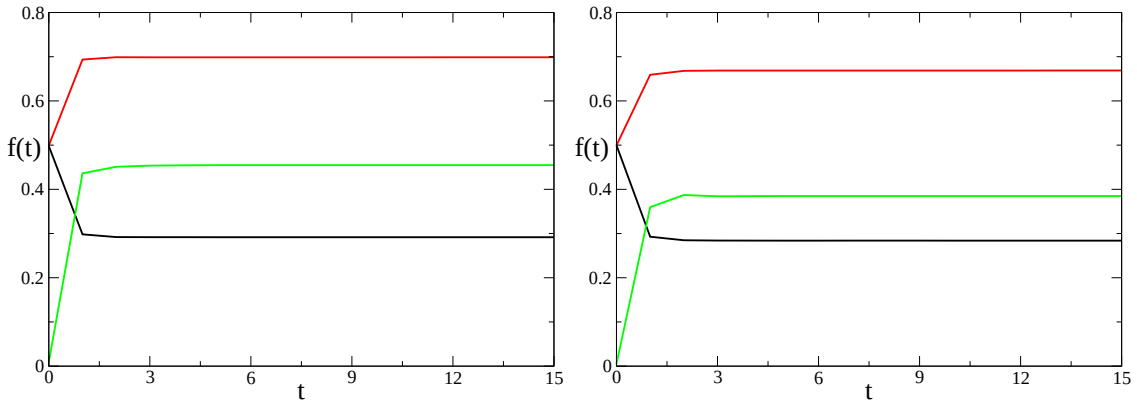

Figure 6.2: Time evolution of opinion given by a fraction of red nodes $f(t)$ as a function of number of iterations t. The red and black curves (top and bottom curves at $t = 15$ respectively) show evolution for two different realizations of random distribution of color with the same initial fraction $f_i = 0.5$ at $t = 0$. The green curve (middle curve at $t = 15$) shows dependence $f(t)$ for the initial state with N_{top} all red nodes with top PageRank K indexes (highest $P(K_i)$ values, $1 \leq K \leq N_{top}$). The evolution is done at $a = b = 0.5$. *Left panel :* Cambridge network with $N_{top} = 2000$. *Right panel :* Oxford network with $N_{top} = 1000$.

To analyze how the final fraction of red nodes f_f depends on its initial fraction f_i we study the time evolution $f(t)$ for a large number N_r of initial random realizations of colors following it up to the convergence time for each realization. We find that the final red nodes are homogeneously distributed in K. Thus there is no specific preference for top society levels for an initial random distribution. The probability distribution W_f of final fractions f_f is shown in Fig. 6.3 as a function of initial fraction f_i at three values of parameter a. These results show two main features of the model: a small fraction of red opinion is completely suppressed if $f_i < f_c$ and its larger fraction dominates completely for $f_i > 1 - f_c$; there is a bistability phase for the initial opinion range $f_b \leq f_i \leq 1 - f_b$. Of course, there is a symmetry with respect to exchange of red and blue colors. For small value $a = 0.1$ we have $f_b \approx f_c$ with $f_c \approx 0.25$ while for large value $a = 0.9$ we have $f_c \approx 0.35$, $f_b \approx 0.45$.

Our interpretation of these results is the following: for small values of $a \to 0$ the opinion of a given society member is determined mainly by the PageRank of neighbors *to whom he points to* (outgoing links). The PageRank probability P of nodes, on which many nodes point to, is usually high since P is proportional to the number of ingoing links. Thus, at $a \to 0$ a society is composed of members who form their opinion listening an elite opinion.

In such a society its elite with one color opinion can impose this opinion to a large fraction of the society. This is illustrated on Fig. 6.4 which shows a dependence of final fraction of red f_f

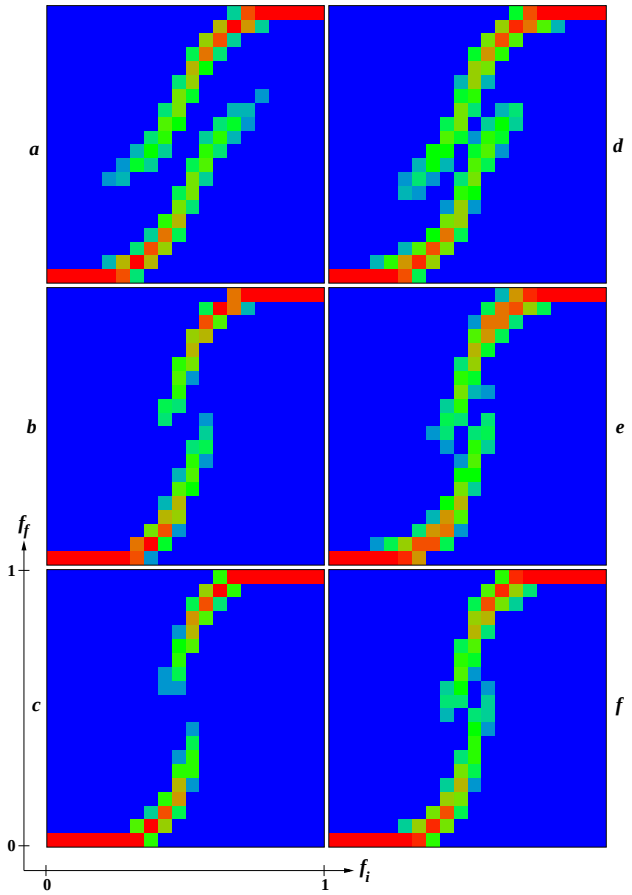

Figure 6.3: Density plot of probability W_f to find a final red fraction f_f, shown in $y-$axis, in dependence on an initial red fraction f_i, shown in $x-$ axis; data are shown inside the unit square $0 \leq f_i, f_f \leq 1$. The values of W_f are defined as a relative number of realizations found inside each of 20×20 cells which cover the whole unit square. Here $N_r = 10^4$ realizations of randomly distributed colors are used to obtained W_f values; for each realization the time evolution is followed up to the convergence time, with up to $t = 20$ iterations; here $T = 0$. *Left column:* Cambridge network (a, b, c); *right column:* Oxford network (d, e, f); here $a = 0.1(a, d)$, $0.5(b, e)$, $0.9(c, f)$ from top to bottom. The probability W_f is proportional to color changing from zero (blue/black) to unity (red/gray).

nodes on parameter a for a small initial fraction of red nodes in the top values of PageRank index ($N_{top} = 2000$). We see that $a = 0$ corresponds to a conformist society which follows in its great majority the opinion of its elite.

For $a = 1$ this fraction f_f drops significantly showing that this corresponds to a regime of tenacious society. It is somewhat surprising that the tenacious society ($a \to 1$) has well defined and relatively large fixed opinion phase with a relatively small region of bistability phase, which is in a contrast to the conformist society at $a \to 0$ when the opinion is strongly influenced by the society elite. We attribute this to the fact that in Fig. 6.3 we start with a randomly distributed opinion, due to that the opinion of elite has two fractions of two colors that create a bistable situation since two fractions of society follows opinion of this divided elite that makes the situation bistable on a larger interval of f_i compared to the case of tenacious society at $a \to 1$.

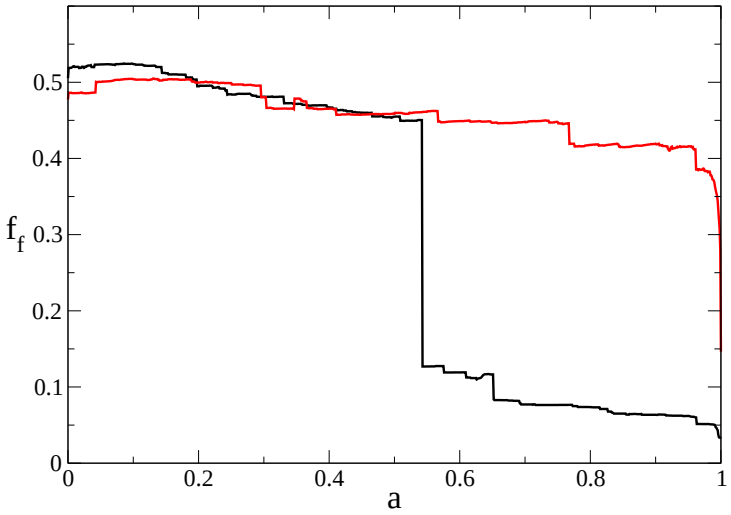

Figure 6.4: Dependence of the final fraction of red nodes f_f on the tenacious parameter a (or conformist parameter $b = 1 - a$) for initial red nodes in $N_{top} = 2000$ values of PageRank index ($1 \leq K \leq N_{top}$); black and red(gray) curves show data for Cambridge and Oxford networks.

6.3 PageRank and Sznajd Model

Model

In this section we will consider a complementary approach, the Sznajd model, which nicely incorporates the well-known trade union principle "United we stand, divided we fall" into the field of voter modeling and opinion formation on regular networks [Sznajd-Weron and Józef, 2000]. The review of various aspects of this model is given in [Castellano et al., 2009].

Here we generalize the Sznajd model to include the features of PROF model and consider it on social networks with their scale-free structure.

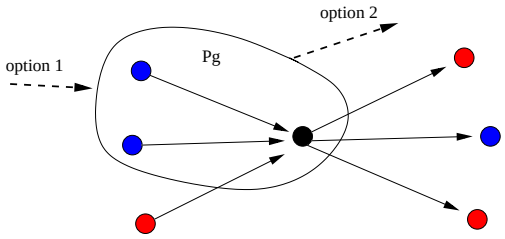

Figure 6.5: A group of size $N_g = 3$ is identified if the node in black happens to have the blue opinion. The group indicated by the circle is made of the highest ranked friends of the considered black node and the influence of this group acts on any node pointing to the group when the option 1 in PROF-Sznajd model is considered or it acts on any node pointing to the group and also any node pointed by the group when the option 2 is considered.

The idea of the original Sznajd model involves several individuals sitting nearby on a regular lattice, forming a *group*, and studies the capacity of this group to influence their individual neighbours. Here our PROF-Sznajd model will take advantage of the network structure to actually define a group of friends or acquaintances, namely the direct neighbours of a node, with an inclination to a higher influential power when the members of the group are socially well ranked thanks to the individual values given by the PageRank vector.

Implementation

Once the PageRank vector of a given network is known, there are two variants of the PROF-Sznajd model whose only parameter is N_g the minimum number of members needed to consider a collection of individuals sharing the same opinion as a group. The algorithm works as :

- *i)* pick by random a node i in the network and consider the polarization of the $N_g - 1$ highest PageRank nodes pointing to it;
- *ii)* if the node i and all its $N_g - 1$ neighbouring nodes have the same color (same spin polarization), then these N_g nodes form a group whose effective PageRank value is the sum of all the individual members values $P_g = \sum_{j=1}^{N_g} P_j$; if it is not the case then we leave the nodes unchanged and perform the next time step;
- *iii)* consider all the nodes *pointing to any member of the group* (this corresponds to the model *option 1*) or consider *all the nodes pointing to any member of the group and all the nodes pointed by any member of the group* (this corresponds to the model *option 2*); then check all these nodes n directly linked to the group: If an individual node PageRank value P_n is less than P_{group} then this node joins the group by taking the same color (polarization) as the group nodes; The PageRank values of added nodes are then added to the group PageRank P_{group}. If it is not the case then the node is left unchanged.

Here again during the converting process no particular order is favored so that a random approach may perhaps reflects better the true social behaviour of groups of people. Moreover if the group cannot convince a node that by chance has the same color, namely the same opinion as the group, then the resisting node does not participate in converting his neighbours because it had his opinion on its own which is unrelated to the actions of the group.

The above time step is repeated many times during time τ, counting the number of steps, by choosing a random node i on each next step and, as before, after choosing a random initial distribution of opinion, we observe the relaxation of the system through the final fraction of red nodes.

Results on Cambridge and Oxford Webpages

A typical example of the time evolution of the fraction of red nodes $f(\tau)$ in the PROF-Sznajd model is shown in Fig. 6.6. It shows that the system converges to a steady-state after a time scale $\tau_c \approx 10N$ that is comparable with the convergence times for the PROF model studied previously. We see that there are still some fluctuations in the steady-state regime that are visibly smaller for the option 2 case because of a larger number of direct links in this case. The number of group nodes N_g gives some variation of f_f but these variations remain on a relatively small scale of a few percents.

Here, we should point on the important difference between PROF and PROF-Sznajd models: for a given initial color realization, in the first case we have convergence to a fixed state after some convergence time while in the second case we have convergence to a steady-state which continues to fluctuate in time, keeping the colors distribution only on average.

The dependence of the final fraction of red nodes f_f on its initial value f_i is shown by the density plot of probability W_f in left panel of Fig. 6.8. The probability W_f is obtained from many initial random realizations in a similar way to the case of Fig. 6.3. We see that there is a significant difference compared to the PROF model : now even at small values of f_i we find small but finite values of f_f while in the PROF model the red color disappears at $f_i < f_c$. This feature is related to the essence of the Sznajd model: here even small groups can resist against totalitarian opinion.

Other features are similar to those found for the PROF model: we again observe bistability of opinion formation. The number of nodes N_g, which form the group, does not affect significantly

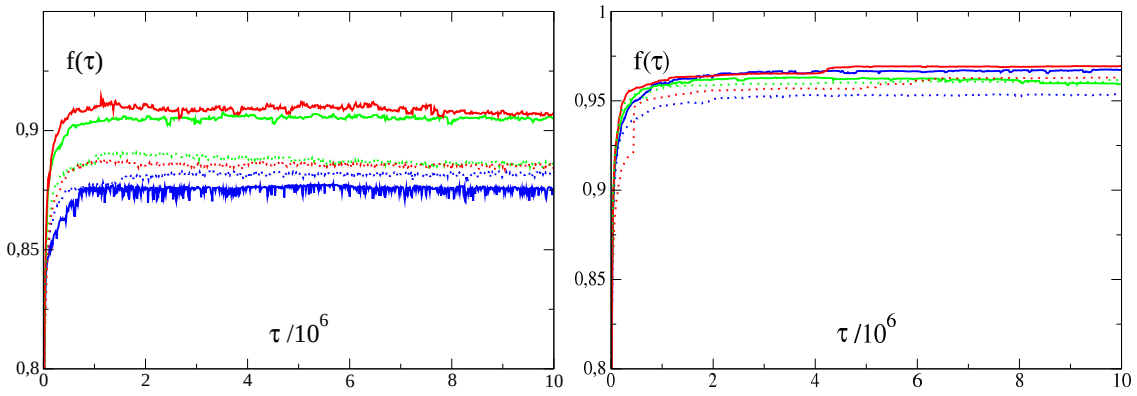

Figure 6.6: Time evolution of the fraction of red nodes $f(\tau)$ in the PROF-Sznajd model with the initial fraction of red nodes $f_i = 0.7$ at one random realization. The curves show data for three values of group size $N_g = 3$ (blue/black); 8 (green/light gray); 13 (red/gray). Full/dashed curves are for Cambridge/Oxford networks; left panel is for option 1; right panel is for option 2.

the distribution W_f, we have smaller fluctuations at larger N_g values but the model works in a stable way already at $N_g = 3$.

The results for the option 2 of PROF-Sznajd model are shown in right panel of Fig. 6.8. In this case the opinions with a small initial fraction of red nodes f_i are suppressed in a significantly stronger way compared to the option 1. We attribute this effect to the fact that large groups can suppress in a stronger way small groups since the outgoing direct links are taken into account in this option.

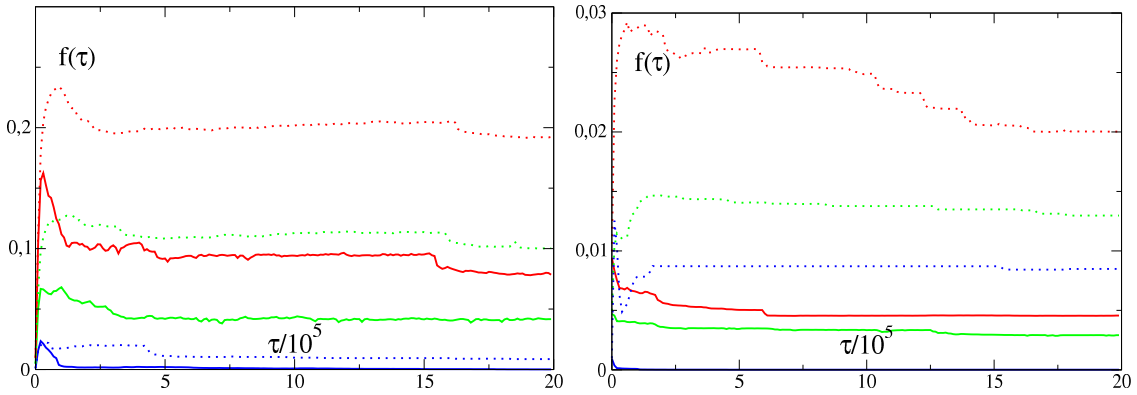

Figure 6.7: Time evolution of the fraction of red nodes $f(\tau)$ in the PROF-Sznajd model with the initial red nodes for the top PageRank nodes: $N_{top} = 200$ (blue); 1000 (green); 2000 (red); here $N_g = 8$. Full/dashed curves are for Cambridge/Oxford networks; left panel is for option 1; right panel is for option 2. Color of curves is red, green, blue from top to bottom at maximal τ on both panels.

The significant difference between the two options of PROF-Sznajd model is well seen from the data of Fig. 6.7. Here, all N_{top} nodes are taken in red. For the option 1 the society elite succeeds to impose its opinion to a significant fraction of nodes which is increased by a factor 5-10. Visibly, this increase is less significant than in the PROF model. However, for the option 2 of PROF-Sznajd model there is practically no increase of the fraction of red nodes. Thus, in the option 2 the society members are very independent and the influence of the elite on their opinion is very weak.

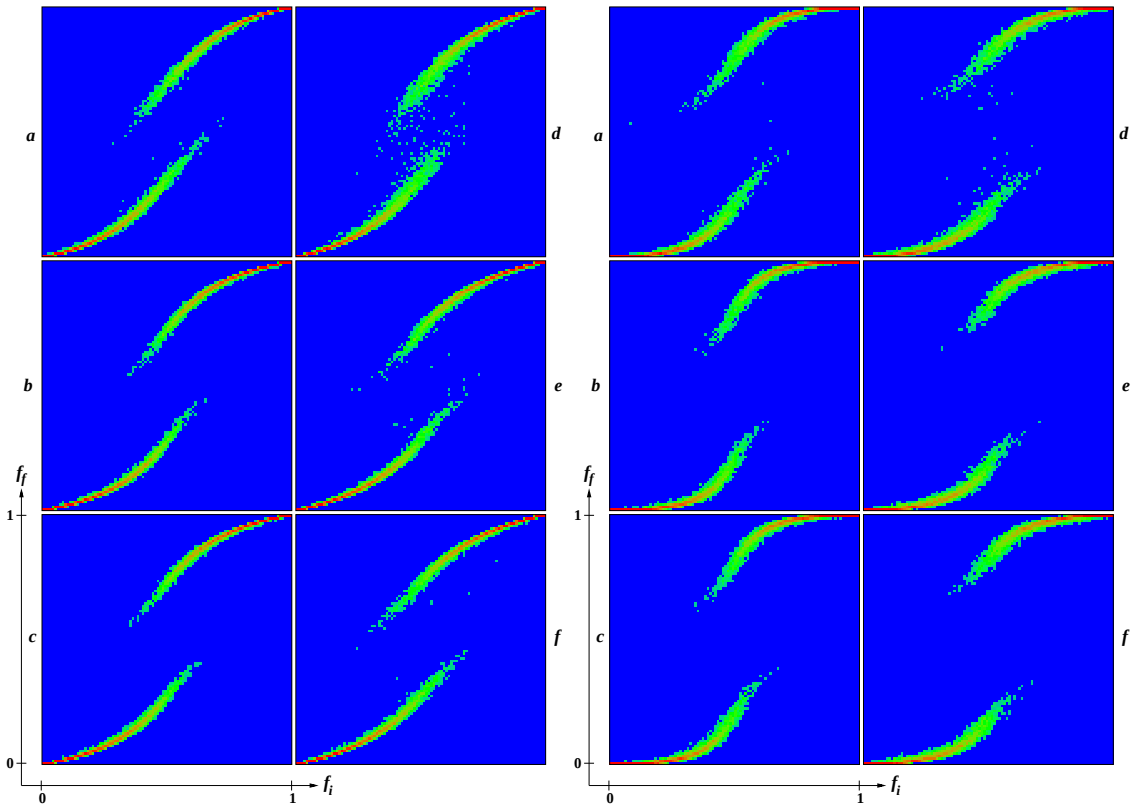

Figure 6.8: PROF-Sznajd model : density plot of probability W_f to find a final red fraction f_f, shown in $y-$axis, in dependence on an initial red fraction f_i, shown in $x-$ axis; data are shown inside the unit square $0 \leq f_i, f_f \leq 1$. The values of W_f are defined as a relative number of realizations found inside each of 100×100 cells which cover the whole unit square. Here $N_r = 10^4$ realizations of randomly distributed colors are used to obtained W_f values; for each realization the time evolution is followed up to the convergence time, with up to $\tau = 10^7$ steps. *Left column:* Cambridge network (a, b, c); *right column:* Oxford network (d, e, f); here $N_g = 3(a, d))$, $8(b, e)$, $13(c, f)$ from top to bottom. The probability W_f is proportional to color changing from zero (blue/black) to unity (red/gray). *Left panel :* Option 1. *Right panel :* Option 2.

6.4 Conclusion

In this section we have seen that the PageRank vector and more generally the complex network approach can be useful in the context of sociophysics. We have proposed a model of opinion formation and tested it on two real webpages networks where we found similar properties that are characterized by the important feature according to which the society elite with a fixed opinion can impose it to a significant fraction of the society members which is much larger than the initial elite fraction. However we have also seen that when the initial opinions of the society members, including the elite, are presented by two options then we find a significant range of opinion fraction within a bistability regime. This range depends on the conformist parameter which characterizes the local aspects of opinion formation of linked society members.

We have also proposed a possible generalization of the Sznajd model for the scale-free social networks where we have observed that finite small size groups can keep their own opinion being different from the main opinion of the majority. In this way the proposed PROF-Sznajd model shows that the totalitarian opinions can be escaped by small sub-communities.

It is clear that the models described here are very simple and cannot precisely represent the

reality with all its complexity, especially despite the fact that the complex network structure was included along with a notion of preferences thanks to the PageRank values, they are limited by the fact that only two states of opinion are allowed.

Nevertheless the modeling of social phenomenon is far from reaching the end and there are plenty of possibility to extend these ideas and add more and more complexity to the whole dynamics.

Chapter 7

Conclusion and Perspective

We have seen throughout this work that there are a wide variety of systems that can be approached via a directed network point of view and, provided that the nodes and the links are carefully defined, we can obtain many useful informations about the organization of the nodes and their relationship to each other which in turn give some insight about the studied system. Due to the omnipresence of the complex networks in various areas of today's research, the need for a deeper understanding of the specificities and the organization of networks increased. Consequently huge interest has been put on the study of complex systems during the last decade and many different approaches and methods were developed to study the topological properties of the networks. The aim of the present thesis was to show that on such directed systems one can apply the Google matrix tools in order to characterize the network structure in an efficient and easy way.

The Google matrix's dominant eigenvector known as the PageRank vector was already proven to be incredibly efficient on large scale-free and directed networks such as the World Wide Web. Here we have seen that it can be used along with the complementary CheiRank vector in order to characterize the networks in two dimensions. We have also discussed in detail the properties of the spectrum and compared the newly studied networks with the known case of the webpages like networks.

These studies were far from comprehensive as there are several opportunities to go further. Indeed apart from the improvements related to a specific topic, as highlighted in the conclusions at the end of each sections for instance, there are ways to explore generic extensions to this Google matrix framework. For example one direct extension could be the introduction of distinction between the types of links : suppose we have r types of directed links in a network of size N then it is possible to build a Google matrix G of size $rN \times rN$ constructed thanks to the connectivity matrix made of $r \times r$ blocks of $N \times N$ matrices thereby describing each directed link between each pair of nodes and differentiating the r different types of edges. In the context of neural networks for instance, this concept could be used to distinguish between different types of neuron connectivity such as axiomatic or somatic connections, excitatory or inhibitatory connections, etc. Also, in [Ermann and Shepelyansky, 2011] the authors made the first step in applying the tools from the Google matrix to the international trade in products between several countries. The exchange of products being described by money transfer, one gets a different network of money flow for each product. Thus, instead of handling the products separately, the edge differenciation technique might be a possible way of handling the whole system at once while keeping the specificities of each products. In a certain sense this method could perhaps be an alternative to the multiplex network approach.

An other straightforward extension would be the modulations allowed by the teleportation matrix $\mathbf{ev^T}$, as mentioned briefly earlier all the work was done with $\mathbf{v^T} = \mathbf{e^T}$ whenever $\alpha \neq 1$. Choosing a different probability distribution $\mathbf{v}$ might yield interesting effects caused by the bias in the random teleportation process. The eigenvalue spectrum will not change but the Google matrix being different will cause an initial state to converge to a different stationary state. Therefore the PageRank will give a different order taking the bias into account. In the case of the game of Go,

we have noticed a greater occurrence of border moves in the top entries of the largest eigenvectors. Tuning the probability vector $\mathbf{v}$ might therefore be an interesting option to give a deliberate bias towards central moves rather than border ones.

When considering the possible extensions, the spirit is that playing with both $\mathbf{v}$ and the link distinction technique allows to incorporate more complicated network definition into the same rigorous framework of the Google matrix theory.

As a final comment to this work, one should keep in mind the main weak point of this framework : the fact that all the network studied should be of fixed size and static. However in some cases, such as the trade network mentioned previously, the time evolution is slow enough to consider a series of static and fixed size networks : it is then possible to build networks as regular temporal slices and study the evolution of the rankings in time (see also [Ermann and Shepelyansky, 2011]).

In real world nearly every network considered is under constant evolution and it is clear that next major step should be the development of a similar framework for studying the networks with dynamical links or networks with varying number of nodes or even networks with both of these qualities.

Appendix Materials

Appendix A - French Summary of the Thesis

Résumé : Introduction

Un *réseau*, ou *graphe* $G = (V, E)$ en terminologie mathématique, est formellement représenté par une collection d'objets appelés *noeuds* (V de l'anglais *vertex*) , ayant des relations entre eux. Ces relations, appelés *liens* (E de l'anglais *edges*), décrivent les interactions entre les objets de cette collection. Historiquement, le grand mathématicien Euler avait déjà utilisé des notions de la théorie des graphes en considérant le fameux problème des ponts de Königsberg où il était question de savoir s'il était possible de visiter tous les ponts et de revenir au point de départ tout en ne visitant chacun des ponts qu'une seule fois. La résolution de ce problème faisait déjà intervenir la notion de topologie des graphes. Dans le cadre de ce travail, nous allons nous restreindre au cas le plus simple en considérant des graphes de taille fixe N (désignant aussi le nombre de noeuds du système) et statique (le nombre de liens L étant également fixé).

Dans les années 1950, la théorie des graphes a connu un grand essor grâce au modèle du graphe aléatoire (*random graph model*), développé par Paul Erdös et Alfréd Rényi, qui permit d'établir de nombreux résultats concernant la topologie des graphes. Plus tard la communauté de physiciens a également contribué de façon importante à l'étude des graphes sous forme de réseaux complexes. La théorie des réseaux complexes étant très riche et vaste, il nous est impossible de fournir une introduction complète ici. Nous allons donc couvrir quelques notions de bases suffisantes pour comprendre l'ensemble de la thèse.

Il existe plusieurs types de liens entre les noeuds, lorsque rien n'est spécifié on considère en général des liens simples reliant deux noeuds du réseau. Dans le cadre de cette thèse nous allons nous concentrer sur l'étude des réseaux dirigés, il s'agit d'un réseau dont les liens sont munis d'un sens. Ainsi un noeud pointe vers un autre, créant ainsi la directionnalité et permettant le classement des liens en deux catégories : les liens *entrants* et les liens *sortants*. Chaque lien sortant d'un noeud est évidemment un lien entrant d'un autre noeud. Techniquement, un lien simple peut également être représenté par une paire de liens dirigés entre les deux noeuds pointant mutuellement l'un vers l'autre. Des boucles peuvent également exister dans les réseaux dirigés, ce sont des noeuds qui se pointent vers eux-mêmes. De manière générale il est souvent plus simple et suffisant de considérer des réseaux non dirigés, cependant la directionnalité ajoute une information supplémentaire intéressante sur l'organisation des noeuds du réseau. Dans certains cas il est même plus naturel d'utiliser l'approche des réseaux dirigés, par exemple les deux notions suivantes sont souvent confondues : Internet, qui est en réalité un ensemble d'ordinateurs interconnectés est non dirigé tandis que le WWW (Wolrd Wide Web), qui lui est un ensemble de documents pointant les uns vers les autres et construits sur Internet, est par nature un réseau dirigé.

Lorsque plusieurs liens connectent deux noeuds du réseau, on parle de liens pondérés et le nombre de fois que le lien est répété est la *multiplicité*. La pondération d'un lien peut être interprétée comme une probabilité lorsqu'elle est correctement normalisée et on parle alors d'un seul lien possédant un certain poids.

La notion de base en théorie des graphes est le degré d'un noeud (*degree* en anglais). Cette quantité décrit le nombre de voisins directes que possède un noeud lorsque la multiplicité n'est

pas prise en compte, ainsi un noeud de degré k possède k voisins. Cette notion peut être aisément étendue aux réseaux dirigés en considérant deux quantités pour chaque noeud : le degré entrant et le degré sortant correspondant respectivement aux degrés des liens entrants et des liens sortants.

Un graphe possède donc un ensemble N de valeurs degrés ($2N$ pour les graphes dirigés). La distribution de ces valeurs $p(k)$ représente la probabilité qu'un noeud tiré aléatoirement possède k connexions, c'est donc une quantité cruciale qui décrit la structure d'un graphe au niveau statistique.

La généralisation du concept de degré nous mène à la notion de la longueur de chemin (*path length* en anglais). Il s'agit intuitivement du nombre d'étapes nécessaires pour atteindre un noeud B à partir d'un noeud A, ainsi la longueur l du chemin entre A et B est une séquence de l liens qui relie ces deux noeuds. La notion du plus court chemin entre deux noeuds du réseau est très souvent considérée via la distribution de ces longueurs comme une caractéristique statistique importante du graphe. La grandeur typique est alors le plus court chemin moyen, donnant la distance typique séparant deux noeuds du système.

Grâce à la théorie des graphes aléatoires, de nombreuses propriétés topologiques ont été étudiées de façon mathématiquement rigoureuse. Cependant, le traitement mathématique nécessite souvent certaines simplifications conduisant à l'étude de cas particuliers comme par exemple les graphes réguliers, les graphes simples, les arbres, etc. En réalité cette théorie est un ensemble de tous les graphes possibles avec une contrainte spécifique telle qu'un nombre de noeuds fixé et un nombre de liens fixé. En pratique, une instance de ce modèle est construite en connectant aléatoirement des paires de noeuds. Ce processus conduit en moyenne à une distribution de Poisson de degrés $p(k) = e^{-\bar{k}}\bar{k}^k/k!$ dans la limite des graphes de grandes tailles, avec $\bar{k}$ comme degré moyen d'un noeud du réseau.

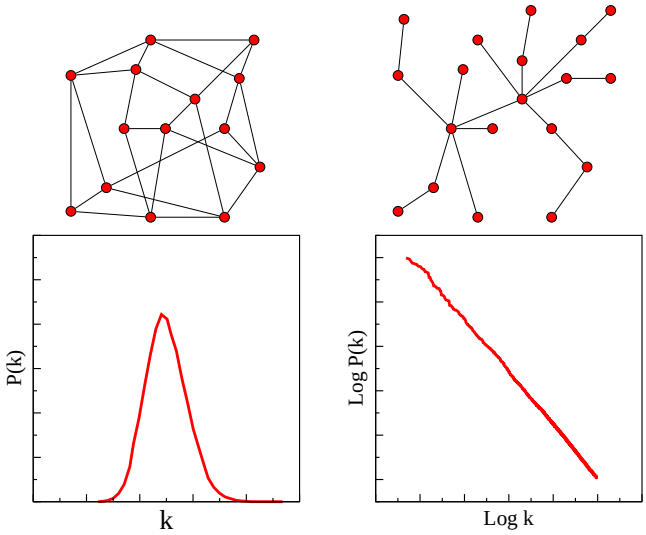

Figure A.1: Exemple illustratif des deux types de graphes avec leur distribution de degrés en dessous : un graphe classique aléatoire (gauche) et un graphe sans échelle (droite).

Durant la fin des années 1990, une révolution a eu lieu parmi les physiciens lorsqu'il s'est avéré, par des observations empiriques, que de nombreux réseaux de la vie courante (dont le WWW) possédaient une distribution de degrés suivant une loi de puissance $p(k) \propto k^{-\gamma}$ avec une valeur typique de l'exposant $2 \leq \gamma \leq 3$. Cela contraste avec les modèles de graphes aléatoires, en effet la loi de puissance indique qu'il n'y a pas d'échelle caractéristique d'où le terme réseau sans échelle (*scale-free* en anglais).

En plus de cette observation, une autre découverte connue sous le nom de l'effet *small-world* a suscité l'intérêt de la communauté. Cet effet se manifeste par le fait que le plus court chemin moyen

s'avère être relativement petit par rapport à l'immensité des réseaux considérés, la dépendance avec sa taille est logarithmique : $\bar{l} \propto logN$.

Par opposition aux graphes simples, ces propriétés structurelles des réseaux complexes conduisent à un comportement fondamentalement différent vis à vis de la propagation de l'information et de l'organisation des noeuds. Il est donc crucial d'étudier et de comprendre en profondeur ces caractéristiques car de nombreux systèmes réels et artificiels montrent une organisation similaire aux réseaux complexes. Il n'y a que peu de systèmes naturels pouvant être décrit par un graphe simple.

Parallèlement au développement de la théorie des réseaux complexes, la technologie a également grandement progressé pour donner naissance au formidable réseau de documents audiovisuels et écrits que constitue le web. Le succès est tel que le web ne cesse de grandir en incorporant de plus en plus de contenu : une estimation menée en 2013 met le nombre de sites web actifs à environ $5 \cdot 10^8$. Tout ce réservoir de documents et de connaissances est aisément accessible non seulement via des ordinateurs mais également des appareils de télécommunications et autres accessoires de technologie moderne. Il est donc crucial de mieux comprendre le comportement des réseaux de si grande taille. Un des plus gros challenge est naturellement de se retrouver parmi cet océan de documents. Contrairement aux archives physiques il n'y a pas de système d'indexation par catégories ni de hiérarchie entre les documents. Il n'y a pas de système centralisé permettant l'accès facile à ce que l'on cherche. Dès les débuts du web, les chercheurs ont élaborés des logiciels complexes, les moteurs de recherches, dont le but est d'explorer les pages web et construire des tables d'indexations pour que les utilisateurs puisses retrouver l'information recherchée. Les premiers moteurs de recherches basés sur la détection des mots clés dans le corps des documents montrèrent vite leurs limites. Il est en effet très difficile d'automatiser le traitement du langage et de déterminer le sens linguistique exprimé par l'utilisateur. Il devenait donc crucial d'abandonner cette approche pour améliorer cette technologie.

C'est dans ce contexte que durant les années 1995 et 1996 deux étudiants doctorants, Sergey Brin et Larry Page, ont développé une idée brillante et révolutionnaire qui les mènera à fonder la société Google, le leader en matière de services Internet. Le succès de leur moteur de recherche s'appuie sur une vison différente de l'importance d'un document. En effet ils ont considéré le réseaux dirigé des pages web comme une sorte de système de recommandation : Si sur mon site web je mets un lien vers un site que j'estime intéressant par rapport au contenu de mon site, j'indique que le site de référence est important pour moi et en pratique je mets un lien pointant de mon site vers la page de référence. L'importance d'un site web, ou d'un noeud du réseau, est donc proportionnel au nombre de liens entrants qu'il possède. De même, la valeur de ma recommandation diminue si j'ai tendance à donner aisément beaucoup de références. Ainsi le poids de ma recommandation diminue en fonction du nombre de sites qui la partage. Finalement le système permet de décrire de manière récursive l'importance d'un noeud car cela signifie qu'un noeud important est pointé par beaucoup de noeuds eux-mêmes importants. Ce système d'attribution des valeurs pour chacun des noeuds d'un réseau est en fait une mesure de centralité donnée par un vecteur de distribution de probabilité appelé le PageRank. Ce vecteur, que l'on peut calculer efficacement, nous fournit un ranking des noeuds du graphe considéré.

Le but de cette thèse est d'explorer l'utilisation de ce système à d'autres réseaux naturels ou artificiels afin d'apporter un autre angle de vue aux problèmes considérés et de comparer les propriétés topologiques des réseaux sous-jacents avec les réseaux bien connus que sont les pages web. Nous allons également montrer que de nombreux systèmes à différentes échelles peuvent être abordés sous la perspective des réseaux dirigés, ainsi nous allons appliquer ces méthodes à l'analyse statistique des chaînes d'ADN (chapitre 2), l'analyse de la connectivité des neurones de *C.elegans* (chapitre 3) ainsi que l'analyse des coups joués lors d'une partie du fameux jeu de Go (chapitre 4). Avant de conclure nous allons également proposer une utilisation différente du PageRank dans le cadre de la modélisation de la formation d'opinions en socio-physique.

Résumé : La matrice Google

La théorie mathématique à l'origine des méthodes de la matrice Google est la théorie des chaînes de Markov. Historiquement, le grand mathématicien Andrei Markov développa vers 1906 un outil issu de la théorie des probabilités pouvant décrire des processus stochastiques connaissant des transitions. Cet outil peut être utilisé pour décrire de nombreux phénomènes dans des domaines variés tels que la physique, la biologie, les finances, etc.

L'idée novatrice des fondateurs de Google était de considérer un surfer aléatoire qui se promène par clics successifs d'un site web à un autre sur la toile. Le comportement de ce surfer serait de suivre aléatoirement des liens hypertextes et de sauter aléatoirement de temps à autre afin d'explorer l'entièreté du réseau. Si l'on attend suffisamment longtemps, le temps que passera le surfer aléatoire sur chacun des sites (ou noeuds du réseau) déterminera son importance relative. Cette idée correspond grossièrement au comportement des internautes qui montrent effectivement une tendance à suivre un ensemble de liens et de temps à autre quitter la zone pour aller chercher une information complètement différente ailleurs et recommencer le processus.

La formule originale du calcul du PageRank fournie par Brin et Page est une sommation où le score $p(i)$ affecté à un noeud (ou site web) i est déterminé par la somme de tous les scores PageRank des sites pointant vers i :

$$p_{t+1} = \sum_{j \in B_i} \frac{p_t(j)}{|j|} \tag{A.1}$$

où B_i est l'ensemble des sites pointant vers i et $|j|$ le nombre total de liens sortants depuis le site j. Cette formule est itérative puisqu'elle nécessite la connaissance des scores de tous les sites. Ainsi au temps t les scores calculés au temps $t-1$ sont utilisés et au temps t_0 une distribution uniforme $p_0(i) = 1/N \quad \forall i$ est utilisée, N étant le nombre de sites web indexés par le moteur de recherche.

Cependant, sous cette forme la formule itérative ne garantit pas l'attribution correcte des scores pour chaque site i. En effet il n'est pas sûr que les scores convergent, les scores ne sont peut-être pas non plus uniques et dépendraient de la distribution $\mathbf{p_0}$ initialement choisie. D'un point de vue physique, l'image du surfer effectuant la marche aléatoire sur un graphe permet de comprendre d'où pourrait provenir ces obstacles. En effet il y a deux situations problématiques : Le surfer pourrait atteindre un noeud qui ne possède pas de liens sortants, il se retrouverait donc coincé sur ce site particulier qui absorberait la probabilité à chaque itération faussant ainsi la distribution des scores. Un tel noeud est appelé *dangling node* en anglais. L'autre situation problématique se manifeste lorsque le surfer aléatoire tombe dans une zone où un ensemble de noeuds sont connectés entre eux avec aucune porte de sortie. Il ne s'agit pas de dangling node à proprement parler mais cet ensemble de noeud agit comme un puits de probabilité piégeant le surfer dans une zone restreinte du réseau.

Pour résoudre ces deux problèmes potentiels, nous allons passer en représentation matricielle et construire la matrice de Google G dont le PageRank sera un vecteur propre. Pour construire cette matrice il faut tout d'abord définir la matrice adjacente asymétrique A qui est une matrice carrée de taille $N \times N$ pour un réseau avec N noeuds. Ces derniers sont représentés le long des colonnes et des lignes de la matrice. Nous allons prendre la convention de noter les liens sortants en colonnes ainsi l'élément de matrice $A_{ij} = m$ lorsqu'il y a m liens partant de j vers i.

Ensuite nous devons normaliser les colonnes de la matrice $A'_{ij} = A_{ij} / \sum_i A_{ij}$, tous les flots sortants sont donc considérés sur le même pied d'égalité permettant ainsi de comparer l'efficacité des connections plutôt que le volume.

Les *dangling nodes* se présentent sous forme de colonnes de 0, pour contourner ce problème nous allons remplacer tous les éléments de toutes les colonnes nulles par $1/N$ ce qui permet d'obtenir

une matrice correctement normalisée. Ce remplacement se traduit par un ajout de saut aléatoire lorsque le surfer atteint un tel noeud. Formellement nous avons donc à ce stade une matrice stochastique $S = A' + (1/N)\mathbf{e}\mathbf{d}^T$ où $\mathbf{e}$ est un vecteur colonne de 1 et $\mathbf{d}$ un vecteur colonne binaire indiquant si le noeud i est un dangling node ou non pour chacun des noeuds du réseau.

Il peut arriver que cette procédure ne soit pas suffisante pour garantir la convergence et l'unicité du vecteur PageRank. Cela est dû aux zones qui peuvent piéger le surfer aléatoire, ainsi pour remédier à ce second obstacle on ajoute une matrice de téléportation de rang 1 dont le but est de téléporter de temps à autre le surfer vers une autre partie du réseau. Ceci permet d'explorer l'entièreté du réseau lorsqu'on attend suffisamment longtemps et cela même avec la présence des puits de probabilités. Mathématiquement cet ajout consiste à rendre la matrice stochastique S primitive, cela est traditionnellement fait en ajoutant la matrice $E = (1/N)\mathbf{e}\mathbf{e}^T$ et la forme finale de la matrice de Google est donnée par :

$$G = \alpha S + (1 - \alpha)\frac{1}{N}\mathbf{e}\mathbf{e}^T \tag{A.2}$$

où α est un paramètre arbitraire, appelé facteur d'amortissement (ou *damping factor* en anglais), et choisi dans l'intervalle $0 \leq \alpha \leq 1$. Ainsi les transitions sont décrites par la structure du réseau avec une probabilité α et par des sauts aléatoires avec une probabilité $1 - \alpha$.

Cette définition de la matrice de Google G garantit qu'elle soit stochastique, irréductible et apériodique. La matrice G appartient à la classe des opérateurs de Perron-Frobenius et le théorème de Perron-Frobenius peut donc être appliqué. La matrice G possède donc une valeur propre dominante $\lambda = 1$ qui est également son rayon spectral.

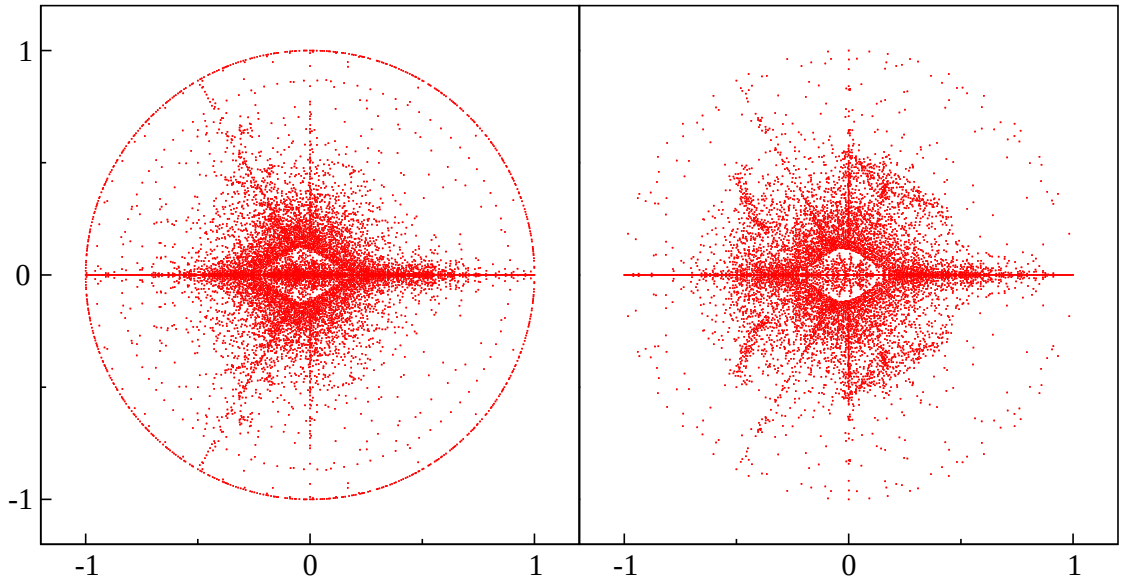

Figure A.2: Spectre de la matrice Google calculé avec $\alpha = 1$ pour le réseau des pages web de l'université de Cambridge (gauche) et Oxford (droite). Toutes les valeurs propres ne sont pas montrées.

De plus le vecteur propre $\mathbf{p}$ associé à $\lambda = 1$ est strictement positif et peut être renormalisée par $p'(i) = p(i)/\sum_i p(i)$ afin de donner une interprétation probabiliste aux scores attribués aux noeuds. Ce vecteur est précisément le PageRank et ses éléments peuvent être réordonnés dans l'ordre décroissant donnant alors l'ordre d'importance relative des différents noeuds du réseau.

Pour des systèmes de taille modeste (jusqu'à environ $N \approx 10^4$) il est possible de diagonaliser la matrice G et obtenir directement les valeurs propres et vecteurs propres associés. Cependant pour des réseaux de très grande taille, tel que pour les sites web, il n'est techniquement pas possible de

gérer et de stocker en mémoire cette matrice. Pour l'application de Brin et Page, il leur suffisait d'obtenir le vecteur propre principal associé à $\lambda = 1$, ils ont donc opté pour une autre méthode permettant de calculer le PageRank sans nécessiter la diagonalisation de la matrice. Parmi les multiples méthodes numériques existantes, la technique la plus adaptée pour leur cas était la méthode des puissances. Elle consiste à résoudre le système suivant :

$$G\mathbf{p} = \mathbf{p} \tag{A.3}$$

en tirant partie des avantages du produit matrice-vecteur. En effet, nous savons par le théorème de Perron-Frobenius que n'importe quelle distribution initiale converge vers un unique état stationnaire, ainsi il suffit de choisir un vecteur arbitraire initial et calculer itérativement son produit avec la matrice G suffisamment de fois pour atteindre un seuil de convergence acceptable. Cette façon de procéder est très efficace dans le cadre des sites web puisqu'en moyenne il n'y a qu'une dizaine de liens sortants non nuls par page web. La structure creuse de la matrice permet de calculer rapidement et sans devoir stocker la matrice dans sa forme complète.

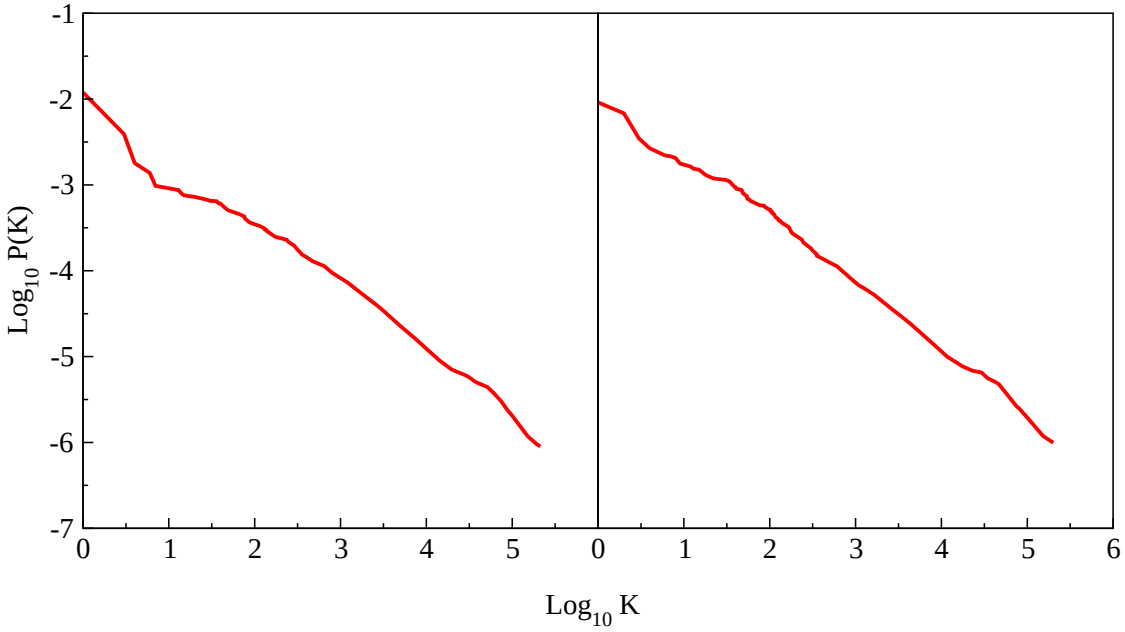

Figure A.3: Décroissance de la probabilité du PageRank pour Cambridge (gauche) et Oxford (droite), calculé avec la méthode des puissances et $\alpha = 0.85$.

Un exemple de spectre de valeurs propres pour deux réseaux de pages web est montré dans la fig. A.2 ainsi que la décroissance des probabilités du vecteur PageRank pour ces réseaux là dans la fig. A.3. Les valeurs propres sont distribuées dans le plan complexe dans le cercle unité, l'espacement entre $\lambda = 1$ et la valeur propre suivante indique la connectivité structurelle du réseau. Ainsi si l'espacement est faible ou inexistant, il existe des zones isolées du réseau mais dont les noeuds sont fortement connectés entre eux. La structure des valeurs propres de grande magnitude peut indiquer la complexité de l'organisation du réseau étudié, ces propriétés sont discutées grâce aux diverses applications dans les chapitres suivants.

Les propriétés du PageRank y seront également illustrées : la décroissance des probabilités peut être approximée de manière satisfaisante par une loi de puissance $p(K) \sim 1/K^{\beta}$ où K est l'index ordonné d'importance des noeuds (les noeuds importants sont en $K = 1, 2, ...$) et l'exposant β est relié à l'exposant de la distribution des liens entrants $p(k) \sim 1/k^{\mu}$ par la formule $\beta = 1/(\mu - 1)$.

Nous allons appliquer ces méthodes à l'analyse des séquences d'ADN dans le chapitre suivant.

Résumé : L'analyse des séquences d'ADN

Pour notre première application nous allons considérer un système à l'échelle moléculaire et constituant la base de la vie dans toutes ses formes : l'ADN. L'acide désoxyribonucléique est une longue chaîne de composés chimiques de base codant l'information génétique des êtres vivants et servant à la régulation et au fonctionnement de l'organisme. L'entièreté de l'information est stockée, répliquée et lue grâce à un ingénieux système de complémentarité de bases. Il existe quatre types de bases : l'adénine (A), la guanine (G), la cytosine (C) et la thymine (T). Les bases A et T s'apparient ensemble de même que les bases C et G, ceci permet la stabilisation de la structure en double hélice de l'ADN et facilite la copie de l'information d'un brin par complétion. Les récents progrès technologiques ont permis de déterminer avec grande précision et à moindre coût l'arrangement exacte de ces bases caractérisant ainsi l'ensemble du code génétique d'un organisme. De nos jours, la quantité de ces données est telle qu'il devient intéressant d'étudier les propriétés statistiques de ces séquences sous la perspective d'une chaîne symbolique. Ici nous allons approcher l'étude statistique des séquences de 5 différentes espèces (taureau, chien, éléphant, poisson zèbre et humain) avec le point de vue des réseaux dirigés.

Il existe de nombreuses bases de données accessibles contenant les codes génétiques de plusieurs espèces, les données sont régulièrement mises à jours avec de nouvelles versions de plus en plus précises. Les données que nous avons utilisées ont une longueur d'environ $L \approx 2 \cdot 10^9$ sauf pour l'homme $L \approx 10^{10}$ car les données sont constituées de la concaténation des séquences de 5 individus.

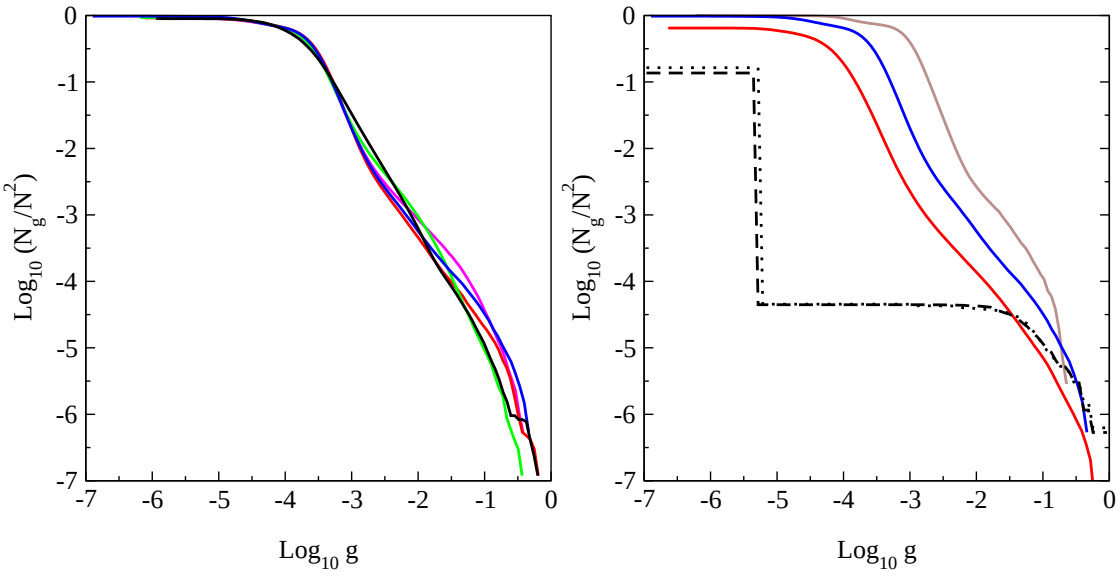

Figure A.4: Fraction intégrée N_g/N^2 des éléments de la matrice Google avec $G_{ij} > g$ en fonction de g. *Panel gauche :* Différentes espèces avec $m = 6$: taureau (magenta), chien (rouge), éléphant (vert), humain (bleu) et poisson zèbre (noir). *Panel droite :* Donnée pour la séquence humaine avec $m = 5$ (brun), $m = 6$ (bleu), $m = 7$ (rouge). On montre pour comparer les courbes noire traitillée et noire pointillée représentant la même distribution pour le réseau de pages web (2006) de Cambridge et d'Oxford respectivement.

Nous construisons le réseau à partir des séquences en lisant la chaîne dans un certain sens en découpant des mots de longueur fixe. Par exemple pour les mots de longueur $m = 6$ nous avons $N = 4^6 = 4096$ possibilités représentant tous les états possibles et donc tous les noeuds du système. On détermine les transitions entre ces mots en parcourant la chaîne d'une extrémité à l'autre. Un lien est ajouté entre le mot (le noeud) j vers le mot (le noeud) i si dans la séquence le mot j précède le mot i. Puisqu'il existe déjà un gap naturel, il n'est pas nécessaire d'introduire la matrice de téléportation et nous pouvons construire la matrice de Google G en suivant la procédure standard.

Il est intéressant d'étudier la statistique des éléments de la matrice G, les fig. A.4 et fig. A.5 montrent la distribution des éléments ainsi que la distribution des sommes en lignes des éléments de matrice. Dans le premier cas on observe un comportement universel en loi de puissance pour toutes les espèces, malgré les petites déviations visibles, l'exposant de décroissance $\nu \approx 2.5$ est très similaire à la décroissance de la distribution des liens sortants des réseaux du type pages web. En revanche dans le second cas, même si le comportement est à nouveau globalement similaire, il y a une décroissance plus rapide conduisant à un exposant plus élevé $\mu \approx 5$ ce qui est bien plus élevé que dans les réseaux du type pages web. Il y a donc une différence structurelle fondamentale entre le réseau des séquences d'ADN et celui du WWW. Cet exposant élevé est également à l'origine de la décroissance lente du PageRank pour toutes les espèces. La longueur des mots n'affecte pas le comportement global des courbes, la statistique est relativement stable.

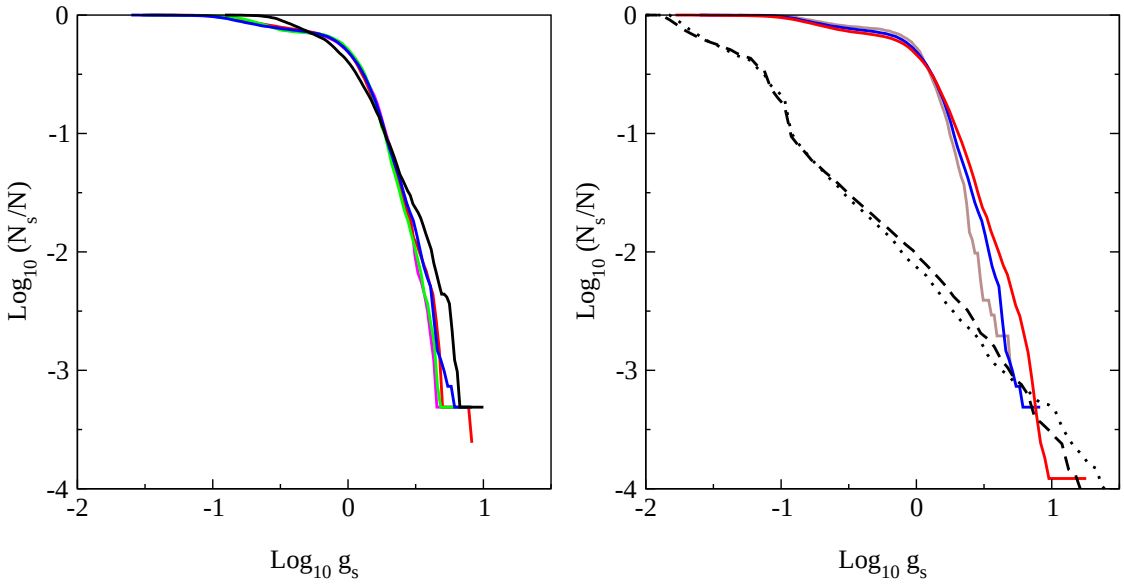

Figure A.5: Fraction intégrée N_s/N des sommes des éléments de matrices entrants avec $\sum_j G_{ij} \geq g_s$. Les panels gauche et droite montrent les mêmes cas que dans la fig. A.4 avec les mêmes couleurs. Les courbes traitillée et pointillée sont déplacées le long de l'axe x d'une unité vers la gauche par commodité.

Les spectres de différentes espèces sont montrés dans la fig. A.6, on constate qu'il existe bien un gap naturel dans tous les cas. La forme du nuage de valeurs propres donne une indication de la structure du réseau et donc des séquences d'ADN. Le poisson zèbre montre une structure qui ressemble à des connexion aléatoires entre les mots car le nuage est condensé mis à part quelques valeurs propres réelles de grande amplitude. L'augmentation de la taille des lettres conduit à un élargissement du nuage des valeurs propres, cela est dû au fait que lorsque le nombre de noeuds augmente il y a moins de chance que chaque connexion possible soit réalisée laissant ainsi paraître plus de structure car la longueur totale de la séquence ne varie pas. Nous avons également essayé d'obtenir un spectre similaire dans le cadre des modèles de matrices aléatoires, pour cela nous avons construit des matrices avec la même distribution d'éléments dont les connexions sont aléatoires. Nous n'avons clairement pas pu reproduire le spectre, cela signifie que seul la distribution statistique des éléments de matrice n'est pas responsable de la forme du nuage de valeurs propres et qu'il faut explorer l'organisation structurelle de la chaîne pour reproduire cette caractéristique.

De manière générale, les mots en tête de liste sortie par le PageRank sont les mêmes pour les espèces mammifères (les lettres A et T répétées) et différents pour le poisson zèbre, en accord avec les fréquences d'occurrences de ces mots. Pour pouvoir comparer la similarité sur une échelle globale, nous pouvons dessiner un diagramme de corrélation $K - K$. Pour comparer deux espèces, on note pour chaque mot les coordonnées formées par les indices de rang donnés par les deux PageRank. Chaque mot ayant deux indices, on obtient un graphe de N points dispersés indiquant

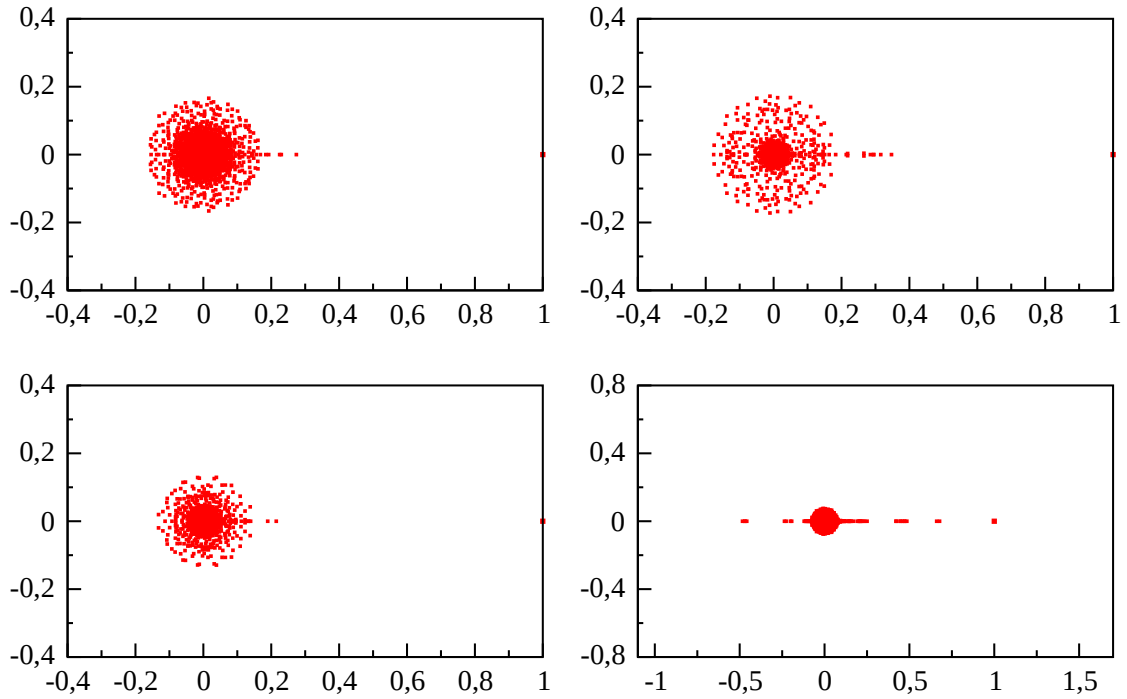

Figure A.6: Spectres de valeurs propres dans le plan complexe λ des matrices de Google de séquences d'ADN : taureau (haut gauche), chien (haut droite), éléphant (bas gauche), poisson zèbre (bas droite). Ici $m = 6$.

la similarité des deux séquences sous le point de vue des réseaux dirigés. Si les deux séquences sont strictement identiques, les points tomberont sur la diagonale, plus la dispersion autour de $y = x$ est grande plus les séquences diffèrent statistiquement l'une de l'autre. La figure fig. A.7 montre les comparaisons entre chacune des espèces animales considérées avec l'homme.

Pour caractériser quantitativement la dispersion, nous pouvons utiliser une mesure empirique $\sigma(s_1, s_2)$ basée sur la différence des rangs des mots dans les séquences des espèces s_1 et s_2 :

$$\sigma(s_1, s_2) = \sqrt{\sum_{i=1}^{N} (K_{s_1}(i) - K_{s_2}(i))^2 / N} \tag{A.4}$$

cette mesure peut en outre être rendue indépendante de la taille du système N. Les valeurs numériques indiquent que les séquences les plus proches sont celles de l'homme et du chien. L'observation qualitative des spectres indiquait déjà une grande ressemblance entre ces deux espèces. Nous avons également comparé les séquences correspondant à deux individus humains différents et la valeur de la dispersion est un ordre de grandeur plus petit que ceux obtenu par comparaison avec les autres espèces.

Nous avons exploré la possibilité d'analyser les séquences d'acides aminées de la même manière. Les triplets de bases de l'ADN correspondant à un acide aminé particulier il est possible d'extraire une séquence codante et de la traduire en une chaîne dont les lettres sont au nombre de 20. L'intérêt est d'analyser la partie codante de l'ADN et d'en étudier les similarités en utilisant le PageRank. Cette méthode a été appliquée aux données de 47 archées pour lesquels nous avons calculé la distribution du PageRank. En principe, la mesure de similarité introduite précédemment respecte les conditions nécessaires pour définir une matrice des distances entre les séquences symboliques analysées. Cette matrice des distances pourrait être traitée par un algorithme afin de produire un arbre phylogénétique donnant ainsi une nouvelle perspective sur la possibilité d'analyser l'évolution des organismes via le PageRank.

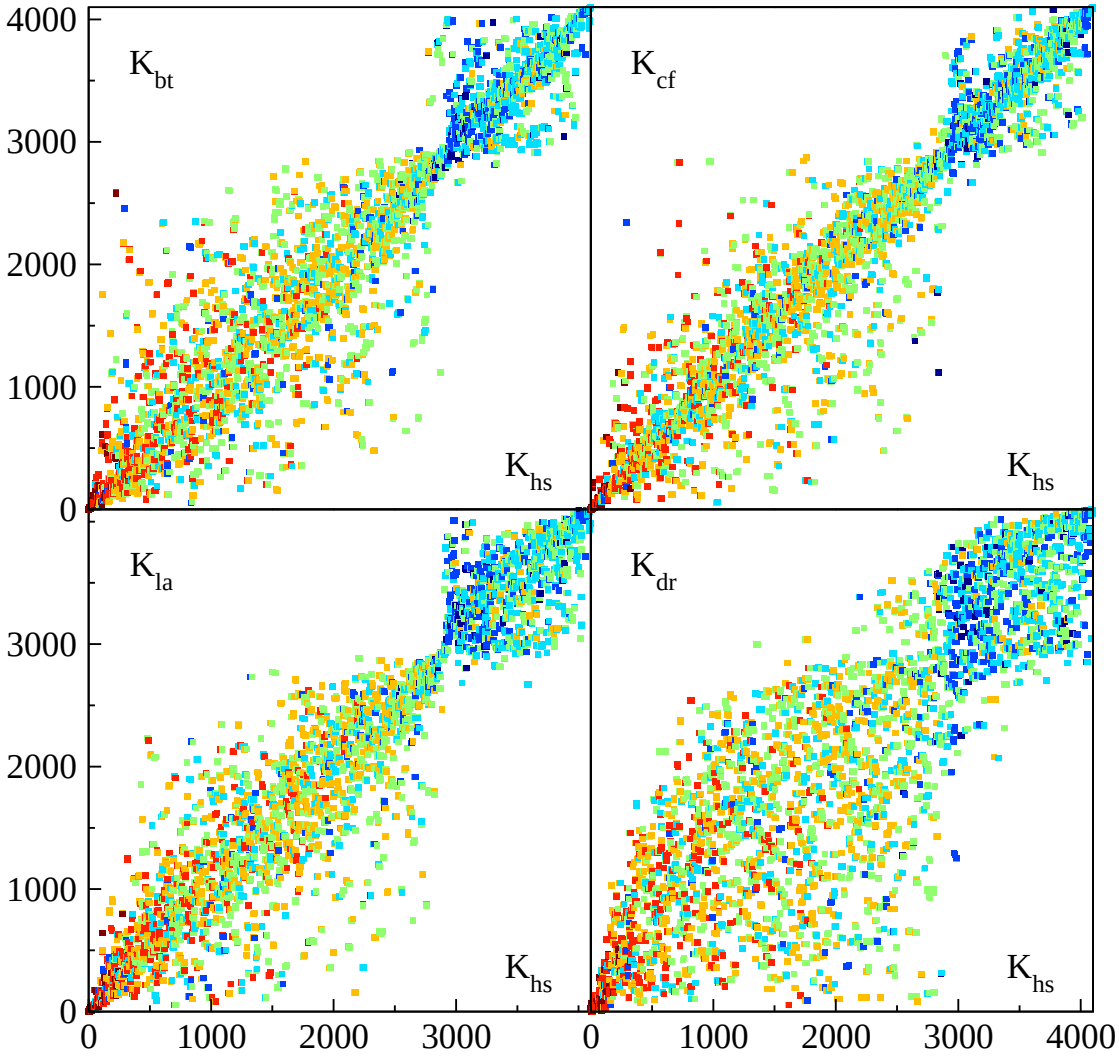

Figure A.7: Diagrammes de proximité PageRank dans le plan $K - K$ pour différentes espèces comparées à *Homo Sapiens* : L'axe x montre l'index du PageRank $K_{hs}(i)$ d'un mot i et l'axe y l'index du même mot i avec $K_{bt}(i)$ (taureau), $K_{cf}(i)$ (chien), $K_{la}(i)$ (éléphant) et $K_{dr}(i)$ (poisson zèbre). Ici $m = 6$ et les couleurs montrent le contenu en A ou T dans un mot i. Les couleurs varient de rouge pour le contenu maximal, brun, jaune, bleu clair et bleu pour les mots qui ne contiennent pas ces lettres.

Résumé : Le réseau des neurones de *C.elegans*

Un des plus grands challenges de la biologie moderne est de comprendre le fonctionnement du cerveau humain et l'émergence de la pensée consciente. Au niveau moléculaire, les recherches avancent rapidement et il est aujourd'hui possible d'imiter le comportement d'un neurone individuel avec une précision telle qu'il devient difficile de distinguer un signal expérimental d'un signal artificiel. Cependant la connaissance des processus chimiques au niveau d'un seul neurone ne suffit pas à comprendre le comportement du cerveau dans son ensemble. Récemment l'accent a été mis sur le comportement collectif d'un grand ensemble de neurones individuels en interaction dynamique pour tenter d'expliquer l'origine de la pensée consciente. L'objectif est donc d'étudier les propriétés d'un immense réseau de neurones afin de mieux comprendre ce que la structure des connections peut apporter au fonctionnement du cerveau. Cependant il est très difficile d'obtenir la carte complète de la connectivité neuronale chez l'homme, le seul organisme suffisamment simple pour lequel nous possédons une connaissance complète des neurones et de leurs connexions est le (très étudié) petit vers *C.elegans*. Nous allons appliquer la méthode de Google au système nerveux principal de ce vers composé de $N = 279$ neurones.

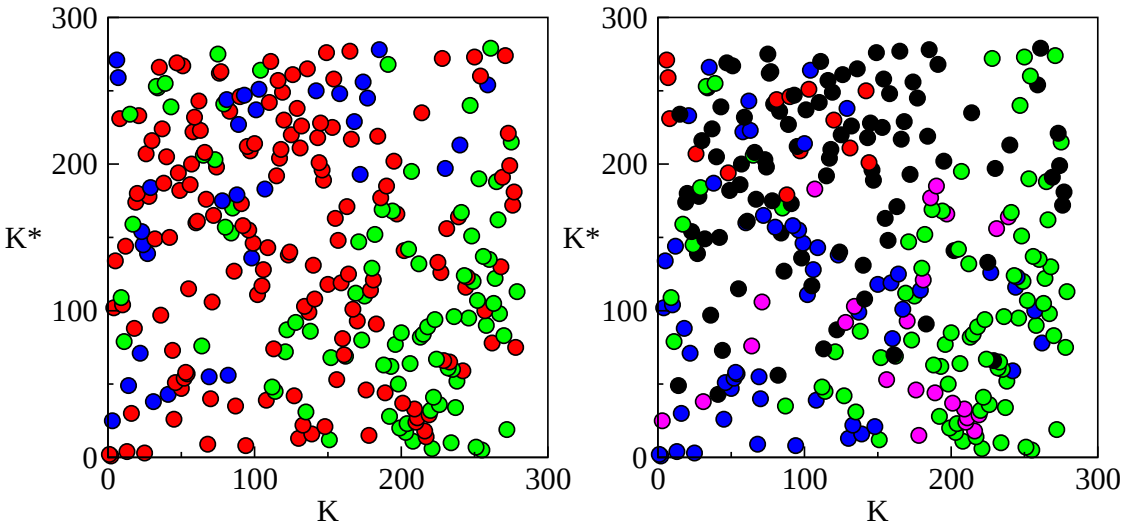

Figure A.8: Plan PageRank-CheiRank (K, K^*) montrant la distribution des neurones selon leur rang. *Panel gauche :* Coloration par région du soma - tête (rouge), milieu (vert), queue (bleu). *Panel droite :* Coloration par type de neurone - senseur (rouge), moteur (vert), interneurone (bleu), polymodal (violet) et non spécifié (noir).

Les neurones peuvent être connectés de deux manières : par des liaisons membranaires avec des cellules adjacentes ou par liaisons synaptiques. Dans le premier cas il n'y a pas de directionnalité et la matrice adjacente associée S_{gap} est symétrique, en revanche dans le second cas l'axone d'un neurone conduit l'influx nerveux et la transmet aux dendrites du neurone sur lequel il pointe donnant ainsi une matrice S_{syn} asymétrique. La matrice de Google est construire avec $S_{gap} + S_{syn}$. Ici nous introduisons le concept du réseau inversé, en effet pour chaque réseau dirigé il existe un réseau complémentaire que l'on peut obtenir en inversant le sens de chaque lien dirigé. Cela revient à transposer la matrice S avant la procédure de normalisation des colonnes. La nouvelle matrice G^* ainsi obtenue est également une matrice de Google et l'on peut calculer son vecteur propre principal nommé CheiRank pour éviter la confusion avec le PageRank, vecteur propre principal du réseau original. Le PageRank P, étant proportionnel aux liens entrants, nous renseigne sur les noeuds importants du réseau dans le sens autoritaires et influents. Le CheiRank P^* apporte une information complémentaire en mettant en évidence des noeuds plus communicatifs. Déterminer les deux rankings permet d'obtenir une meilleure classification des noeuds en 2 dimensions et permet de mieux comprendre l'organisation du réseau en étudiant les corrélations entre les deux

distributions de probabilités. La fig. A.8 montre le diagramme des corrélations où chaque point correspond à un neurone. L'image montre une dispersion des points dans tout le plan indiquant que le PageRank et le CheiRank ne sont pas ou que très peu corrélés. Cela est typique d'un réseau de type "fonctionnel" et se retrouve dans des réseaux tels que les procédures d'appel de fonctions du coeur de linux ou encore du commerce international en un produit particulier. L'existence d'une corrélation se serait manifestée par un arrangement des points le long de la diagonale. Pour déterminer quantitativement la corrélation, le corrélateur κ a été défini comme :

$$\kappa = N \sum_i P(i)P^*(i) - 1 \tag{A.5}$$

et ici nous avons $\kappa \approx 0$. La coloration en fonction du type et de la position des neurones permet de mieux voir dans le plan $K - K^*$ la répartition des noeuds importants en PageRank et en CheiRank. On observe que les noeuds au top du Cheirank sont principalement des neurones moteurs et se trouvent localisés dans la tête du vers. De même les neurones situés au milieu du vers se retrouvent bas dans le classement de PageRank mais haut dans le CheiRank. On retrouve une partie des résultats établis de l'importance des noeuds appartenant au *rich club*.

	PR	CR	2DR	EOPR	EOCR	IMPR	IMCR
1	AVAR	AVAL	AVAL	PHAL	AS07	RMGL	RMGL
2	AVAL	AVAR	AVAR	PHAR	VA10	URXL	AVAL
3	PVCR	AVBR	AVBL	VC04	AS08	ADEL	ASHL
4	RIH	AVBL	AVBR	FLPL	AS10	AIAL	AVBR
5	AIAL	DD02	PVCR	ASKL	DB06	IL2L	URXL
6	PHAL	VD02	AVKL	ASKR	DB05	ADLL	AVEL
7	PHAR	DD01	PVCL	AVFL	AS01	PVQL	RIBL
8	ADEL	RIBL	PVPR	AVG	VA02	ALML	RMDR
9	HSNR	RIBR	RIGL	PVPL	DA07	ASKL	RMDL
10	RMGR	VD04	PVPL	RIFR	VA03	CEPDL	RMDVL
11	VC03	VD03	RIS	PQR	VD03	ASHL	AVAR
12	AIAR	VD01	AVDR	VC05	AS09	AWBL	SIBVR
13	AVBL	AVER	RIGR	AVJL	VA06	SAADR	AIBR
14	PVPL	RMEV	AVDL	PVQR	VA03	RMHR	ADAL
15	AVM	RMDVR	AVKR	RIFL	VD02	RMHL	RMHL
16	AVKL	AVEL	RIBR	ASHR	DA06	RIH	AVBL
17	HSNL	VD05	DVC	VD13	VA05	OLQVL	SIBVL
18	RMGL	SMDDR	AIBL	AIMR	AS04	AIML	ASKL
19	AVHR	DD03	DVA	AVHR	AS06	HSNL	RID
20	AVFL	VA02	AVJL	PVPR	DD01	SDQR	SMBVL

Figure A.9: Top 20 des neurones du PageRank (PR), CheiRank (CR), 2D Rank (2DR), PageRank et CheiRank d'opportunité égale (EOPR et EOCR) ainsi que l'impactRank de G (IMPR) et de G^* (IMCR) de l'état initial RMGL avec $\gamma = 0.7$. La coloration indique le type des neurones : interneurones (bleu), neurones moteurs (vert), neurones senseurs (rouge) et neurones polymodaux (violet).

la fig. A.9 présente les top neurones mis en évidence par plusieurs ordres différents. Mis à part le PageRank et CheiRank, il es possible de les combiner en un rank 2D où l'on remarque que les interneurones sont sélectionnés. On peut également définir les PageRank et CheiRank d'opportunité égale qui consiste a renormaliser les éléments par le degré des noeuds.

Résumé : Le jeu de Go d'un point de vue des réseaux complexes

Dans cette partie nous allons appliquer les méthodes de la matrice de Google à un antique jeu de stratégie sur plateau. Le jeu de Go est né en Chine il y a plusieurs milliers d'années et s'est progressivement étendu dans toute l'Asie puis plus récemment au monde entier lorsque les premiers joueurs occidentaux ont atteint un niveau professionnel. Le jeu de go se joue à deux, chaque joueur possède un ensemble de pierres blanches pour l'un et noires pour l'autre. Le but du jeu est de poser une pierre sur l'une des 19×19 intersections du plateau, à tour de rôle, de sorte à pouvoir délimiter le plus de territoire possible. Lors de la partie, un joueur peut capturer les pierres ennemies si ces dernières sont entourées et qu'il ne leur reste plus aucune liberté. On enlève alors les pierres du jeu et on compte le territoire gagné dans le score de l'attaquant. A la fin de la partie le joueur possédant le plus de territoire gagne la partie.

Ce jeu est l'un des derniers qui résiste encore à l'intelligence artificielle, en effet les progrès en informatique ont permis de résoudre un bon nombre de jeux de stratégie jusqu'à pouvoir élaborer un programme informatique capable de battre le meilleur des joueurs humains. Contrairement aux autres jeux, le très grand nombre de configurations (plusieurs ordres de grandeurs supérieur aux échecs par exemple) dans le jeu de go est une des raisons principales de la difficulté à créer un tel programme. L'autre raison en est qu'il est très difficile pour un ordinateur d'estimer la pertinence d'un coup afin de décider quel serait le meilleur coup à jouer pour une situation donnée. Malgré l'immobilité des pierres, les combinaisons tactiques sont immenses et complexes et les meilleurs programmes existants aujourd'hui ne sont pas capable de battre un joueur professionnel de haut niveau. La meilleure approche jusqu'à présent consiste à explorer les arbres des coups en jouant un grand nombre de fois et aléatoirement jusqu'à la fin de la partie pour estimer la valeur d'un coup donné. Quelques améliorations supplémentaires tel que l'exploration de coups rares peuvent augmenter un peu l'efficacité du programme mais il est nécessaire d'approcher le problème d'un autre angle pour pouvoir révolutionner d'avantage leur efficacité.

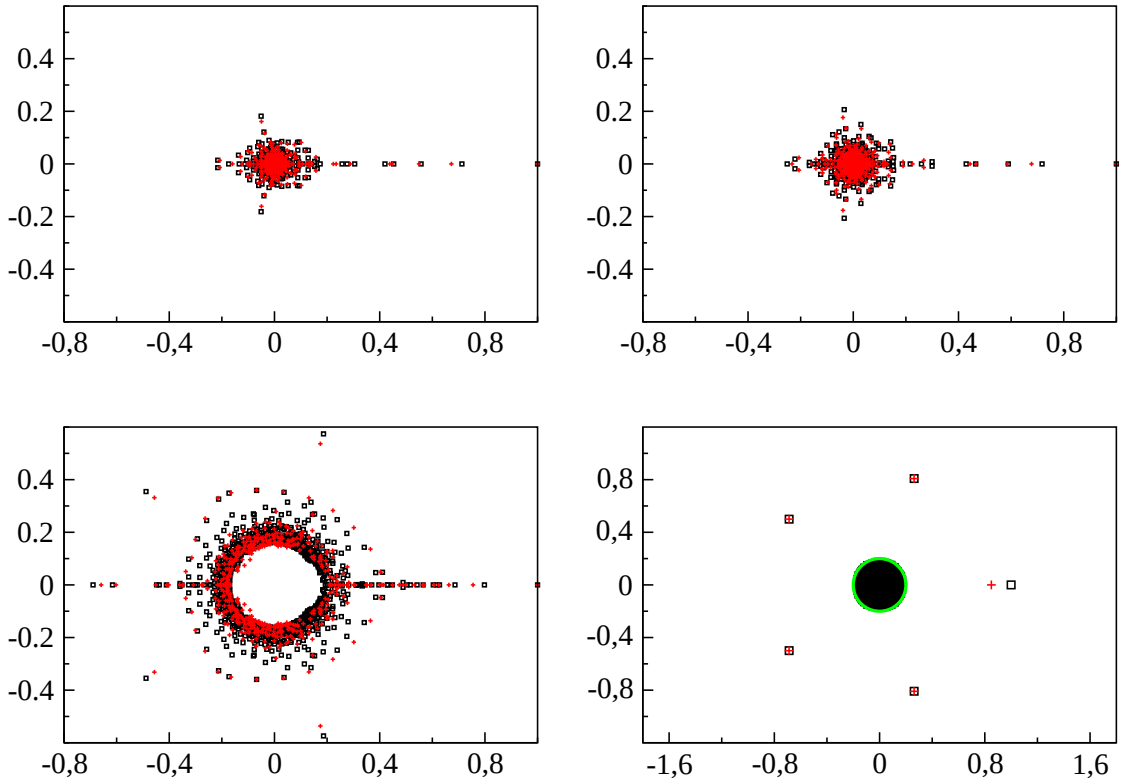

Figure A.10: Spectre de G dans le plan complexe (carrés noires) et G^* (croix rouges) pour les trois réseaux différents : I (haut gauche), II (haut droite) et III (bas).

Nous allons tenter d'apporter une nouvelle approche en appliquant la méthode de la matrice de Google à l'énorme base de donnée existante des parties jouées. L'espoir étant de tirer partie des communautés de coups que la méthode permet de mettre en évidence grâce à l'approche des réseaux dirigés et de pouvoir améliorer l'estimation des valeurs de ces coups au sein de l'algorithme Monte Carlo go afin d'augmenter significativement l'efficacité des programmes. Dans ce travail nous allons nous limiter à définir un réseau dirigé de manière pertinente et montrer qu'il existe des informations dans les vecteurs de ranking et les vecteurs propres suivants qui pourraient servir à regrouper certains coups similaires. Pour ce faire, nous avons écrit un code qui joue une partie entière à partir d'un fichier .sgf répertoriant la partie entière entre deux joueurs. Durant la partie, nous détectons la configuration environnante à l'intersection où la pierre va être posée. Nous définissons cet environnement comme étant une plaquette et nous identifions toutes les symétries de rotations et miroirs afin d'en retenir l'essence de la configuration. L'existence des pierres de handicap permet aux joueurs de niveaux très différents de pouvoir s'affronter sur un pied d'égalité, en effet les pierres déjà posées sur le plateau représentent un immense avantage tactique pour le joueur de plus faible niveau. Ainsi la notion de gagnant pour le jeu de go est ambigu et il n'est pas forcément intéressant de séparer les deux joueurs. Nous construisons donc le réseau d'une partie en considérant tous les coups joués par les deux joueurs et pour cela nous symétrisons également les plaquettes par rapport à l'échange des pierres blanches et noires. Les noeuds du réseau sont donc toutes les configurations environnantes possibles pour une forme de plaquette donnée. Ici nous considérons trois réseaux différents, le premier et le plus simple était le réseaux des plaquettes carrées, à savoir les huit voisins entourant la pièce centrale. Le second réseau est de même forme mais on distingue le statut des premiers voisins directes afin de distinguer les pierres en atari (les pierres appartenant à une chaîne qui ne possède plus qu'une seule liberté). Il n'est pas possible de considérer des carrés de plus grande taille compte tenu du nombre de possibilités en revanche on peut considérer les plaquettes en diamant, constitué du carré et des seconds voisins directes. Nous obtenons respectivement $N = 1107$, $N = 2051$ et $N = 193995$ plaquettes uniques après symétrisation et donc ce sont le nombre de noeuds des réseaux considérés. Pour définir les liens, nous suivons le déroulement de la partie et posons un lien du coup j vers le coup i lorsque le coup i suit le coup j dans une zone de taille $d = 4$. La définition de cette zone permet de distinguer les coups tactiques joués ensembles dans une certaine zone de ceux qui sont subitement joué dans un autre coin du plateau car il pourrait y avoir plusieurs combats se déroulant simultanément dans plusieurs zones du plateau. Les coups passant d'une zone à l'autre ne doivent donc pas être considérés comme reliés.

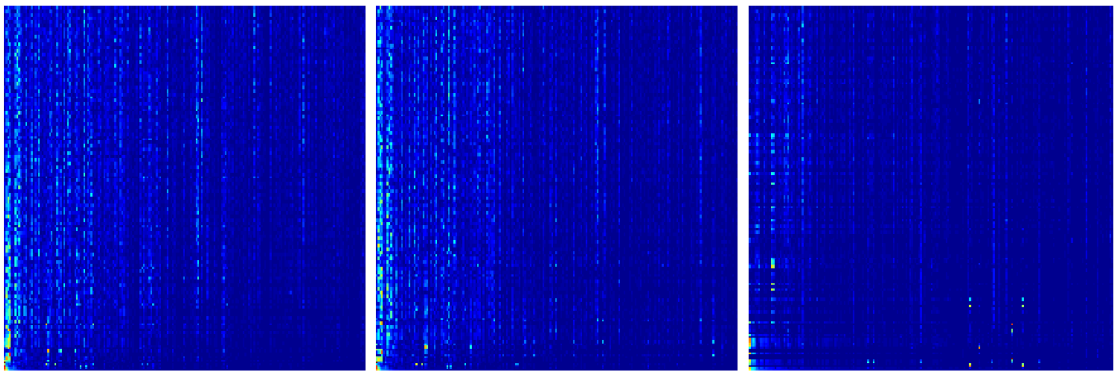

Figure A.11: Figure de corrélations des vecteurs propres de la matrice G pour les trois réseaux différents : I (gauche), II (milieu) et III (droite). Les top 200 vecteurs propres dans l'ordre décroissant des modules de valeurs propres sont montrés horizontalement de bas en haut. Seulement les 200 premiers éléments sont affichés dans la base du PageRank. Les couleurs sont proportionnels au module des composants (la normalisation d'un état propre ψ est $\sum_i |\psi|^2 = 1$), allant du bleu (minimum) au rouge (maximum).

Pour les deux premiers réseaux il est possible de diagonaliser la matrice de Google car le système est suffisamment petit. En revanche pour le réseau en diamant il faut trouver une autre méthode si l'on veut étudier plusieurs vecteurs propres. La méthode d'Arnoldi permet de résoudre ce problème. Cette méthode est basée sur le calcule des sous-espaces de Krylov et une procédure d'orthonormalisation, elle permet d'obtenir avec une grande précision quelques valeurs propres de grands modules et les vecteurs propres associés de très larges matrices asymétriques. Nous avons pu calculer quelques centaines de vecteurs propres des matrices de G et G^* de cette manière, construites grâce à une base de donnée d'environ 135000 parties enregistrées.

La fig. A.10 montre les spectres des valeurs propres des trois réseaux considérés pour G et le réseau inversé G^*. Malgré l'information supplémentaire apportée par le statut d'atari, les coups ne sont pas suffisamment désambiguïsés. Nous voyons que pour le cas du diamant on observe plus aisément le nuage de valeurs propres qui prend une forme intéressante indiquant la présence de communautés de coups spécifiques tant pour G que G^*. Les vecteurs PageRank et CheiRank montrent des lois de puissance dans la décroissance de leur distribution indiquant qu'ils mettent en évidence aisément les coups importants. Ces coups ressemblent aux coups les plus fréquents mais ne sont pas strictement identiques, cela signifie que les vecteurs de ranking apportent un petit plus qui diffère du simple comptage de l'occurrence de coups. En fait le PageRank met en évidence les coups qui suivent beaucoup de coups différents et le CheiRank montre les coups qui peuvent ouvrir la voie à beaucoup d'autres coups. Une certaine symétrie existe puisque en moyenne un coup possède un prédécesseur et un suivant mais ce n'est qu'au niveau statistique. Une différence notable existe lorsqu'on observe les plaquettes. Il est également intéressant de noter les coups sur lesquelles se localisent les vecteurs propres suivants.

La fig. A.11 montre qu'il existe des zones qui concentrent la probabilité des vecteurs propres même éloigné dans la base du PageRank. Ainsi les vecteurs propres ne sont pas forcément des petites variations du PageRank mais contiennent véritablement des groupes de coups similaires qui ressortent de manière significative. Des exemples de coups topologiquement similaires mis en évidence par les vecteurs propres de G sont montrés dans la fig. A.12 où l'on remarque qu'il peut exister certains mélanges entre les coups fréquents (ou les coups du PageRank) avec ceux spécifiquement soulignés par le vecteur propre. Il est donc utile de considérer des méthodes d'extraction de communautés. Dans notre travail nous avons procédé en filtrant les coups du PageRank une première fois puis en essayant de regrouper les coups par ancêtres communs. En effet la méthode de Google étant basée sur les liens dirigés, choisir les coups partageant un parent commun duquel ils proviennent peut être une façon logique de déterminer une communauté. La définition de communauté est par définition subjective et elle l'est d'autant plus dans le contexte du jeu de go qu'il est difficile de déterminer précisément la nature de ce qui regroupe ces coups. Ainsi il est utile d'explorer plusieurs voies possibles comme par exemple de considérer des mélanges des coups qui apparaissent dans plusieurs vecteurs propres. D'autres définitions de réseaux peuvent également aider à la compréhension du jeu et peuvent apporter des informations utiles comme par exemple la division des phases de jeu en trois parties : début, milieu et fin de partie. Une autre possibilité pourrait être de séparer les joueurs en fonction de leur niveau de jeu et observer les différences afin de mieux comprendre ce qui donne l'avantage à un joueur de haut niveau. Plusieurs pistes restent à explorer et l'espoir est de pouvoir utiliser ce type de technique pour mieux cerner et évaluer la pertinence d'un coup lors d'une partie.

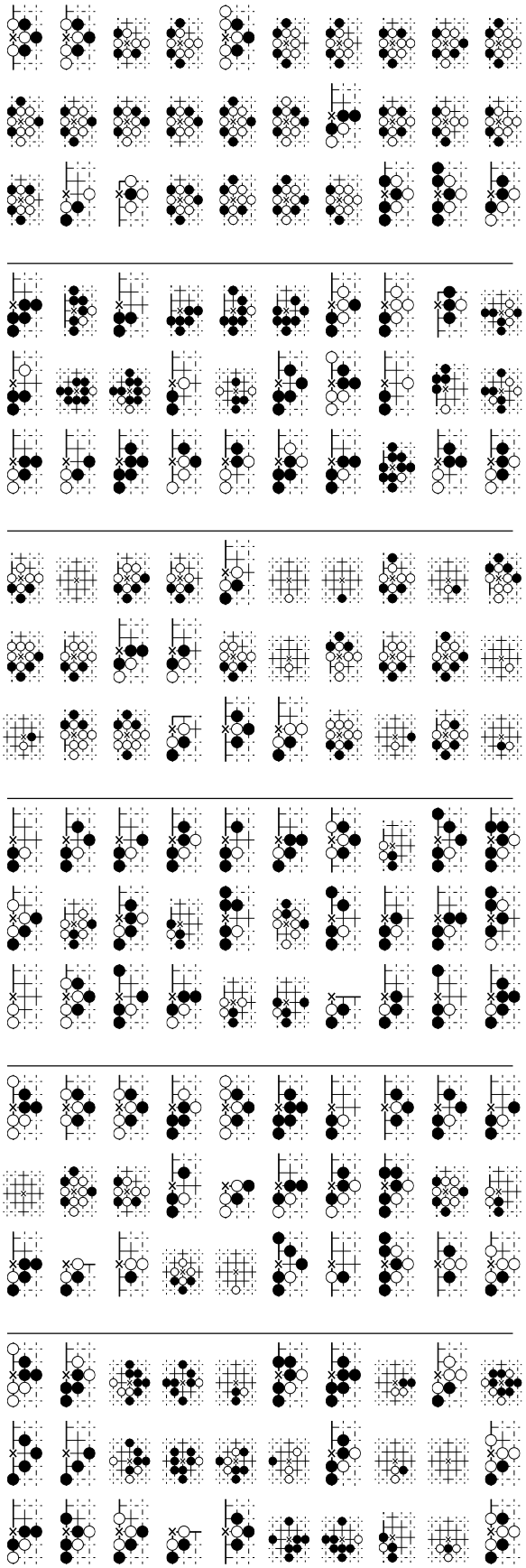

Figure A.12: Exemples des top 30 noeuds où se localisent les vecteurs propres de G pour le réseau en diamant. *De haut en bas:* $\lambda_7 = -0.6158$, $\lambda_{11} = 0.1865 - 0.5739i$, $\lambda_{13} = 0.5651$, $\lambda_{21} = -0.4380$, $\lambda_{22} = 0.4294 + 0.0006481i$ et $\lambda_{32} = 0.3847 + 0.04677i$.

Résumé : L'utilisation du PageRank dans le modèle de formation d'opinion

Dans cette partie nous allons sortir un peu du contexte de la matrice de Google pour voir comment le PageRank pourrait être utilisé dans un autre contexte : l'étude de la formation d'opinion. La sociophysique est un domaine qui s'intéresse à l'étude des comportements humains par les moyens provenant de la physique statistique. Il est vrai que la complexité d'un individu ne permet pas son approximation grossière et que la sociologie ne pourrait être remplacée par le sciences dures, cependant il a été montré que dans un large groupe chacun des individus se comporte de manière radicalement différente et l'ensemble du groupe peut adopter des comportements prédictibles et simples. L'intérêt de la sociophysique est d'étudier avant tout les phénomènes collectifs qui émergent dans des larges groupes d'humains. La littérature et les modèles de sociophysique abondent et traitent en détails les phénomènes de vote et de propagation d'opinions, cependant dans la plupart des cas les individus sont considérés comme des agents placés sur des grilles régulières. Ici nous proposons d'incorporer deux dimensions plus réalistes grâce au PageRank. Premièrement, nous utilisons la structure d'un réseau dirigé réel (Cambridge et Oxford) afin de simuler le véritable réseau de connaissances qui est en réalité un réseau sans échelle. Deuxièmement, on suppose qu'on préfère écouter ou suivre les conseils des gens de notre entourage qui ont le mieux réussi, le PageRank permet de fournir cet ordre en listant les amis dans l'ordre d'importance global qu'ils ont.

Nous proposons deux modèles, le premier désigné par PROF est un modèle d'opinion binaire où chaque noeud prend une opinion basée sur la valeur de Σ_i dépendant des opinions et des valeurs de ses voisins. Ainsi la formule suivante décrit la condition de choix de l'opinion :

$$\Sigma_i = a \sum_j P^+_{j,in} + b \sum_j P^+_{j,out} - a \sum_j P^-_{j,in} - b \sum_j P^-_{j,out} \tag{A.6}$$

l'implémentation est directe : on utilise un réseau (ici Cambridge et Oxford) puis on choisit une distribution aléatoire de noeuds rouges et de noeuds bleus (les deux opinions). On calcule ensuite la condition pour chaque noeud et on laisse le système se relaxer puis on détermine la fraction finale de noeuds rouges. En répétant ce procédé un grand nombre de fois nous obtenons le diagramme de densité montré en fig. A.13 où on montre également l'effet du paramètre a. On remarque qu'il existe une région de bistabilité plus forte pour les valeurs de a faibles ou la population suit l'opinion de l'élite.

Une approche complémentaire et importante est donné par le modèle de Sznajd qui considère l'influence d'un groupe sur ses voisins. Encore une fois nous allons tirer partie de la structure de réseau et du PageRank. Pour l'implémentation nous définissons un groupe et convertissons les voisins directes qui pointent vers ce groupe. La conversion se passe si le PageRank de l'individu est plus faible que celui du groupe qui n'est autre que la somme des PageRank des éléments qui constituent le groupe. Les figures de densité dans fig. A.14 sont construites de la même façon que ceux du modèle PROF sauf que l'on attend suffisamment longtemps parce que dans ce cas il y a une compétition induite entre les groupes de noeuds qui résulte en un état stationnaire mais non figé. On constate dans ce cas que même une fraction initiale petite de noeuds rouges résiste tout de même à la pression de la société indiquant qu'en groupe on survit mieux au totalitarisme. La taille du groupe n'affecte que peu le phénomène, on voit déjà à partir de $N_g = 3$ l'effet en question et augmenter la taille du groupe résulte en une stabilisation des fluctuations. Il est vrai que cette approche est limitée car une généralisation directe à plus de deux opinions est difficile. Cependant nous avons observé l'essence de la formation d'opinion et de sa propagation selon deux approches complémentaires.

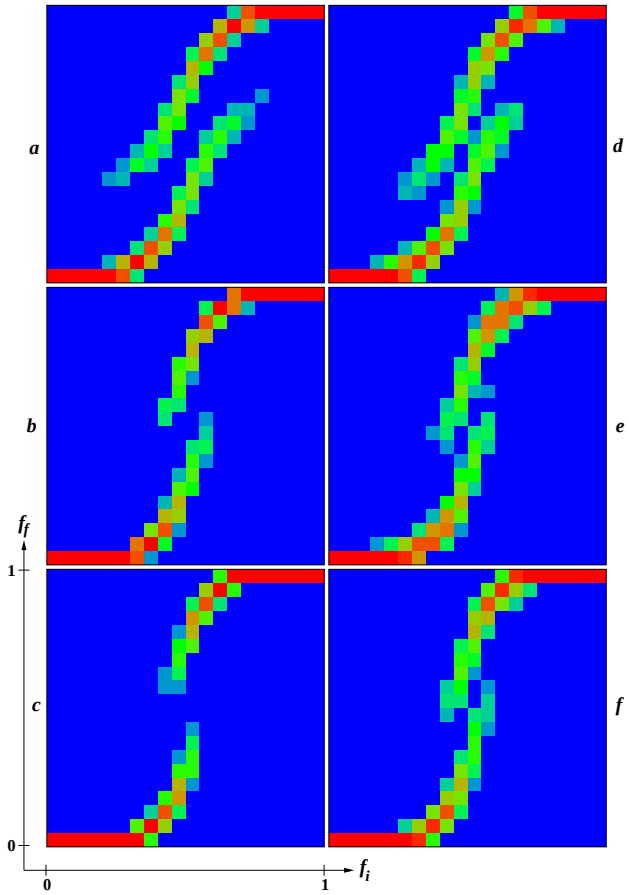

Figure A.13: Densité de probabilité W_f de trouver une fraction finale f_f, montré sur l' axe y, dépendant d'une fraction initiale de rouge f_i, montré sur l'axe x; les données sont montrées dans un carré unité $0 \leq f_i, f_f \leq 1$. Les valeurs de W_f sont définies comme un nombre relatif de réalisations trouvés dans chaque cellule 20×20 qui couvre le carré unité. Ici $N_r = 10^4$ réalisations de distribution aléatoire de couleurs sont utilisés pour obtenir les valeurs de W_f. Pour chaque réalisation, l'évolution dans le temps est suivi jusqu'au point de convergence de $t = 20$ itérations; Cambridge (colonne de gauche) et Oxford (colonne de droite) et $a = 0.1$ pour (a,d), $a = 0.5$ pour (b,e) et $a = 0.9$ pour (c,f). La probabilité W_f est proportionnelle à la variation de couleur allant de zéro (bleu) à un (rouge).

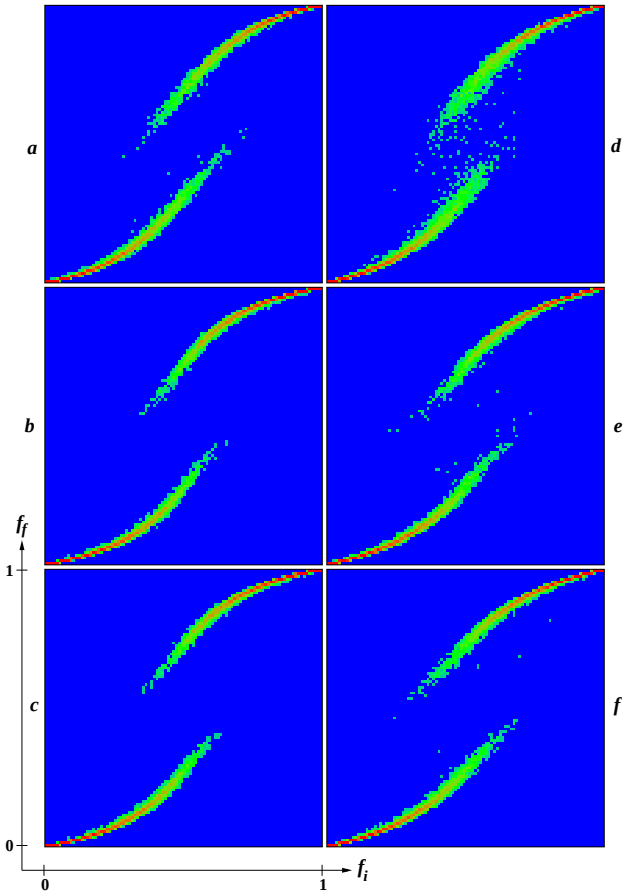

Figure A.14: Densité de probabilité W_f de trouver une fraction finale f_f, montré sur l' axe y, dépendant d'une fraction initiale de rouge f_i, montré sur l'axe x; les données sont montrées dans un carré unité $0 \leq f_i, f_f \leq 1$. Les valeurs de W_f sont définies comme un nombre relatif de réalisations trouvés dans chaque cellule 100×100 qui couvre le carré unité. Ici $N_r = 10^4$ réalisations de distribution aléatoire de couleurs sont utilisés pour obtenir les valeurs de W_f. Pour chaque réalisation, l'évolution dans le temps est suivi jusqu'au point de convergence de $\tau = 10^7$ itérations; Cambridge (colonne de gauche) et Oxford (colonne de droite) et $N_g = 3$ pour (a,d), $N_g = 8$ pour (b,e) et $N_g = 13$ pour (c,f). La probabilité W_f est proportionnelle à la variation de couleur allant de zéro (bleu) à un (rouge).

Résumé : Conclusion et Perspective

Nous avons vu dans ce travail qu'il y a une large variété de systèmes qui peuvent être étudiés grâce à l'approche des réseaux dirigés pourvu que l'on définisse correctement les noeuds et les liens. On peut obtenir des informations utiles concernant l'organisation structurel des noeuds aidant à leur tour à comprendre le système étudié. Par conséquent, durant ces dernières décennies, l'approche des réseaux complexes a connu un très grand essor et le but de cette présente thèse était de montrer que l'on peut appliquer les outils de la méthode de la matrice Google sur de tels réseaux complexes dirigés afin de caractériser sa structure de manière simple et efficace.

Il a déjà été montré que le vecteur propre principal de la matrice Google (le PageRank) était très efficace sur les réseaux larges et sans échelles et dirigés tels que le World Wide Web. Ici nous avons montré qu'il peut être utilisé en complément avec le CheiRank pour caractériser des réseaux en deux dimensions. Nous avons également discuté en détails les spectres et comparé les réseaux nouvellement étudiés avec les cas connus des réseaux de pages web.

Ces études ne sont de loin pas complètes, en effet il y a plusieurs opportunités pour progresser dans le domaine. Mis à part les améliorations que l'on peut apporter spécifiquement à un sujet précis, il y a également la possibilité d'explorer des extensions génériques du contexte de la matrice Google. Par exemple une extension directe consisterait à introduire une distinction entre les types des liens : Supposons que nous ayons r types de liens dirigés dans un réseau de taille N, il est alors possible de construire une matrice de Google G de taille $rN \times rN$ construite grâce à la matrice de connectivité faite de $r \times r$ blocs de matrices $N \times N$ permettant ainsi de décrire chaque lien dirigé entre chaque paire de noeuds tout en différenciant les r types de liens. Cette méthode pourrait être une alternative à l'approche des réseaux multiplexes.

Une autre extension à ne pas négliger serait la modulation permise par la matrice de téléportation $\mathbf{e}\mathbf{v}^T$ et comme mentionné précédemment ce travail a été effectué avec $\mathbf{v} = \mathbf{e}$. Choisir une autre distribution de probabilité pour $\mathbf{v}$ pourrait mettre en lumière des effets intéressants causés par le biais dans la téléportation aléatoire. Le spectre des valeurs propres ne va pas changer mais la convergence vers un état stationnaire sera différente, le PageRank donnerait donc un résultat différent en incorporant le biais.

En guise de remarque finale à cette thèse, il ne faut pas oublier le point faible de cette méthode : l'étude de réseaux de taille fixe et statique. En réalité presque chaque réseau considéré évolue constamment et il est clair que la prochaine étape majeure est l'élaboration d'un cadre similaire applicable aux liens dynamiques ou aux réseaux avec un nombre de noeuds changeants.

Appendix B - Some Useful Mathematical Results

The mathematical framework behind the Markov chain theory and more generally the Perron-Frobenius theorem is extremely rich and vast. The literature on the subject abounds and one can easily find some complementary materials and detailed explanations on numerous aspects of these theories. Here we give a synthesis of the key points and a guiding line to prove the theorems without thoroughly discussing each and every step in detail. The readers interested in additional informations, or proof for properties that are not discussed here, are strongly encouraged to visit the references.

Before starting the discussions it is useful to clarify some concepts and notations :

- $\mathbf{A} \in \mathcal{R}^{n\times n}$ is a square matrix of size n with real entries.
- $\mathbf{A} > \mathbf{B}$ represents the element-wise inequality $a_{ij} > b_{ij} \quad \forall(i,j)$.
- $\sigma(\mathbf{A})$ is the set of eigenvalues $\{\lambda_1, ..., \lambda_n\}$ of $\mathbf{A}$ and $\rho(\mathbf{A})$ is its spectral radius.
- $\rho(\mathbf{A}) = \max_i(|\lambda_i|)$ is the magnitude of the largest eigenvalue of $\mathbf{A}$.
- The bar notation $|\mathbf{A}|$ (or for a vector $|\mathbf{x}|$) indicates the absolute value for each element $|a_{ij}|$ ($|x_j|$ for a vector).

The eigenvalue $\lambda = 1$ plays such a central role in our work that it is interesting to see where it comes from. This value stems from the probabilistic nature of the matrices, indeed the spectral radius $\rho(\mathbf{A}) = 1$ for a stochastic matrix $\mathbf{A}$. This can be understood thanks to the Gelfand's formula. The following developments are taken from the course [Williams, 2011].

Let's consider the one norm for a finite vector $||\mathbf{x}||_1 = \sum_i |x_i|$ and for a matrix $||\mathbf{A}||_1 = \max_{||\mathbf{x}||_1=1} ||\mathbf{A}\mathbf{x}||_1$. Then the spectral radius is given by :

$$\rho(\mathbf{A}) = \lim_{k\to\infty} ||\mathbf{A}^k||_1^{1/k}. \tag{B.1}$$

To use this, let's consider first the fact that the product of two stochastic matrices is a stochastic matrix. Indeed if $\mathbf{P}$ and $\mathbf{Q}$ are stochastic then $(\mathbf{PQ})_{il} = \sum_j p_{ij}q_{jl}$. Summing along the columns yields :

$$\sum_i (\mathbf{PQ})_{il} = \sum_i \sum_j p_{ij}q_{jl} = \sum_j \sum_i p_{ij}q_{jl} = \sum_j q_{jl} \sum_i p_{ij} = 1 \tag{B.2}$$

Therefore if $\mathbf{A}$ is stochastic then $\mathbf{A}^k$ is stochastic for any integer $k > 0$.

Let's also consider the fact that if $\mathbf{P}$ is stochastic then $||\mathbf{P}||_1 = \max_{||\mathbf{x}||_1=1} ||\mathbf{P}\mathbf{x}||_1 = 1$ because :

$$\sum_i (\mathbf{P}\mathbf{x})_i = \sum_i \sum_j p_{ij}x_j = \sum_j x_j \sum_i p_{ij} = \sum_j x_j \tag{B.3}$$

therefore we have that $||\mathbf{P}\mathbf{x}||_1 = ||\mathbf{x}||_1$ and $||\mathbf{P}||_1 = \max_{||\mathbf{x}||_1=1} ||\mathbf{P}\mathbf{x}||_1 = \max_{||\mathbf{x}||_1=1} ||\mathbf{x}||_1 = 1$. Finally these properties allow us to compute the spectral radius of a stochastic matrix $\mathbf{A}$ as $\rho(\mathbf{A}) = \lim_{k\to\infty} ||\mathbf{A}^k||_1^{1/k} = \lim_{k\to\infty} 1_1^{1/k} = 1$.

In addition to this property, we can discuss the existence of at least one eigenvalue $\lambda = 1$ for the stochastic matrices. It is related to the fact that finite state space time homogeneous Markov chains (which are modeled by stochastic matrices) have at least one stationary distribution. Indeed if $\mathbf{P}$ is stochastic then :

$$0 = \sum_i p_{ij} - \sum_i Id_{ij} = \sum_i (p_{ij} - Id_{ij}) \quad \forall j \tag{B.4}$$

meaning that the rows of $\mathbf{P} - \mathbf{Id}$ are linearly dependant and consequently $det(\mathbf{P} - \mathbf{Id}) = 0$. It is known that if λ is an eigenvalue of $\mathbf{P}$ then $det(\mathbf{P} - \lambda\mathbf{Id}) = 0$, therefore by identification we obtain that $\lambda = 1$ is an eigenvalue of $\mathbf{P}$.

Regarding the theorem of the existence and unicity of the stationary distribution in the Markov chain, we will present the guiding line for the proof based on the courses from [Srikant, 2009] and [Sigman, 2009]. The idea is to show that a finite state space forces the existence of at least one recurrent state which is shared by all the states due to the irreducibility. Then we must establish that there cannot be a null recurrent state so the chain is positive recurrent and we will write the evolution of the chain in terms of revisit cycles to establish that in the long run the chain will converge to a stationary state. The unicity fellows from the unique representation related to the expectation of the first return times and finally the aperiodicity makes this stationary solution to coincide with the iterative limiting distribution.

Theorem 1. *Every irreducible Markov chain with a finite state space is positive recurrent, thus having a unique stationary distribution π. And if the chain is aperiodic, π is the limiting distribution $\pi = \lim_{k\to\infty} P^{(k)}\mathbf{v}$ for any probability distribution $\mathbf{v}$.*

Proof. Let $\{X_n\}_{n\geq 0}$ be a Markov chain with finite number N of states. If all the N states were transient, by definition, in the long run they would be visited not at all or a finite number of times, which is impossible. There is therefore at least one recurrent state, let that state be i.

The irreducibility implies that $\exists t > 0$ such that $P(X_t = j|X_0 = i) > 0$ for any pair (i, j). This means that in terms of the graph point of view there is a path between any node towards any other node. Therefore there exist a path between the recurrent state i and any other state and similarly a path from any other state towards i exists and consequently other states are also recurrent. In fact irreducibility is a *class property* meaning that it is shared among all the states inside the same class. A class is defined here in the sense of equivalence relation, for example the relation *communicate* is given by : if there is a set of transitions from state a towards state b (written $a \to b$) and if a set of reverse transitions exists from b towards a (written $b \leftarrow a$) then the states a and b communicate and it is denoted by $a \leftrightarrow b$. The states that communicate with each other belongs to the same class. Since the state space is finite, the Markov chain has at least one class and since the irreducibility allows the communication of each pair of states, a finite irreducible Markov chain has a single class. Therefore if one state i is recurrent, then all the states are recurrent. Next we need to show that the chain is positive recurrent.

We have seen that if $\mathbf{P}$ is stochastic then $\mathbf{P}^m$ and $\sum_i p_{ij}^{(m)} = 1$ for any integer $m > 0$. Let's consider the stochastic matrix $\mathbf{P}$ associated to $\{X_n\}_{n\geq 0}$ and suppose that there is a state j that is null recurrent, then in the long run the fraction of time spent in that state would be :

$$\lim_{n\to\infty} \frac{1}{n} \sum_{m=1}^{n} \delta(X_m, j) = \lim_{n\to\infty} \frac{1}{n} \sum_{m=1}^{n} p_{ij}^m = 0 \tag{B.5}$$

then summing along the columns we get :

$$\sum_i \lim_{n\to\infty} \frac{1}{n} \sum_{m=1}^{n} p_{ij}^{(m)} = \lim_{n\to\infty} \frac{1}{n} \sum_{m=1}^{n} \sum_i p_{ij}^{(m)} = \lim_{n\to\infty} \frac{1}{n} \sum_{m=1}^{n} 1 = 1 \tag{B.6}$$

where the permutation of the limit and the sum is possible because the sum is finite. The equations (B.5) and (B.6) are in contradiction, the chain is therefore positive recurrent.

Now to compute the stationary distribution in terms of expectation of the first return time, let's assume that $X_0 = j$ at time $t_0 = 0$ and $t_1 = \tau_{jj}$ the first return time to state j. We can define formally the set of times $t_n = min\{k > t_{n-1} : X_k = j\}$ at which the chain visits the state j. We can also define the time intervals $Y_n = t_n - t_{n-1}$ between the revisit of state j and the nth visit can be expressed at time $t_n = t_0 + Y_1 + ... + Y_n$. The evolution of the Markov chain is broken into cycles whose duration are independent and identically distributed following the same distribution as τ_{jj}. In particular we have that $\mathbb{E}(Y_n) = \mathbb{E}(\tau_{jj})$ for all $n \geq 1$.

The jth component of the stationary distribution vector π_j describes the fraction of time spent by the Markov chain in the long run on the state j, or equivalently the fraction of number of visits to that state j and recalling that there are n visits at time t_n we have :

$$\pi_j = \lim_{n\to\infty} \frac{n}{\sum_{i=1}^{n} Y_i} = \lim_{n\to\infty} \frac{1}{\frac{1}{n}\sum_{i=1}^{n} Y_i} = \frac{1}{\mathbb{E}(\tau_{jj})} \tag{B.7}$$

where the last equality stems from the strong law of large numbers. And since the chain is positive recurrent, the expectation of the first return time is finite for all state j and consequently $\pi_j > 0$ for all j. We have concluded that there exists a unique stationary distribution to the finite and irreducible Markov chain related to the expectation values and the unicity is coming form the unique representation of $\pi_j = 1/\mathbb{E}(\tau_{jj})$.

However in the limit distribution computed above we have considered a time averaged summation, the average here is in the sense of Cesáro (means of the partial sums). In practice, in order to converge to a stationary solution by iteration from any given initial distribution we need a non averaged convergence to the limiting distribution. Here is where the aperiodicity comes into play, indeed if there exist a periodic state j then the limit $\lim_{n\to\infty} \frac{1}{n}\sum_{m=1}^{n} \mathbf{P}^m$ does not exist for the matrix $\mathbf{P}$ and any iteration will result in a periodic alternation of the distributions. Therefore if the Markov chain is aperiodic, the unaveraged convergence to the limiting distribution $\pi_j = \lim_{n\to\infty} P(X_n = j)$ is possible and π is precisely the unique stationary solution of the given Markov chain.

□

Here we will discuss the famous Perron-Frobenius theorem which is often considered not only as useful but also as especially beautiful mathematics. All the following developments are taken from the book [Meyer, 2001] where the author discusses the details of each and every aspect of this theory in a whole chapter. However it is useful to summarize the whole story and discuss some important points of interest. The Perron-Frobenius theorem is an extension to the Perron's theorem for positive matrices whose core idea is to explore to what extent the positivity of a matrix reflects on the properties of its eigenvalues and their corresponding eigenvectors. Thus we will first discuss some points of the Perron's theorem and then explore how it can be extended to non-negative matrices without additional assumptions. Then we will introduce Frobenius's work which consisted of recovering most of the properties that weren't true for non-negative matrices by using the additional irreducibility assumption. Finally the effect of the primitivity of matrices is discussed to fully recover the properties of the Perron's theorem.

Theorem 2. *Let $\mathbf{A} > 0$ be a square positive matrix and $r = \rho(\mathbf{A})$ its spectral radius.*

- *$r > 0$, $r \in \sigma(\mathbf{A})$ and $mult(r) = 1$.*
- *$\exists \mathbf{x} > 0$ such that $\mathbf{A}\mathbf{x} = r\mathbf{x}$. The Perron vector $\mathbf{p} = \frac{\mathbf{x}}{||\mathbf{x}||} > 0$ such that $\sum_i p_i = 1$ is unique.*
- *r is the only eigenvalue on the spectral circle of $\mathbf{A}$.*
- *$r = \max_{x \in \mathcal{N}} f(x)$ (Collatz-Wielandt formula) where $f(x) = \min_{1 \leq i \leq n, x_i \neq 0} \frac{(\mathbf{A}\mathbf{x})_i}{x_i}$ and $\mathcal{N} = \{\mathbf{x} : \mathbf{x} \geq 0 \text{ with } \mathbf{x} \neq 0\}$.*

Proof. To show the first point we will start to notice that if $\mathbf{A} > 0$ then $r > 0$ because otherwise $\sigma(\mathbf{A}) = \{0\}$ which would mean that $\mathbf{A}$ is nilpotent. $\mathbf{A}$ cannot be nilpotent because $a_{ij} > 0 \ \forall (i, j)$.

Now we can always assume without loss of generality that $r = \rho(\mathbf{A}) = 1$ because $\mathbf{A}$ can always be renormalized by its spectral radius. Let's consider an eigenpair $(\lambda, \mathbf{x})$ of $\mathbf{A}$ with $|\lambda| = 1$ then the following inequality $|\mathbf{x}| \leq \mathbf{A}|\mathbf{x}|$ should always hold because :

$$|\mathbf{x}| = |\lambda||\mathbf{x}| = |\lambda\mathbf{x}| = |\mathbf{A}\mathbf{x}| \leq |\mathbf{A}||\mathbf{x}| = \mathbf{A}|\mathbf{x}| \tag{B.8}$$

For simplicity let's call $\mathbf{z} = \mathbf{A}|\mathbf{x}|$ and define $\mathbf{y} = \mathbf{z} - |\mathbf{x}|$. We notice that $\mathbf{y} \geq 0$ because of the inequality $\mathbf{A}|\mathbf{x}| - |\mathbf{x}| \geq 0$. Now suppose that for some i we have $y_i > 0$ then we have that $\mathbf{A}\mathbf{y} > 0$ and $\mathbf{z} > 0$, so there exists a number $\epsilon > 0$ such that $\mathbf{A}\mathbf{y} > \epsilon\mathbf{z}$ and therefore $\mathbf{A}\mathbf{z} - \mathbf{A}|\mathbf{x}| = \mathbf{A}\mathbf{z} - \mathbf{z} > \epsilon\mathbf{z}$ or equivalently :

$$\frac{\mathbf{A}}{1+\epsilon}\mathbf{z} > \mathbf{z} \tag{B.9}$$

and one can in principle multiply both sides of the inequality by $\frac{\mathbf{A}}{1+\epsilon}$ any number of times which will give rise to an ordered series because :

$$\left(\frac{\mathbf{A}}{1+\epsilon}\right)^k \mathbf{z} > \mathbf{z} \tag{B.10}$$

for all $k = 1, 2, ...$ But because $\rho(\mathbf{A}/(1+\epsilon)) = 1/(1+\epsilon) < 1$ we have that :

$$\lim_{k \to \infty} \left(\frac{\mathbf{A}}{1+\epsilon}\right)^k = 0 \tag{B.11}$$

and therefore we end up with the contradiction $0 > \mathbf{z}$. We can conclude that there are no positive elements in the vector $\mathbf{y}$ and we have $\mathbf{y} = 0 = \mathbf{A}|\mathbf{x}| - |\mathbf{x}|$ and consequently $|\mathbf{x}|$ is an eigenvector of $\mathbf{A}$ associated to the eigenvalue $\lambda = \rho(\mathbf{A}) = 1$. We can observe that the eigenvector is positive because $|\mathbf{x}| = \mathbf{A}|\mathbf{x}| = \mathbf{z} > 0$.

To show that $r = 1 = \rho(\mathbf{A})$ is a simple eigenvalue, let's suppose that its multiplicity is $m > 1$. Then there would be m linearly independent vectors associated to $\lambda = 1$. If $\mathbf{x}$ and $\mathbf{y}$ are two such independent eigenvectors then $\mathbf{x} \neq \alpha\mathbf{y}$ for all values of α. However if we select a non zero component of $\mathbf{y}$, for example $y_i \neq 0$, and set $\mathbf{z} = \mathbf{x} - (x_i/y_i)\mathbf{y}$, then because of $\mathbf{A}\mathbf{z} = \mathbf{z}$ we would have that $\mathbf{A}|\mathbf{z}| = |\mathbf{z}| > 0$ as shown before. There is a contradiction with the fact that the component $z_i = x_i - (x_i/y_i)y_i = 0$ and therefore $m > 1$ is impossible and $mult(r) = 1$ (see [Meyer, 2001] for the discussion about the semisimplicity of $\rho(\mathbf{A})$). We have also shown that there exists a positive eigenvector associated to $\lambda = 1$ and now we know that the associated subspace is one dimensional and therefore the Perron vector $\mathbf{p} > 0$ such that $\mathbf{A}\mathbf{p} = r\mathbf{p}$ with $\sum_i p_i$ is unique and it is given by $\mathbf{p} = \mathbf{x}/||\mathbf{x}||$.

In fact the implication is stronger because there are no non-negative eigenvectors other than the multiples of the Perron vector $\mathbf{p}$. To show that property one must first notice that $\mathbf{A} > 0$ implies that $\mathbf{A}^T > 0$ and since $\rho(\mathbf{A}) = \rho(\mathbf{A}^T)$ there is, in addition to the Perron eigenpair $(r, \mathbf{p})$ for $\mathbf{A}$, a Perron eigenpair $(r, \mathbf{q})$ for $\mathbf{A}^T$.

Now suppose that $(\lambda, \mathbf{y})$ is an eigenpair for $\mathbf{A}$ such that $\mathbf{y} \geq 0$, and let $\mathbf{x} > 0$ be the Perron vector for $\mathbf{A}^T$. We have that $\mathbf{x}^T\mathbf{y} > 0$ and since $\rho(\mathbf{A})\mathbf{x}^T = \mathbf{x}^T\mathbf{A}$ we get :

$$\rho(\mathbf{A})\mathbf{x}^T\mathbf{y} = \mathbf{x}^T\mathbf{A}\mathbf{y} = \lambda\mathbf{x}^T\mathbf{y} \tag{B.12}$$

implying that the eigenvalue must be the spectral radius $\lambda = \rho(\mathbf{A})$.

Now we discuss the important property which states that $\rho(\mathbf{A})$ is in fact the only eigenvalue on the spectral circle of $\mathbf{A}$. Indeed let $(\lambda, \mathbf{x})$ be an eigenpair of $\mathbf{A}$ with $|\lambda| = 1$, then as we have seen $\mathbf{A}|\mathbf{x}| = |\mathbf{x}| > 0$ so we can write $0 < |x_k| = (\mathbf{A}|x|)_k = \sum_j a_{kj}|x_j|$. At the same time we also have $|x_k| = |\lambda||x_k| = |\lambda x_k| = |(\mathbf{A}\mathbf{x})_k| = |\sum_j a_{kj}x_j|$ and therefore we obtain :

$$\left|\sum_j a_{kj}x_j\right| = \sum_j a_{kj}\,|x_j| = \sum_j |a_{kj}x_j| \tag{B.13}$$

This relation is the equality bound in the triangle inequality and therefore it is possible if and only if there exists a set of numbers $\alpha_j > 0$ such that $a_{kj}x_j = \alpha_j(a_{k1}x_1)$ or equivalently $x_j = v_j x_1$ with $v_j = (\alpha_j a_{k1}/a_{kj}) > 0$. In other words if $|\lambda| = 1$, then $\mathbf{x} = x_1\mathbf{v}$ where $\mathbf{v} = (1, v_2, ..., v_n)^T > 0$ so $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$ implies that :

$$\lambda\mathbf{v} = \mathbf{A}\mathbf{v} = |\mathbf{A}\mathbf{v}| = |\lambda\mathbf{v}| = |\lambda|\mathbf{v} = \mathbf{v} \tag{B.14}$$

and thus we conclude that $\lambda = 1$ is the only eigenvalue of $\mathbf{A}$ that lies on the spectral circle. We have thus shown how to prove the main points of the Perron's theorem apart from the Collatz-Wielandt formula that is not of directed interest in the scope of this thesis and whose proof is also discussed in [Meyer, 2001]. □

The question now is the following : can those results be generalized to non-negative matrices without additional assumptions ? The answer is that only a few results can be extended.

Theorem 3. *Let $\mathbf{A} \geq 0$ be a non negative matrix and $r = \rho(\mathbf{A})$ its spectral radius.*

- $r \in \sigma(\mathbf{A})$, *(but $r = 0$ is possible).*
- $\mathbf{A}\mathbf{z} = r\mathbf{z}$ *for some* $\mathbf{z} \in \mathcal{N} = \{\mathbf{x} : \mathbf{x} \geq 0 \ \textit{with} \ \mathbf{x} \neq 0\}$.
- $r = \max_{x\in\mathcal{N}} f(x)$ *(Collatz-Wielandt formula) where* $f(x) = \min_{1\leq i\leq n, x_i\neq 0} \frac{(\mathbf{A}\mathbf{x})_i}{x_i}$.

Proof. To prove these point we will study the behaviour of the limit when going from positive matrices towards a non-negative one. Consider the sequence of matrices given by $\mathbf{A}_k = \mathbf{A} + (1/k)\mathbf{E} > 0$ where $\mathbf{E}$ is a full matrix of 1's. We have for each $\mathbf{A}_k$ a Perron vector $\mathbf{p}_k > 0$ associated to $r_k > 0$. We can observe that the sequence $\{\mathbf{p}_k\}_{k=1}^{\infty}$ is a bounded set in the unit sphere in $\mathcal{R}^n$ and therefore the theorem of Bolzano-Weierstrass applies and one can extract a convergent subsequence $\{\mathbf{p}_{k_i}\}_{i=1}^{\infty} \to \mathbf{z}$ where $\mathbf{z} \geq 0$ is a non zero vector (because the Perron vectors are positive and of norm 1).

In one direction we have that $\mathbf{A}_1 > \mathbf{A}_2 > ... > \mathbf{A}$ so that $r_1 \geq r_2 \geq ... \geq r$, we can see that $\{r_k\}_{k=1}^{\infty}$ is a monotonic sequence of positive numbers bounded below by r. Therefore $\lim_{k\to\infty} r_k = r^*$ exists and $r^* \geq r$. In particular we can extract a convergent subsequence $\lim_{i\to\infty} r_{k_i} = r^* \geq r$.

In the other direction we can notice that $\lim_{k\to\infty} \mathbf{A}_k = \mathbf{A}$ implies $\lim_{i\to\infty} \mathbf{A}_{k_i} = \mathbf{A}$ and using the fact that, if all the limits exist, the limit of a product is the product of the limits, we obtain :

$$\mathbf{A}\mathbf{z} = \lim_{i\to\infty} \mathbf{A}_{k_i}\mathbf{p}_{k_i} = \lim_{i\to\infty} r_{k_i}\mathbf{p}_{k_i} = r^*\mathbf{z} \tag{B.15}$$

which means that $r^* \in \sigma(\mathbf{A})$ and consequently $r^* \leq r$. Therefore we conclude that $r^* = r$ and $\mathbf{A}\mathbf{z} = r\mathbf{z}$ with $\mathbf{z} \geq 0$ and $\mathbf{z} \neq 0$. □

Instead of ending the discussion here, Frobenius saw that the existence of some zero elements in the matrices is not a problem in itself, it is rather their position which is crucial to the validity of Perron's properties. For example some of the properties that do not hold for $\mathbf{A} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ are still respected for $\mathbf{B} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$. Frobenius was a genius, he understood that the difference between $\mathbf{A}$ and $\mathbf{B}$ is the irreducibility of the matrix and succeeded in relating it to the spectral properties of non-negative matrices. To understand how the irreducibility saves most of Perron's properties in the case of non negative matrices, we will need the following lemma.

Theorem 4. *If $\mathbf{A}_{n\times n} \geq 0$ is irreducible, then $(\mathbf{Id} + \mathbf{A})^{n-1} > 0$.*

Proof. This rule provides an extremely useful conversion tool from a non-negative matrix to a positive one, however it needs the irreducibility assumption. Indeed if we notice that an entry $a_{ij}^{(k)}$ of $\mathbf{A}^k$ is expressed as :

$$a_{ij}^{(k)} = \sum_{h_1,...,h_{k-1}} a_{ih_1} a_{h_1h_2} ... a_{h_{k-1}j} > 0 \tag{B.16}$$

if and only if there exists a set of indices $h_1, h_2, ..., h_{k-1}$ such that $a_{ih_1} > 0$ and $a_{h_1h_2} > 0$ and ... and $a_{h_{k-1}j} > 0$. From a graph point of view, it means that there is a sequence of k paths leading from node i to node j if and only if $a_{ij}^{(k)} > 0$. The irreducibility of $\mathbf{A}$ ensures that its associated graph is strongly connected so for any pair (i, j) of nodes there is a sequence of k paths connecting them and therefore the following relation is guaranteed for each i and j :

$$\left[(\mathbf{Id} + \mathbf{A})^{n-1}\right]_{ij} = \left[\sum_{k=0}^{n-1} \binom{n-1}{k} \mathbf{A}^k\right]_{ij} = \sum_{k=0}^{n-1} \binom{n-1}{k} a_{ij}^{(k)} > 0. \tag{B.17}$$

□

Now we can look at the extension of Perron's theorem with the irreducibility assumption, the Perron-Frobenius theorem.

Theorem 5. *Let $\mathbf{A} \geq 0$ be an irreducible matrix and $r = \rho(\mathbf{A})$ its spectral radius.*

- *$r > 0$, $r \in \sigma(\mathbf{A})$ and $mult(r) = 1$.*
- *$\exists \mathbf{x} > 0$ such that $\mathbf{Ax} = r\mathbf{x}$. The Perron vector $\mathbf{p} = \frac{\mathbf{x}}{||\mathbf{x}||} > 0$ such that $\sum_i p_i = 1$ is unique.*
- *$r = \max_{x\in\mathcal{N}} f(x)$ (Collatz-Wielandt formula)*
  where $f(x) = \min_{1\leq i\leq n, x_i\neq 0} \frac{(\mathbf{Ax})_i}{x_i}$ and $\mathcal{N} = \{\mathbf{x} : \mathbf{x} \geq 0 \text{ with } \mathbf{x} \neq 0\}$.

Proof. We already know that $r = \rho(\mathbf{A}) \in \sigma(\mathbf{A})$. To show that the multiplicity of r is one, let's consider the matrix $\mathbf{B} = (\mathbf{Id} + \mathbf{A})^{n-1} > 0$ thanks to the irreducibility of $\mathbf{A}$, then $\lambda \in \sigma(\mathbf{A})$ if and only if $(1 + \lambda)^{n-1} \in \sigma(\mathbf{B})$, and $(1 + \lambda)^{n-1}$ and λ have the same multiplicity. Consequently if $\mu = \rho(\mathbf{B})$, then :

$$\mu = \max_{\lambda\in\sigma(\mathbf{A})} |(1 + \lambda)^{n-1}| = \left\{\max_{\lambda\in\sigma(\mathbf{A})} |(1 + \lambda)|\right\}^{n-1} = (1 + r)^{n-1} \tag{B.18}$$

Since $\mathbf{B} > 0$, the multiplicity of μ cannot be $mult(\mu) > 1$ and therefore $mult(r) = 1$.

To check that $\mathbf{A}$ has a positive eigenvector associated with r, recall that if $(\lambda, \mathbf{x})$ is an eigenpair of $\mathbf{A}$ then $(f(\lambda), \mathbf{x})$ is an eigenpair of $f(\mathbf{A})$. We already know that there exists a non negative eigenvector $\mathbf{x} \geq 0$ associated with r so $(\lambda, \mathbf{x})$ being an eigenpair for $\mathbf{A}$ implies that $(\mu, \mathbf{x})$ is an eigenpair for $\mathbf{B}$. We also know from the Perron's theorem that $\mathbf{x}$ must be a positive multiple of

the Perron vector of $\mathbf{B}$ and therefore it must in fact be positive $\mathbf{x} > 0$. And now because $\mathbf{A} \geq 0$ and $\mathbf{x} > 0$ forces $\mathbf{Ax} > 0$, we have that $\mathbf{Ax} \neq 0$ and therefore $r > 0$.

The unicity can be proven using the same development from the theorem for non negative matrices and the Collatz-Wielandt formula which was also valid for the non negative case and is still valid here is not discussed here, the proof can be found in the reference [Meyer, 2001]. □

Finally the only property left that the irreducibility cannot recover is the statement that the spectral radius is the only eigenvalue on the spectral circle. In fact the set of non-negative irreducible matrices are divided into two classes based on whether they admit more than one eigenvalue on the spectral circle or not, those are respectively called *imprimitive* and *primitive* matrices. A non negative and irreducible matrix $\mathbf{A}$ with $r = \rho(\mathbf{A})$ is primitive if and only if the limit $\lim_{k\to\infty}(\mathbf{A}/r)^k$ exists. In fact this requirement is akin to the aperiodicity requirement for Markov chains to remove the unaveraged alternating states in the long run.

Frobenius also showed that there is a simple way of testing if a matrix is primitive.

Theorem 6. *A matrix $\mathbf{A} \geq 0$ is primitive if and only if $\mathbf{A}^m > 0$ for some $m > 0$.*

Proof. First let's assume that for some $m > 0$ we have $\mathbf{A}^m > 0$, then $\mathbf{A}$ is irreducible because otherwise there would exist a permutation matrix such that :

$$\mathbf{A} = \mathbf{P}\begin{pmatrix}\mathbf{X} & \mathbf{Y}\\ \mathbf{0} & \mathbf{Z}\end{pmatrix}\mathbf{P}^T \implies \mathbf{A}^m = \mathbf{P}\begin{pmatrix}\mathbf{X}^m & \star\\ \mathbf{0} & \mathbf{Z}^m\end{pmatrix}\mathbf{P}^T \text{ has zero entries.} \tag{B.19}$$

Suppose that $\mathbf{A}$ has h eigenvalues $\{\lambda_1, \lambda_2, ..., \lambda_h\}$ on its spectral circle so that $r = \rho(\mathbf{A}) = |\lambda_1| = ... = |\lambda_h| > |\lambda_{h+1}| > ... > |\lambda_n|$. Since $\lambda \in \sigma(\mathbf{A})$ implies $\lambda^m \in \sigma(\mathbf{A}^m)$ with $mult(\lambda_k) = mult(\lambda_k^m)$. Perron's theorem insures that $\mathbf{A}^m$ has only one eigenvalue (which must be r^m) on its spectral circle, so $r^m = \lambda_1^m = \lambda_2^m = ... = \lambda_h^m$. But this means that $mult(r) = mult(r^m) = h$, and therefore $h = 1$.

Conversely, if $\mathbf{A}$ is primitive with $r = \rho(\mathbf{A})$, then $\lim_{k\to\infty}(\mathbf{A}/r)^k > 0$. Hence there must be some m such that $(\mathbf{A}/r)^m > 0$ and thus $\mathbf{A}^m > 0$. □

For the last theorem, we present here the proof directly taken from [Langville and Meyer, 2006].

Theorem 7. *If the spectrum of the stochastic matrix S is $\{1, \lambda_1, \lambda_2, ..., \lambda_n\}$, then the spectrum of the Google matrix $\mathbf{G} = \alpha\mathbf{S} + (1-\alpha)\mathbf{e}\mathbf{v}^T$ is $\{1, \alpha\lambda_1, \alpha\lambda_2, ..., \alpha\lambda_n\}$, where $\mathbf{v}^T$ is a probability vector.*

Proof. Since $\mathbf{S}$ is stochastic, $(1, \mathbf{e})$ is an eigenpair of $\mathbf{S}$. Let $\mathbf{Q} = \begin{pmatrix}\mathbf{e} & \mathbf{X}\end{pmatrix}$ be a non singular matrix that has the eigenvector $\mathbf{e}$ as its first column. Let $\mathbf{Q}^{-1} = \begin{pmatrix}\mathbf{y}^T\\ \mathbf{Y}^T\end{pmatrix}$. Then $\mathbf{Q}^{-1}\mathbf{Q} = \begin{pmatrix}\mathbf{y}^T\mathbf{e} & \mathbf{y}^T\mathbf{X}\\ \mathbf{Y}^T\mathbf{e} & \mathbf{Y}^T\mathbf{X}\end{pmatrix} = \begin{pmatrix}1 & 0\\ \mathbf{0} & \mathbf{Id}\end{pmatrix}$, which gives two useful identities, $\mathbf{y}^T\mathbf{e} = 1$ and $\mathbf{Y}^T\mathbf{e} = \mathbf{0}$. As a result the similarity transformation :

$$\mathbf{Q}^{-1}\mathbf{S}\mathbf{Q} = \begin{pmatrix}\mathbf{y}^T\mathbf{e} & \mathbf{y}^T\mathbf{S}\mathbf{X}\\ \mathbf{Y}^T\mathbf{e} & \mathbf{Y}^T\mathbf{S}\mathbf{X}\end{pmatrix} = \begin{pmatrix}1 & \mathbf{y}^T\mathbf{S}\mathbf{X}\\ 0 & \mathbf{Y}^T\mathbf{S}\mathbf{X}\end{pmatrix} \tag{B.20}$$

reveals that $\mathbf{Y}^T\mathbf{S}\mathbf{X}$ contains the remaining eigenvalues of $\mathbf{S}$, $\lambda_2, ..., \lambda_n$. Applying the similarity transformation to $\mathbf{G} = \alpha\mathbf{S} + (1-\alpha)\mathbf{e}\mathbf{v}^T$ gives :

$$\mathbf{Q}^{-1}(\alpha\mathbf{S} + (1-\alpha)\mathbf{e}\mathbf{v}^T)\mathbf{Q} = \alpha\mathbf{Q}^{-1}\mathbf{S}\mathbf{Q} + (1-\alpha)\mathbf{Q}^{-1}\mathbf{e}\mathbf{v}^T\mathbf{Q} \tag{B.21}$$

$$= \begin{pmatrix} \alpha & \alpha\mathbf{y}^T\mathbf{S}\mathbf{X} \\ 0 & \alpha\mathbf{Y}^T\mathbf{S}\mathbf{X} \end{pmatrix} + (1-\alpha)\begin{pmatrix} \mathbf{y}^T\mathbf{e} \\ \mathbf{Y}^T\mathbf{e} \end{pmatrix}\begin{pmatrix} \mathbf{v}^T\mathbf{e} & \mathbf{v}^T\mathbf{X} \end{pmatrix} \tag{B.22}$$

$$= \begin{pmatrix} \alpha & \alpha\mathbf{y}^T\mathbf{S}\mathbf{X} \\ 0 & \alpha\mathbf{Y}^T\mathbf{S}\mathbf{X} \end{pmatrix} + \begin{pmatrix} (1-\alpha) & (1-\alpha)\mathbf{v}^T\mathbf{X} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \tag{B.23}$$

$$= \begin{pmatrix} 1 & \alpha\mathbf{y}^T\mathbf{S}\mathbf{X} + (1-\alpha)\mathbf{v}^T\mathbf{X} \\ \mathbf{0} & \alpha\mathbf{Y}^T\mathbf{S}\mathbf{X} \end{pmatrix}. \tag{B.24}$$

Therefore, the eigenvalues of $\mathbf{G} = \alpha\mathbf{S} + (1-\alpha)\mathbf{e}\mathbf{v}^T$ are $\{1, \alpha\lambda_2, \alpha\lambda_3, ..., \alpha\lambda_n\}$.

□

Appendix C - References

Publications

[1] V. Kandiah and D. Shepelyansky, *PageRank model of opinion formation on social networks*, Physica A, 391 (2012).
Chapter 6 - The use of PageRank in opinion formation models

[2] V. Kandiah and D. Shepelyansky, *Google Matrix Analysis of DNA Sequences*, PLoS ONE, 8 (2013).
Chapter 3 - The analysis of DNA sequences

[3] V. Kandiah and D. Shepelyansky, *Google matrix analysis of C.elegans neural network*, Phys. let. A, 378 (2014).
Chapter 4 - The network of *C.elegans* neurons

[4] V. Kandiah, B. Georgeot and O. Giraud, *Move ordering and communities in complex networks describing the game of go*, EPJB (2014).
Chapter 5 - The game of Go from a complex network perspective

Bibliography

[Abel and Shepelyansky, 2011] Abel, M. W. and Shepelyansky, D. L. (2011). Google matrix of business process management. *The European Physical Journal B*, 84(4):493–500.

[Albert and Barabási, 2002] Albert, R. and Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97.

[Albert et al., 1999] Albert, R., Jeong, H., and Barabási, A.-L. (1999). Internet: Diameter of the world-wide web. *Nature*, 401(6749):130–131.

[Arnoldi, 1951] Arnoldi, W. E. (1951). The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, 9:17–29.

[Arora and Barak, 2009] Arora, S. and Barak, B. (2009). *Computational Complexity: A Modern Approach*. Cambridge University Press, Cambridge ; New York, 1 edition edition.

[Bánky et al., 2013] Bánky, D., Iván, G., and Grolmusz, V. (2013). Equal opportunity for low-degree network nodes: A PageRank-based method for protein target identification in metabolic graphs. *PLoS ONE*, 8(1):e54204.

[Barabási and Albert, 1999] Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439):509–512.

[Barrett et al., 1987] Barrett, R., Berry, M., Chan, T. F., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., and Vorst, H. v. d. (1987). *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. Society for Industrial and Applied Mathematics, Philadelphia, 1 edition edition.

[BBP, 2014] BBP (2014). Blue brain project. http://bluebrain.epfl.ch/.

[Berners-Lee, 1989] Berners-Lee, T. (1989). Information management: A proposal.

[Blasius and Tönjes, 2009] Blasius, B. and Tönjes, R. (2009). Zipf's law in the popularity distribution of chess openings. *Physical Review Letters*, 103(21):218701.

[Boldi, 2005] Boldi, P. (2005). Totalrank: Ranking without damping.

[Bollobás, 1980] Bollobás, B. (1980). A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. *European Journal of Combinatorics*, 1(4):311–316.

[Bouzy and Chaslot, 2005] Bouzy, B. and Chaslot, G. (2005). Monte-carlo go reinforcement learning experiments. *Proc. IEEE Symp.Comput. Intell. Games, Colchester, U. K. , p. 176*.

[Browne et al., 2012] Browne, C., Powley, E., Whitehouse, D., Lucas, S., Cowling, P., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., and Colton, S. (2012). A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43.

[Burks et al., 1946] Burks, A. W., Goldstine, H. H., and Von Neumann, J. (1946). Preliminary discussion of the logical design of an electronic computer instrument. http://deepblue.lib.umich.edu/bitstream/2027.42/3972/5/bab6286.0001.001.pdf.

[Caldarelli, 2007] Caldarelli, G. (2007). *Scale-Free Networks: Complex Webs in Nature and Technology*. Oxford Finance Series.

[Cancho and Solé, 2001] Cancho, R. F. i. and Solé, R. V. (2001). The small world of human language. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 268(1482):2261–2265.

[Castellano et al., 2009] Castellano, C., Fortunato, S., and Loreto, V. (2009). Statistical physics of social dynamics. *Reviews of Modern Physics*, 81(2):591–646.

[Chaslot et al., 2006] Chaslot, G., Saito, J. T., Bouzy, B., Uiterwijk, J., and van den Herik, H. J. (2006). Monte-carlo strategies for computer go. *Proc. of the 18th BeNeLux Conf. on Artificial Intelligence*.

[Chepelianskii, 2010] Chepelianskii, A. D. (2010). Towards physical laws for software architecture. *arXiv:1003.5455 [physics]*.

[Cohen and Havlin, 2003] Cohen, R. and Havlin, S. (2003). Scale-free networks are ultrasmall. *Physical Review Letters*, 90(5):058701.

[Corominas-Murtra et al., 2009] Corominas-Murtra, B., Valverde, S., and Solé, R. (2009). The ontogeny of scale-free syntax networks : Phase transitions in early language acquisition. *Advances in Complex Systems*, 12(03):371–392.

[Coulom, 2007a] Coulom, R. (2007a). Computing elo ratings of move patterns in the game of go. *Proc. Comput. Games Workshop, Amsterdam, The Netherlands*, page 113.

[Coulom, 2007b] Coulom, R. (2007b). Proceedings of the 5th international conference on computer and games. *Lect. Notes in Comp. Sciences*, 4630:72.

[Dai et al., 2008] Dai, Q., Yang, Y., and Wang, T. (2008). Markov model plus k-word distributions: a synergy that produces novel statistical measures for sequence comparison. *Bioinformatics (Oxford, England)*, 24(20):2296–2302.

[Donato et al., 2004] Donato, D., Laura, L., Leonardi, S., and Millozzi, S. (2004). Large scale properties of the webgraph. *The European Physical Journal B - Condensed Matter*, 38(2):239–243.

[Dorogovtsev, 2010] Dorogovtsev, S. N. (2010). Lectures on complex networks.

[Dorogovtsev et al., 2008] Dorogovtsev, S. N., Goltsev, A. V., and Mendes, J. F. (2008). Critical phenomena in complex networks. *Reviews of Modern Physics*, 80(4):1275.

[Dorogovtsev and Mendes, 2003] Dorogovtsev, S. N. and Mendes, J. F. F. (2003). Evolution of networks: From biological nets to the internet and WWW.

[Eguíluz et al., 2005] Eguíluz, V. M., Chialvo, D. R., Cecchi, G. A., Baliki, M., and Apkarian, A. V. (2005). Scale-free brain functional networks. *Physical Review Letters*, 94(1):018102.

[EMBL, 2013] EMBL (2013). Ensembl genome data base. ftp://ftp.ensembl.org/pub/release-62/genbank/.

[Eom et al., 2014] Eom, Y.-H., Aragón, P., Laniado, D., Kaltenbrunner, A., Vigna, S., and Shepelyansky, D. L. (2014). Interactions of cultures and top people of wikipedia from ranking of 24 language editions. *arXiv preprint arXiv:1405.7183*.

[Erdős, 1959] Erdős, R. (1959). On random graphs, i. *Publ. Math. Debrecen*, 6:290 – 297.

[Erdős, 1960] Erdős, R. (1960). On the evolution of random graphs. In *Publication of the mathematical institute of the hungarian academy of sciences*, pages 17–61.

[Ermann et al., 2012] Ermann, L., Chepelianskii, A. D., and Shepelyansky, D. L. (2012). Toward two-dimensional search engines. *Journal of Physics A: Mathematical and Theoretical*, 45(27):275101.

[Ermann et al., 2013] Ermann, L., Frahm, K. M., and Shepelyansky, D. L. (2013). Spectral properties of google matrix of wikipedia and other networks. *The European Physical Journal B*, 86(5).

[Ermann and Shepelyansky, 2011] Ermann, L. and Shepelyansky, D. L. (2011). Google matrix of the world trade network. *Acta Physica Polonica A*, 120(6A).

[Euler, 1736] Euler, L. (1736). Solutio problematis ad geometriam situs pertinentis. *Comment. Acad. Sci. U. Petrop.*, 8:128–140.

[Fortunato, 2010] Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(3–5):75–174.

[Frahm et al., 2014] Frahm, K. M., Eom, Y.-H., and Shepelyansky, D. L. (2014). Google matrix of the citation network of physical review. *Physical Review E*, 89(5):052814.

[Frahm et al., 2011] Frahm, K. M., Georgeot, B., and Shepelyansky, D. L. (2011). Universal emergence of PageRank. *Journal of Physics A: Mathematical and Theoretical*, 44(46):465101.

[Frahm and Shepelyansky, 2012] Frahm, K. M. and Shepelyansky, D. L. (2012). Poincaré recurrences of DNA sequences. *Physical Review E*, 85(1).

[Franklin and Gosling, 1953] Franklin, R. and Gosling, R. (1953). Molecular configuration in sodium thymonucleate. *Nature*, 171(4356):740–741.

[Freeman, 1977] Freeman, L. (1977). A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41.

[Freeman, 1979] Freeman, L. C. (1979). Centrality in social networks; conceptual clarification. *Social Networks*, 1:215–239.

[Frobenius, 1912] Frobenius, G. (1912). Über matrizen aus nicht negativen elementen. *Preussiche Akademie der Wissenschaften zu Berlin*, pages 456–477.

[Gabaix, 1999] Gabaix, X. (1999). Zipf's law for cities: An explanation. *The Quarterly Journal of Economics*, 114(3):739–767.

[Galam, 1986] Galam, S. (1986). Majority rule, hierarchical structures and democratic totalitarism: a statistical approach. *J. Math. Psychology*, 30(426).

[Galam, 2005] Galam, S. (2005). Local dynamics vs. social mechanisms: A unifying frame. *EPL (Europhysics Letters)*, 70(6):705.

[Galam, 2008] Galam, S. (2008). Sociophysics : A review of galam models. *International Journal of Modern Physics C*, 19(03):409–440.

[Galam and Walliser, 2010] Galam, S. and Walliser, B. (2010). Ising model versus normal form game. *Physica A: Statistical Mechanics and its Applications*, 389(3):481–489.

[Gelly et al., 2012] Gelly, S., Kocsis, L., Schoenauer, M., Sebag, M., Silver, D., Szepesvári, C., and Teytaud, O. (2012). The grand challenge of computer go: Monte carlo tree search and extensions. *Commun. ACM*, 55(3):106–113.

[Gelly and Silver, 2011] Gelly, S. and Silver, D. (2011). Monte-carlo tree search and rapid action value estimation in computer go. *Artificial Intelligence*, 175(11):1856–1875.

[Georgeot and Giraud, 2012] Georgeot, B. and Giraud, O. (2012). The game of go as a complex network. *EPL (Europhysics Letters)*, 97(6):68002.

[Georgeot et al., 2010] Georgeot, B., Giraud, O., and Shepelyansky, D. L. (2010). Spectral properties of the google matrix of the world wide web and other directed networks. *Physical Review E*, 81(5):056109.

[Ginsberg et al., 2009] Ginsberg, J., Mohebbi, M. H., Patel, R. S., Brammer, L., Smolinski, M. S., and Brilliant, L. (2009). Detecting influenza epidemics using search engine query data. *Nature*, 457(7232):1012–1014.

[Giraud et al., 2009] Giraud, O., Georgeot, B., and Shepelyansky, D. L. (2009). Delocalization transition for the google matrix. *Physical Review E*, 80(2):026107.

[Girvan and Newman, 2002] Girvan, M. and Newman, M. E. J. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826.

[Gutiérrez et al., 2007] Gutiérrez, M. C., Castillo, A. M., Kamekura, M., Xue, Y., Ma, Y., Cowan, D. A., Jones, B. E., Grant, W. D., and Ventosa, A. (2007). Halopiger xanaduensis gen. nov., sp. nov., an extremely halophilic archaeon isolated from saline lake shangmatala in inner mongolia, china. *International Journal of Systematic and Evolutionary Microbiology*, 57(Pt 7):1402–1407.

[Halpern et al., 2007] Halpern, D., Chiapello, H., Schbath, S., Robin, S., Hennequet-Antier, C., Gruss, A., and El Karoui, M. (2007). Identification of DNA motifs implicated in maintenance of bacterial core genomes by predictive modeling. *PLoS Genet*, 3(9):e153.

[Henmon and Zipf, 1936] Henmon, V. A. C. and Zipf, G. K. (1936). The psycho-biology of language: An introduction to dynamic philology. *The Modern Language Journal*, 21(2):125.

[Holland, 2006] Holland, B. (2006). Distance based methods for estimating phylogenetic trees.

[Huang et al., 2011] Huang, S.-C., Coulom, R., and Lin, S.-S. (2011). Monte-carlo simulation balancing in practice. In Herik, H. J. v. d., Iida, H., and Plaat, A., editors, *Computers and Games*, number 6515 in Lecture Notes in Computer Science, pages 81–92. Springer Berlin Heidelberg.

[Humphries et al., 2006] Humphries, M., Gurney, K., and Prescott, T. (2006). The brainstem reticular formation is a small-world, not scale-free, network. *Proceedings of the Royal Society B: Biological Sciences*, 273(1585):503–511.

[Izhikevich, 2007] Izhikevich, E. M. (2007). *Dynamical Systems in Neuroscience*. MIT Press.

[Kaltenbrunner et al., 2014] Kaltenbrunner, A., Aragón, P., Laniado, D., and Volkovich, Y. (2014). Not all paths lead to rome: Analysing the network of sister cities. In *Self-Organizing Systems*, pages 151–156. Springer Berlin Heidelberg.

[Kleinberg, 1999] Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632.

[Krapivsky et al., 2010] Krapivsky, P. L., Redner, S., and Ben-Naim, E. (2010). A kinetic view of statistical physics.

[Krzakala et al., 2013] Krzakala, F., Moore, C., Mossel, E., Neeman, J., Sly, A., Zdeborová, L., and Zhang, P. (2013). Spectral redemption in clustering sparse networks. *Proceedings of the National Academy of Sciences*, 110(52):20935–20940.

[Langville and Meyer, 2006] Langville, A. N. and Meyer, C. D. (2006). *Google's PageRank and Beyond: The Science of Search Engine Rankings*. Princeton University Press, Princeton, NJ, USA.

[Lee et al., 2005] Lee, W.-C. A., Huang, H., Feng, G., Sanes, J. R., Brown, E. N., So, P. T., and Nedivi, E. (2005). Dynamic remodeling of dendritic arbors in GABAergic interneurons of adult visual cortex. *PLoS Biol*, 4(2):e29.

[Leicht and Newman, 2008] Leicht, E. A. and Newman, M. E. J. (2008). Community structure in directed networks. *Physical Review Letters*, 100(11):118703.

[Levin et al., 2008] Levin, D. A., Peres, Y., and Wilmer, E. L. (2008). *Markov Chains and Mixing Times*. American Mathematical Society, Providence, R.I, 1 edition edition.

[Li et al., 2004] Li, S., Armstrong, C. M., Bertin, N., Ge, H., Milstein, S., Boxem, M., Vidalain, P.-O., Han, J.-D. J., Chesneau, A., Hao, T., Goldberg, D. S., Li, N., Martinez, M., Rual, J.-F., Lamesch, P., Xu, L., Tewari, M., Wong, S. L., Zhang, L. V., Berriz, G. F., Jacotot, L., Vaglio, P., Reboul, J., Hirozane-Kishikawa, T., Li, Q., Gabel, H. W., Elewa, A., Baumgartner, B., Rose, D. J., Yu, H., Bosak, S., Sequerra, R., Fraser, A., Mango, S. E., Saxton, W. M., Strome, S., Heuvel, S. v. d., Piano, F., Vandenhaute, J., Sardet, C., Gerstein, M., Doucette-Stamm, L., Gunsalus, K. C., Harper, J. W., Cusick, M. E., Roth, F. P., Hill, D. E., and Vidal, M. (2004). A map of the interactome network of the metazoan c. elegans. *Science*, 303(5657):540–543.

[Liggett, 1999] Liggett, T. M. (1999). *Stochastic Interacting Systems: Contact, Voter and Exclusion Processes*. Springer, Berlin.

[Malliaros and Vazirgiannis, 2013] Malliaros, F. D. and Vazirgiannis, M. (2013). Clustering and community detection in directed networks: A survey. *Physics Reports*, 533(4):95–142.

[Mantegna et al., 1995] Mantegna, R., Buldyrev, S., Goldberger, A., Havlin, S., Peng, C.-K., Simons, M., and Stanley, H. (1995). Systematic analysis of coding and noncoding DNA sequences using methods of statistical linguistics. *Physical Review E*, 52(3):2939–2950.

[Markov, 1906] Markov, A. (1906). Extension of the law of large numbers to dependent quantities (in russian). *Izvestiya Fiziko-Matematicheskikh Obschestva Kazan University*, 15:135–156.

[Mehta, 2004] Mehta, M. (2004). *Random Matrices*. Elsevier/Academic Press.

[Meyer, 2001] Meyer, C. D. (2001). *Matrix Analysis and Applied Linear Algebra Book and Solutions Manual*. SIAM: Society for Industrial and Applied Mathematics, Philadelphia, Pa, har/cdr edition edition.

[Meyn et al., 2009] Meyn, S., Tweedie, R. L., and Glynn, P. W. (2009). *Markov Chains and Stochastic Stability*. Cambridge University Press, Cambridge ; New York, 2 edition edition.

[Milgram, 1967] Milgram, S. (1967). The small world problem. *Psychology Today*, 67:61–67.

[Mises and Pollaczek-Geiringer, 1929] Mises, R. V. and Pollaczek-Geiringer, H. (1929). Praktische verfahren der gleichungsauflösung . *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 9(2):152–164.

[Nelsen, 2006] Nelsen, R. B. (2006). *An Introduction to Copulas*. Springer - Series in statistics.

[Neuzil et al., 2006] Neuzil, P., Zhang, C., Pipper, J., Oh, S., and Zhuo, L. (2006). Ultra fast miniaturized real-time PCR: 40 cycles in less than six minutes. *Nucleic Acids Research*, 34(11):e77.

[Newman, 2005] Newman, M. J. (2005). A measure of betweenness centrality based on random walks. *Social Networks*, 27(1):39–54.

[Norris, 1998] Norris, J. R. (1998). *Markov Chains*. Cambridge University Press, Cambridge, UK ; New York.

[Okuyama et al., 1999] Okuyama, K., Takayasu, M., and Takayasu, H. (1999). Zipf's law in income distribution of companies. *Physica A: Statistical Mechanics and its Applications*, 269(1):125–131.

[Page et al., 1999] Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The PageRank citation ranking: Bringing order to the web.

[Park et al., 2014] Park, S.-J., Ghai, R., Martín-Cuadrado, A.-B., Rodríguez-Valera, F., Chung, W.-H., Kwon, K., Lee, J.-H., Madsen, E. L., and Rhee, S.-K. (2014). Genomes of two new ammonia-oxidizing archaea enriched from deep marine sediments. *PLoS ONE*, 9(5):e96449.

[Perron, 1907] Perron, O. (1907). Grundlagen für eine theorie des jacobischen kettenbruchalgorithmus. *Mathematische Annalen*, 64:248–263.

[Preis et al., 2010] Preis, T., Reith, D., and Stanley, H. E. (2010). Complex dynamics of our economic life on different scales: insights from search engine query data. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 368(1933):5707–5719.

[Radicchi, 2011] Radicchi, F. (2011). Who is the best player ever? a complex network analysis of the history of professional tennis. *PLoS ONE*, 6(2):e17249.

[Radicchi et al., 2009] Radicchi, F., Fortunato, S., Markines, B., and Vespignani, A. (2009). Diffusion of scientific credits and the ranking of scientists. *Physical Review E*, 80(5):056103.

[Randić and DeAlba, 1997] Randić, M. and DeAlba, L. M. (1997). Dense graphs and sparse matrices. *J. Chem. Inf. Comput. Sci.*, 6(37):1078–1081.

[Redner, 2005] Redner, S. (2005). Citation statistics from 110 years of physical review. *Physics Today*, 58(6):49–54.

[Reinert et al., 2009] Reinert, G., Chew, D., Sun, F., and Waterman, M. S. (2009). Alignment-free sequence comparison (i): Statistics and power. *Journal of Computational Biology*, 16(12):1615–1634.

[Robin et al., 2005] Robin, S., Rodolphe, F., and Schbath, S. (2005). DNA, words and models : Statistics of exceptional words.

[Rodier et al., 2014] Rodier, X., Le Couédic, M., Hautefeuille, F., Leturcq, S., Jouve, B., and Fieux, E. (2014). From space to graphs to understand spatial changes using medieval and modern fiscal sources. *Archaeology in the Digital Era*, 31:427.

[Rudnick and Gaspari, 2004] Rudnick, J. and Gaspari, G. (2004). *Elements of the Random Walk: An introduction for Advanced Students and Researchers*. Cambridge University Press, Cambridge ; New York.

[Saavedra et al., 2011] Saavedra, S., Duch, J., and Uzzi, B. (2011). Tracking traders' understanding of the market using e-communication data. *PLoS ONE*, 6(10):e26705.

[Saitou and Nei, 1987] Saitou, N. and Nei, M. (1987). The neighbor-joining method : a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.*, 4(4):406–425.

[Schmittmann and Mukhopadhyay, 2010] Schmittmann, B. and Mukhopadhyay, A. (2010). Opinion formation on adaptive networks with intensive average degree. *Physical Review E*, 82(6):066104.

[Schraudolph et al., 1994] Schraudolph, N. N., Dayan, P., and Sejnowski, T. J. (1994). Temporal difference learning of position evaluation in the game of go. In *Advances in Neural Information Processing Systems 6*, page 817–824. Morgan Kaufmann.

[SCRG, 2006] SCRG (2006). Academic web link database project. http://cybermetrics.wlv.ac.uk/database/.

[Shepelyansky and Zhirov, 2010] Shepelyansky, D. L. and Zhirov, O. V. (2010). Towards google matrix of brain. *Physics Letters A*, 374(31–32):3206–3209.

[Sigman, 2009] Sigman, K. (2009). Markov chains ii: recurrence and limiting (stationary) distributions. http://www.columbia.edu/ ks20/stochastic-I/stochastic-I-MCII.pdf.

[Small et al., 2008] Small, M., Xu, X., Zhou, J., Zhang, J., Sun, J., and Lu, J.-A. (2008). Scale-free networks which are highly assortative but not small world. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 77(6 Pt 2):066112.

[Sokal and Michener, 1958] Sokal, R. R. and Michener, C. D. (1958). A statistical method for evaluating systematic relationships. *University of Kansas Scientific Bulletin*, 28:1409–1438.

[Sood and Redner, 2005] Sood, V. and Redner, S. (2005). Voter model on heterogeneous graphs. *Physical Review Letters*, 94(17):178701.

[Srikant, 2009] Srikant, R. (2009). Discrete-time markov chains. https://sites.google.com/site/srikantece534/lecture-notes.

[Stewart, 1995] Stewart, W. J. (1995). *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, Princeton, N.J.

[Sznajd-Weron and Józef, 2000] Sznajd-Weron, K. and Józef, S. (2000). Opinion evolution in closed community. *International Journal of Modern Physics C*, 11(06):1157–1165.

[Titz et al., 2008] Titz, B., Rajagopala, S. V., Goll, J., Häuser, R., McKevitt, M. T., Palzkill, T., and Uetz, P. (2008). The binary protein interactome of treponema pallidum – the syphilis spirochete. *PLoS ONE*, 3(5):e2292.

[Towlson et al., 2013] Towlson, E. K., Vértes, P. E., Ahnert, S. E., Schafer, W. R., and Bullmore, E. T. (2013). The rich club of the c. elegans neuronal connectome. *The Journal of neuroscience: the official journal of the Society for Neuroscience*, 33(15):6380–6387.

[Tromp and Farnebäck, 2007] Tromp, J. and Farnebäck, G. (2007). Proceedings of the 5th international conference on computer and games. *Lect. Notes in Comp. Sciences*, 4630:84.

[Turing, 1937] Turing, A. M. (1937). On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42(1):230–265.

[U-go, 2013] U-go (2013). http://www.u-go.net/.

[van den Herik et al., 2002] van den Herik, H. J., Uiterwijk, J. W. H. M., and van Rijswijck, J. (2002). Games solved: Now and in the future. *Artificial Intelligence*, 134(1–2):277–311.

[Vespignani, 2009] Vespignani, A. (2009). Predicting the behavior of techno-social systems. *Science*, 325(5939):425–428.

[Wang and Gelly, 2007] Wang, Y. and Gelly, S. (2007). Modifications of uct and sequence-like simulations for monte-carlo go. *IEEE Symposium on Computational Intelligence and Games CIG 2007, 175.*

[Watson and Crick, 1953] Watson, J. and Crick, F. (1953). Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid. *Nature*, 171(4356):737–738.

[Watts and Dodds, 2007] Watts, D. and Dodds, P. (2007). Influentials, networks, and public opinion formation. *Journal of Consumer Research*, 34(4):441–458.

[Watts and Strogatz, 1998] Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442.

[West et al., 2010] West, J. D., Bergstrom, T. C., and Bergstrom, C. T. (2010). The eigenfactor metrics: a network approach to as sessing scholarly journals. *Coll. Res. Lib.*, 71(236). http://www.eigenfactor.org/.

[Wilf, 2002] Wilf, H. S. (2002). *Algorithms and Complexity.* A K Peters/CRC Press, Natick, Mass, 2 edition edition.

[Wilkins et al., 1953] Wilkins, M., Stokes, A., and Wilson, H. (1953). Molecular structure of deoxypentose nucleic acids. *Nature*, 171(4356):738–740.

[Williams, 2011] Williams, L. R. (2011). Markov chains. https://www.cs.unm.edu/ williams/cs530/markov4.pdf.

[Woese et al., 1990] Woese, C., Kandler, O., and Wheelis, M. (1990). Towards a natural system of organisms: proposal for the domains archaea, bacteria, and eucarya. *Proceedings of the National Academy of Sciences*, 87(12):4576–4579.

[Wormatlas, 2013] Wormatlas (2013). http://www.wormatlas.org/.

[Zaller, 1992] Zaller, J. R. (1992). *The Nature and Origins of Mass Opinion.* Cambridge University Press, Cambridge England ; New York, NY, USA.

[Zhirov et al., 2010] Zhirov, A. O., Zhirov, O. V., and Shepelyansky, D. L. (2010). Two-dimensional ranking of wikipedia articles. *The European Physical Journal B*, 77(4):523–531.

[Zipf, 1949] Zipf, G. K. (1949). *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology.* Addison-Wesley, Cambridge, MA.

[Zuo et al., 2012] Zuo, X.-N., Ehmke, R., Mennes, M., Imperati, D., Castellanos, F. X., Sporns, O., and Milham, M. P. (2012). Network centrality in the human functional connectome. *Cerebral Cortex (New York, N.Y.: 1991)*, 22(8):1862–1875.

Abstract

La théorie des réseaux complexes est un domaine récent et important de la recherche qui consiste à étudier divers systèmes naturels ou artificiels d'un point de vue des graphes en considérant une collection d'objets interdépendants. Parmi les différents aspects de la théorie des réseaux complexes, cette thèse se concentre sur l'analyse des propriétés structurelles des réseaux dirigés. L'outil principal utilisé dans ce travail est la méthode de la matrice Google qui est une méthode dérivée de la théorie des chaînes de Markov.

La construction de cette matrice et son lien avec les chaînes de Markov sont expliqués dans le second chapitre et les propriétés spectrales des valeurs propres y sont également discutées. L'accent est mis sur le vecteur propre principal de la matrice (le PageRank). La base du système de ranking donné par le PageRank y est expliquée en détail et illustrée à travers plusieurs exemples dans les chapitres suivants.

Les systèmes considérés ici sont : les séquences d'ADN de quelques espèces animales, le système nerveux du vers *C.elegans* ainsi que l'antique jeu de stratégie sur plateau, le jeu de go. Dans le premier cas nous analysons les propriétés statistiques des chaînes symboliques sous le point de vue des réseaux dirigés et nous proposons une mesure simple de similarité entre les espèces basée sur le PageRank. Dans le second cas nous introduisons le concept du ranking complémentaire (le CheiRank) permettant de caractériser en deux dimensions les réseaux dirigés. Dans le troisième cas nous utilisons les vecteurs propres principaux pour mettre en évidence les coups importants joués lors d'une partie de Go et nous montrons que les vecteurs propres suivants peuvent contenir des informations de communautés de coups.

Ces diverses applications montrent que l'information apportée par le PageRank peut s'avérer utile dans de nombreuses situations différentes afin d'obtenir un aperçu du problème sous un angle différent, qui est l'approche des réseaux dirigés, enrichissant ainsi notre compréhension des systèmes étudiés.

Mots-clés : matrice de Google, PageRank, réseaux dirigés, ranking, centralité, communauté

Abstract

The complex network theory is a recent field of great importance to study various systems under a graph perspective by considering a collection of interdependent objects. Among the different aspects of the complex networks, this thesis is focused on the analysis of structural properties of directed networks. The primary tool used in this work is the Google matrix method which is derived from the Markov chain theory.

The construction of this matrix and its link with Markov chains are explored and the spectral properties of the eigenvalues are discussed with an emphasis on the dominant eigenvalue with its associated eigenvector (PageRank vector). The ranking system given by the PageRank is explained in detail and illustrated through several examples.

The systems considered here are the DNA sequences of some animal species, the neural system of the *C.elegans* worm and the ancient strategy board game : the game of Go. In the first case, the statistical properties of symbolic chains are analyzed through a directed network viewpoint and a similarity measure of species based on PageRank is proposed. In the second case, the complementary ranking system (CheiRank vector) is introduced to provide a two dimensional characterization of the directed networks. In the third case, the dominant eigenvectors are used to highlight the most important moves during a game of Go and it is shown that those eigenvectors contain more information than mere frequency counts of the moves. It is also discussed that eigenvectors other than the dominant ones might contain information about some community structures of moves.

These applications show how the information brought by the PageRank can be useful in various situations to gain some interesting or original insight about the studied system and how it is helping to understand the organization of the underlying directed network.

Keywords : Google matrix, PageRank, directed networks, ranking, centrality, community structure