memoire de DEA

RECENT DUPLICATIONS IN GENOMES: A GRAPH THEORY APPROACH

Nadia Pisanti

DEA INFORMATIQUE FONDAMENTALE ET APPLICATIONS
1997-1998

Université de Marne-la-Vallée
École Nationale des Ponts et Chaussées
École Supérieure d'Ingénieurs en Électronique et Électrotechnique
Université Paris VII - Denis Diderot

Stage realized at the *Atelier de Bioinformatique* of Institute Curie
under the supervision of prof. Alain Viari.

Contents

Introduction

Duplication of genes is studied by biologists mostly because it can be used to guess information about the evolution steps that have taken place in organisms. We will summarize the reasons and the bottom line of such studies in section 1.1.2 after having introduced the necessaries notions of biology in section 1.1.1.

Our data are all the genes of the complete genome of a bacterium, and a distance function between them. Within such a genome, the distance is computed for every pair of genes and thus similarities among them are detected as a family of genes whose distance is smaller than a given threshold or whose distance is considered as *significant* according to a statistical model. The resulting data, depending upon the threshold, can be represented by a graph (section 1.2.2). The usual approaches fix such a threshold and then study the properties of the resulting families of similar genes. This is not our case: we build the graph and study its properties for thresholds that cover all the ranges of connectivity of the graph (that correspond to all the degrees of similarity required for genes to end in the same family). Such a graph is in fact our subject of study, and its properties are investigated with respect to those of a corresponding *random graph* (we will summarize in section 1.2.3 the terms of such a comparison). Our basic idea is that the properties of the graph that models the biological data are interesting wherever they differ from the properties of random graphs. Thus, we compare the biological data with the one obtained through a shuffled version of the genes (this method is already known in the literature, although usually the comparison takes place not at graph level, but rather at sequence level). Next to this, the aim of this work was to find a suitable model of random graphs having properties such that it is actually comparable to the one representing the biological data. After a preliminary investigation (section 1.2.4), we chose a particular model of random graphs that is detailed in chapter 2. This results in a much better model than the basic one, but not yet safer than traditional techniques.

Finally, chapter 3 describes the results of the application of the methods to a bacterial genome, *Mycoplasma genitalium*. After having briefly presented, in section 3.1, some characteristics of the organism, we compare for all the thresholds graphs representing the biological data obtained with all the models we mention in this work, including data resulting from the shuffling of the genome's sequence. We then analyse the behaviour of one particular model for some particular thresholds that resulted in the detection of unusual properties of the graphs representing *Mycoplasma genitalium*. In fact, we computed numerical values for suitable differences between the graphs (section 3.2), and we investigate some unusual data.

Chapter 1

Generalities of the problem

1.1 The biological problem

1.1.1 Some useful biological information

All of us have always heard that genetical information resides in DNA. How does this information make the organisms be the way they are? Proteins are the fundamental agents of life; the human genome, for example, contains something like $100,000$ different proteins. Their properties and their interactions determine organisms. Enzymes are proteins and they are the molecules that cause all the chemical transformations that take place in a cell. It is the DNA that contains all the necessary information to synthesize proteins. Within DNA, such information is grouped into chromosomes, and in each chromosome there can be thousands of *genes*: each gene contains the information for the synthesis of a single protein[1].

DNA can be seen as a sequence of *nucleotides*. The four nucleotides that compose a strand of DNA are: adenine (A), guanine (G), cytosine (C) and thymine (T). Thus, for computer scientists, DNA can be seen just as a string over the alphabet $\{A, C, G, T\}$. Instead, the *primary structure*[2] of a protein is a linear sequence made of a combination of 20 different amino acids. Although that these two sequences are strings on different alphabets, they are related in the sense that a portion of a DNA sequence (the gene) contains the information that determines the sequence of amino acids of the protein encoded by that particular gene. This is the reason for scientists to use the expression *genetic code*: each triplet of nucleotides in the DNA sequence, named *codon*, codifies one of the 20 amino acids.[3]

[1]Actually, there are also genes that encode something different from a protein, i.e., they contain information that is necessary for the event itself of the synthesis of a protein (e.g. tRNA genes).

[2]The first product of protein synthesis.

[3]There are $64 = 4^3$ possible triplets of nucleotides while there are only 20 amino acids.

Mycoplasma genitalium is a bacterium. Such bacteria being *prokaryotes*,[4] each one of them consists of a unique chromosome. This genome has been recently sequenced and consists of 468 genes.

The final aim of our work is to study duplications in this organism. Below we explain why biologists are interested in studying duplications within a genome. A part from any other use we may do of the data gathered about duplications in this genome, we just remark that such a study is one example of an abstraction that could make genomes comparable at a higher than molecular level. In fact, since in recent years more and more complete genomes have been made available, it is important to build models that can help in the comparative investigation of different genomes, possibly even of different species.

1.1.2 Why studying duplications?

As in the cases of other genomes, the sequencing of *Mycoplasma genitalium* and *Mycoplasma pneumoniae* reveals the presence of duplicated genes. The most studied genome so far, the one of the bacterium *Escherichia coli*[5] e.g., revealed a percentage of repeated genes close to 50% ([10, 12]). The topology and amount of duplications in chromosomes are useful information for studying the evolution of organisms. In fact, according to an hypothesis made in the seventies, gene duplication is the predominant mechanism for the evolution of new gene functions: once redundant gene copies are available, mutations may accumulate in one of the copies, thus creating a new gene, while the other one(s) can go on carrying out the original function. Actually, not every duplicated gene gives rise to a new gene function (it can also become what is called a *pseudogene*), but it has also been observed that random inactivation of duplicated copies can change the linkage relationship among active genes, and thus any duplication can still have important evolutionary consequences ([11]).

If genes have duplicated in order to let the genome gain new functions, then we can assume that a classification of them grouping similar sequences

This unavoidable redundancy (for obvious reasons two nucleotides would not have been enough, as $4^2 = 16$ is lower than the number of different amino acids) is explained since there are amino acids that are encoded by several different codons, and there are codons that encode other information such as the beginning and the end of the gene sequence.

[4]The genome of procaryotes is organized in a (possibly repeated) single chromosome. Moreover, such a genome uses to consist in a circular molecule of DNA, and in a cell where there is no nucleus. *Eucaryotes*, instead, are higher order organisms with possibly several chromosomes contained in the nucleus of the cell, isolated by a suitable membrane.

[5]The complete genome of E.coli has become available only very recently, but this is the genome whose sequencing was started first, and then the one whose partial data has mostly been used by biologists.

could reasonably correspond to families of inheritance. Thus, while listing such families, we actually produce sets of genes that share a common origin and that have been recently duplicated to let part of them evolve into new functions. Once we accept this, it is easy to understand why analysing such sets of recently duplicated genes provide interesting data (possibly allowing both to answer questions and, why not?, to formulate new hypotheses) about the way genes evolve, how their ordering and allocation changes along the chromosome, and many other properties that could be observed.

Next to the study of evolution itself, grouping genes that have evolved from the same ancestor can also lead to some other kind of useful results. Once we have classified a set of genes having a common origin (because their sequences share extensive homologies), if we knew well the proteins resulting from a subset of them, then we could infer information about the remaining ones of the family.

We now list some representative studies of evolution that start from an analysis of duplicated genes. In the literature, such analyses are carried out both for eukaryotes and prokaryotes. Most of the methods and hypotheses are valid for both of them, but there are also important differences that have to be taken into account. While studying homologous genes in eukaryotes, one may also be interested in grouping similar genes belonging to different chromosomes. Such comparisons cannot be made in prokaryotes organisms, since they consist of a single chromosome. Moreover, several studies include the detection of similarities among genes of different genomes. In the literature, these two different kind of homologies are named *orthologs* and *paralogs*. The former indicates genes in different species that have evolved from a common ancestral gene by speciation, and the homologies concern (genes that encode for) proteins that have the same function in different genomes. The latter, (the homologies we are interested in), are defined instead as ([12]) *homologous genes in the same organism whose products perform related but not identical functions*. These are the results of duplications.

While comparing gene sequences in eukaryotes, there are many events taking place and that may be accidentally confused, or deliberately considered, as duplications. These and other reasons obviously lead to many more complications when addressing higher order organisms like eukaryotes. The reader should then be aware that, at least for now, our analyses refer to prokaryotes only, and that the similarities we are interested in are only those that can result from duplications or translocations[6] of genes.

[6] Rearrangement of a chromosome in which a segment is moved from one location to another. This is sometimes reciprocal, when one fragment is exchanged for another one.

In [15], e.g., a general model of evolution by gene duplication in eukaryotes is analysed by simulating the evolution of a genetic system, starting from a single gene copy and following by the occurrence of several events with suitably maintained properties. The interest of such a study is to understand the origin of large multigene families with functionally different gene members in higher order organisms. The experimental results of this paper show how copy number, pseudogene number, and other parameters change with time, under different intensities of selection after a certain number of generations.

In [11] the problem of the fate of a recently duplicated gene is addressed: it can either become a pseudogene, or it can mutate in such a way that it gives rise to a new function. The probabilities of these two events are estimated.

Computer simulations, next to analytic methods, are used in [6] to estimate some parameters describing the evolution of a yeast genome. This analysis considers in particular the ordering (both of genes in a same chromosome and different chromosomes in the whole genome) and other topological properties of the genome. On the contrary, we fix our attention on a particular aspect of duplicated genes. In fact, our purpose is to simplify (at least in a first approach) the analysis, only considering non-topological parameters, and then to focus on simpler but important aspects of genes' families. More precisely, we decided to start by studying of the amount of different families, and the number of different genes they contain, which is related to their cardinalities, as these simple data are already relevant from an evolutionary point of view.

An analysis of that kind was performed in [17], after a suitable clustering of genes of several different organisms.[7] Two parameters are taken into account. The first one is S_n, the total number of sequences that appear in clusters of n sequences, and the second is C_n, the number of clusters of n sequences. Here with sequences we mean gene sequences, also known as *Open Reading Frames*. Results of their pairwise alignments and subsequent clustering, show an unexpected regularity in the frequency of duplications:

$$S_n = N/2^{(n+1)}$$

and

$$C_n = N/n2^{(n+1)}$$

where N is the total number of sequences in a given genome and n is 2 or 3, 4, 5 etc. Thus 1/8 of the total number of genes in a given genome is present in clusters of two (and only two) homologous sequences (setting $n = 2$), 1/16 in clusters of three (and only three) homologous sequences (with $n = 3$), and

[7] In that paper, both clusters of paralogous (within the same genome) and orthologous (among different genomes) sequences are built. Here we report only the results concerning paralogous homologies.

so on. This paper is an example of an analysis of genes families, made with the assumption that the families are actually clusters, i.e., equivalence classes. This can only be achieved (as it is actually done in the paper) performing a transitive closure on the similarity relation. Once again our choice here is different: we will not perform any transitive closure on the biological data.

1.2 The graph theory approach

1.2.1 The Levenshtein distance

It is now worth to spend some words on the kind of biological data we will work on. We have all the amino acid sequences corresponding to the genes of the genome we investigate. In order to study homologous genes in a specific genome, it is necessary to choose a notion of similarity. Since, as we already stated, we consider a couple of genes to be similar if the distance between them is smaller than a certain threshold, we need to choose the distance we will use in this pairwise comparison of genes.

The comparison of biosequences has become one of the most important tools for molecular biologists. This explains the huge literature developed in recent years, devoted to sequence comparisons and especially designed for molecules (see, e.g., [7, 14]). These include algorithms and methods for the detection and the measurement of similarities among both nucleotide and amino acid sequences.

In particular, what especially has been developed, are the dynamic programming approaches for the *alignment* of sequences. One observation we can make is that, when comparing two sequences, one does not always know which part of a sequence must correspond to a certain part of the other one (that in general may even have a different length) in order to obtain the highest similarity. This is the reason why we talk about *alignments*: the goal is to seek along the sequences in order to maximize a suitable *score* function or minimize a defined *distance*. For this purpose, dynamic programming algorithms inductively compute, for all possible alignments, the distance (to be minimized, or a score to be maximized) of the prefixes A_i and B_j (resp. of length i and j) of two sequences A and B:

$$d(A_i, B_j) = min/max \begin{cases} d(A_{i-1}, B_{j-1}), & +w(a_i, b_j) \\ d(A_{i-1}, B_j), & +w_{DEL} \\ d(A_i, B_{j-1}), & +w_{INS} \end{cases}$$

where a_i is the $i-th$ element of the sequence A and b_j is the $j-th$ of B, and $w(a_i, b_j)$ indicates the price to be paid for the $a_i \leftarrow b_j$ substitution in the case that a_i and b_j are different amino acids, and the score gained in the case of a matching between them. Moreover, w_{INS} and w_{DEL} are the costs of, respectively, insertion and deletion of an amino acid.

One of the simplest such distances that has been suggested is the *Levenshtein* one. We will use this one to estimate the similarity of genes. In the recurrence formula we have just described, this corresponds to $w_{DEL} = w_{INS} = 1$ and $w(a_i, b_j) = 0$ if $a_i = b_j$, and $= 1$ otherwise. This means that the substitution, insertion and deletion, cost 1 in an alignment. The Levenshtein distance is a *global distance* in the sense that the aim is to entirely align both sequences so that if the lengths of A and B are different, this will cause a larger number of forced insertions or deletions. The internal similarities between one sequence and part of the other one will then be hardly detected.

1.2.2 The graph $\mathcal{G}_\Gamma$

The graph $\mathcal{G}_\Gamma$ is the one that represents, in our study, the biological data. As we have already mentioned, this one is built for several different thresholds covering all the ranges from a graph with almost no edges (i.e., only perfectly identical sequences are detected as similar) to a fully connected graph (i.e., all genes are similar among each other).

Once we know the pairwise Levenshtein distance for all the genes, we construct the graph $\mathcal{G}_\Gamma$ with respect to the threshold Γ. For each threshold Γ we build $\mathcal{G}_\Gamma$ by creating a node for each gene, and then inserting the edge (i, j) if and only if the (Levenshtein) distance between the gene i and the gene j is smaller than or equal to Γ, that is

$$(i, j) \in \mathcal{G}_\Gamma \Leftrightarrow d_L(i, j) \leq \Gamma$$

Since the distance between i and j is the same as the one between j and i, such edges are not directed and each one of them represents the similarity property of the unordered pair (i, j). We consider 100 different thresholds that filter the *most similar* pairs of genes, namely from 0.00 to 0.99 such that the former corresponds to not allowing any difference and the latter corresponds to allowing the maximum distance we detected between pairs of genes. Of course, the interesting cases are not these extreme ones.

We will denote by *Levenshtein graphs* all graphs built in this way where the Levenshtein distance is used and a threshold is fixed. Thus, our graphs $\mathcal{G}_\Gamma$ are indeed Levenshtein graphs. The novelty of our approach is that we operate on these graphs directly, and not on the distance. That is, we study the biological data at a graph level skipping the sequence one. In fact, in the literature, in order to study properties of biosequences, random biosequences are generated and compared to the biological data. What we try to do here is something truly different, as our randomly generated data is not at the

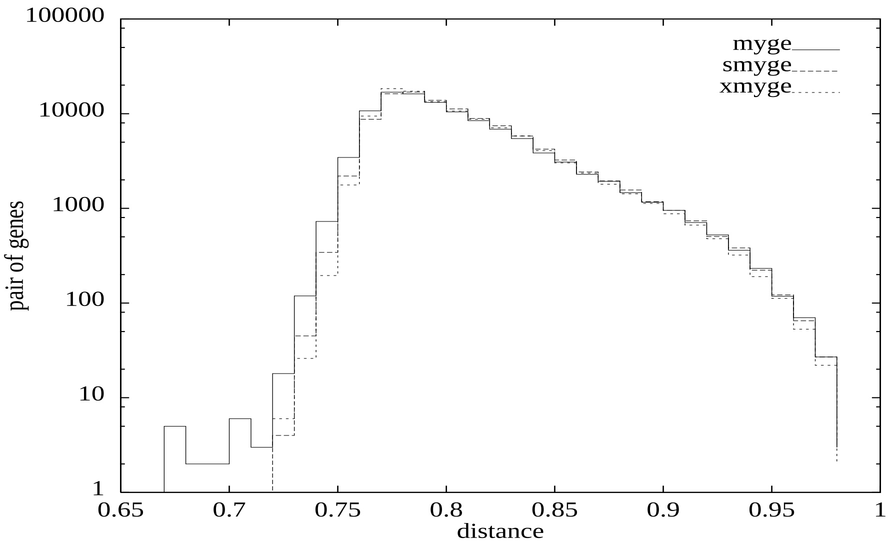


Figure 1.1: Distance distributions in *Mycoplasma genitalium*; myge is the distribution of the organism, smyge is the one obtained from the shuffling within genes, and xmyge is the one from the shuffling of the complete genome.

level of sequences, but directly at the level of graphs. Since the use of random sequences is, among pre-existing methods, the most similar to our idea of comparing with random graphs, we will use this known method to test the behaviour of our models. In fact, since our final aim is to detect cases where the biological data differ from the random one, it would make no sense to compare a new random model directly with the biological data, as we could not know whether possible differences are due to the significance of the biological data (which is what we are looking for), or to a bad choice of the model (which is what we want to avoid), resulting in graphs that cannot be compared at all. For this reason, the first test we do of our random model is done by comparing it with Levenshtein graphs built from random sequences, as these are models that are known to be comparable to the biological data. More precisely, given the genome to be analysed, we generate two versions of shuffled sequences. In the first version, the sequence of each gene is randomly shuffled. In the second version, all the sequences of the genes in the genome are shuffled, and then the original number of genes (with their original sizes) is generated. In this way, we do have random data comparable to the one we are studying, and we will use this to test our new approach before possibly applying it. This is confirmed by the histograms of figures 1.1 and 1.2 where the distribution of all distances among genes is shown for the biological data and for the two shuffled versions. Figure 1.1 refers to *Mycoplasma genitalium*, while figure 1.2 concerns *Mycoplasma pneumoniae*.[8]

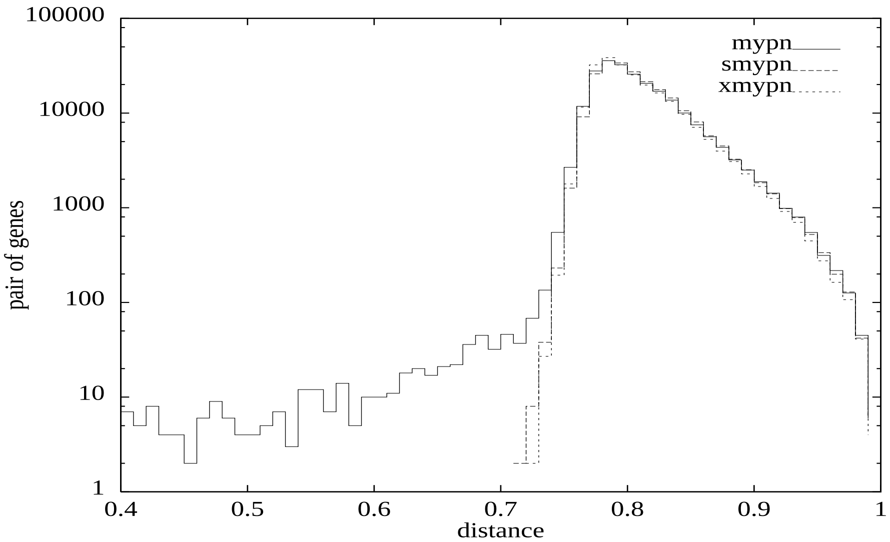


Figure 1.2: Distance distributions in *Mycoplasma pneumoniae*; mypn is the distribution of the organism, smypn is the one obtained from the shuffling within genes, and xmypn is the one from the shuffling of the complete genome.

It is evident that things do not change much between the two shuffling techniques. What is also evident, especially in *Mycoplasma pneumoniae*, is the presence of a few more small distances in the unshuffled biological data. Seen as a preliminary use of the *comparing-to-random* approach, this shows how biologically significant things happen for small values of the threshold where the biology shows more similarities than the expected. In the next chapter we will see the reasons of this.

Starting from the distance distributions showed in figures 1.1 and 1.2, we will obtain, for different values of Γ, very different graphs, that are mostly not at all comparable between themselves. Thus, what we will compare to $\mathcal{G}_\Gamma$ is a suitable graph, with a number of edges similar to the one of $\mathcal{G}_\Gamma$, that is depending on Γ. In order to have an idea of how in Levenshtein graphs the number of edges changes with respect to the threshold, figure 1.3 shows

[8]This is another bacterium belonging to the Mycoplasma family and with characteristics analogous to *Mycoplasma genitalium* that are the reasons for which these are the only two mycoplasmas sequenced so far. It has 676 genes.

the number of resulting edges in the graphs for all values of the threshold in *Mycoplasma genitalium*.

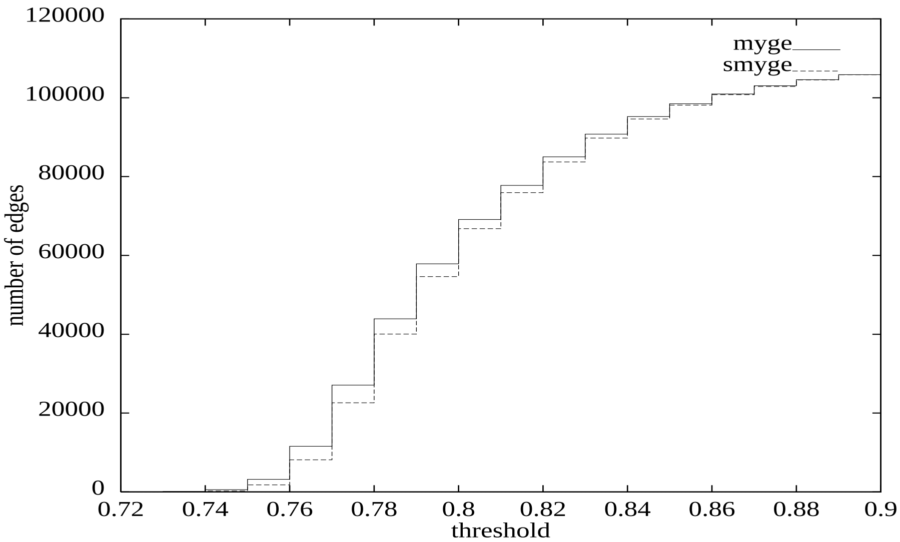


Figure 1.3: Number of edges in Levenshtein graphs for all possible values of the threshold from 0.00 to 1.00. The curve myge refers to the pure biological data, while smyge refers to the shuffled sequences.

1.2.3 Comparing with random graphs

Before showing how our random graphs are built, we describe which are the properties of $\mathcal{G}_\Gamma$ that we want to compare with the ones of the corresponding random graph. Returning to the original biological problem, there are several graph properties that reflect interesting characteristics of genes. An initial important one is the degree of the nodes. The degree of node i of $\mathcal{G}_\Gamma$ corresponds to the number of genes that are similar to the $i-th$ gene. Concerning the degrees of nodes, we will take the following parameters into account:

Number of nodes: Of course this value is always the same for all the $\mathcal{G}_\Gamma$ representing data of a same genome, as it corresponds to the number of genes. Hence, this will also be the number of nodes of the random graphs we will use for the comparison. In the following we will denote this value by n.

Number of edges: This is an important data of the $\mathcal{G}_\Gamma$ which we will use for the generation of the corresponding random graph. In fact, only once Γ is known and $\mathcal{G}_\Gamma$ built, we do know how many edges we need for the random graph (sometimes only approximatively). In the sequel we will denote this value by m. At this point we need to report a choice we made concerning loops of the graphs. Both in the graph representing the biological data and in the random model, we do have all edges (i, i), because obviously every gene is similar to itself (whatever the threshold). But since we are eventually interested in the *differences* between the biological and the random data only, we will not consider loops in the statistics to be compared, as they will never result in interesting information. This choice will involve also the values of all the parameters that follow.

Mean degree: It is the mean value of the degrees of all nodes.

Variance of degree: This is the variance of the degree.

Degree distribution: This is actually a histogram that, for each possible value k of the degree, shows how many nodes have degree k.

Another set of parameters concerns the lengths of paths in the graph. Given $\mathcal{G}_\Gamma$, we compute the minimal paths among all pair of nodes, i.e., the paths having minimal number of edges that go from one node to another. Since a minimal path from i to j is the reverse of a minimal one from j to i, (and in particular it has the same length), in our statistics we only count once the path from one node to the other ignoring the reverse one. For the computation of all paths of the graph we used the Floyd algorithm ([2]), which has time complexity $O(n^3)$. This is the most time consuming for the computation of the parameters, as the degree information can be computed more quickly. The parameters we will consider are the following:

Number of minimal paths: The total number of minimal paths that appear in the graph. It obviously corresponds to the number of all possible pairs of nodes (that is $n(n-1)/2$, as we count only one path for each pair of nodes if the minimal path is not unique) when the graph represents a single connected component, and is lower than that otherwise.

Mean path length: The mean value of all minimal path lengths.

Variance of path length: The variance of all minimal path lengths.

Path lengths distribution: This is another histogram that shows, for each possible value of path length, how many pairs of nodes are connected by a minimum path of such length.

Number of connected components: Finally, this is the number of connected components.

These are the values we computed for the graphs $\mathcal{G}_\Gamma$ for all values of Γ. For each one of them, we then generated the random graph with (approximately) the same number m of edges, and also for this latter the values of these parameters are computed and compared to those of $\mathcal{G}_\Gamma$. The two histograms of the degree distribution and paths length distribution include all the other information, and for this reason our attention focused on them. Our first attempt was to use a simple random graph whose only requirements were that they should have n nodes and m edges. The results of this comparison are described in section 1.2.4. For a reason that will soon be clear to the reader, we successively modified this simple model, and conceived then a new better one whose description and comparison results are described in chapter 2.

1.2.4 First results for a basic model of random graphs

There is a huge literature on random graps because of their usefulness in testing the behaviour of graph algorithms. The two main models in the literature (see [3] for a good review) are $\mathcal{G}(n,m)$ and $\mathcal{G}(n,p)$. The first one consists of all graphs having n nodes and m edges, in which all the graphs have the same probability. The second model, $\mathcal{G}(n,p)$ with $0<p<1$, contains all graphs having n nodes and in which each of the possible edges may appear independently with probability p. Let N be the total number of possible edges that can appear in a graph having n nodes. Clearly, $N=n(n-1)/2$ in undirected graphs (not considering loops). It is quite intuitive that, as concerns most of graph properties, these two models are practically interchangeable provided $m \approx Np$. We will use this opportunity when we generate our desired random graphs. In fact, once we have built $\mathcal{G}_\Gamma$, as we described in section 1.2.3, we compute some statistical properties among which the number of edges m, and then it is clear that the required random graphs is the one in $\mathcal{G}(n,m)$ with such an m. But a random generator is much easier to conceive for $\mathcal{G}(n,p)$ and, moreover, many interesting results about properties of nodes degrees of a random graph are known for $\mathcal{G}(n,p)$ only. Thus, once we have the number of edges of the *biological graph* $\mathcal{G}_\Gamma$, we compute the corresponding $p = m/N$, and we generate the random graph using the model $\mathcal{G}(n,p)$ instead of the $\mathcal{G}(n,m)$ one.

Our first attempt was to simply use a random graph belonging to $\mathcal{G}(n,m)$, where n and m are the same as those of the graph $\mathcal{G}_\Gamma$. Thus, we compared the degree distribution of such a simple model with the one of the corresponding Levenshtein graphs. Here we used a result of the theory of random graphs: in the model $\mathcal{G}(n,p)$, the degree distribution follows a binomial law with parameters $n-1$ and p ([3]). The binomial law is one of the simplest ones in

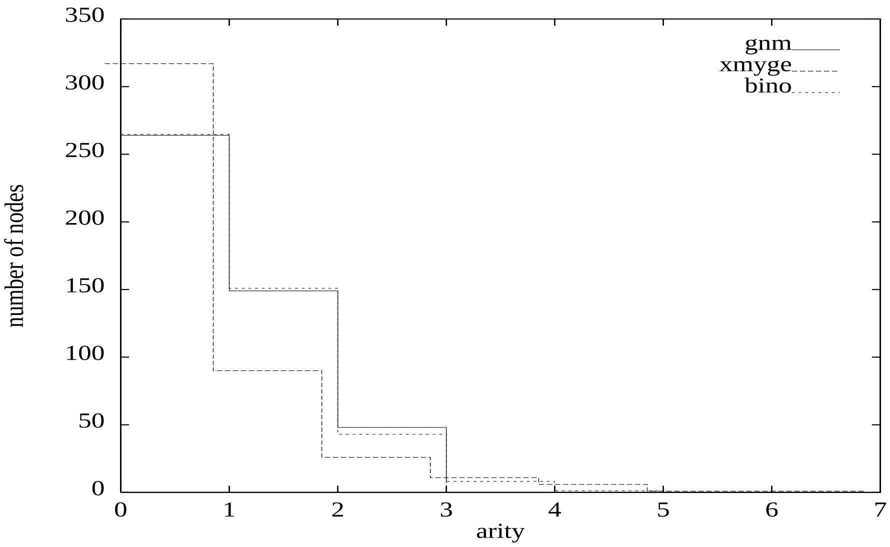


Figure 1.4: Degree distributions for a random graph of the model $\mathcal{G}(468, m=133)$ (curve gnm) and for a shuffled *Mycoplasma genitalium* (curve xmyge) with a threshold 0.74. Both the graphs have 133 edges. Moreover, the curve bino shows the binomial distribution of parameters $n-1 = 467$ and $p=133/N$. The histograms represent, for all k, the number of nodes of degree k.

probability theory, and according to it, the just mentioned result gives the following formula for the probability of a node to have degree k:

$$Prob(degree = k) = \binom{n-1}{k} p^k (1-p)^{n-1-k}$$

Using this formula (and [19] for its implementation), we generated the histogram of the theoretical degree distribution of a random graph having n nodes and m edges (here we changed from $\mathcal{G}(n, m)$ to $\mathcal{G}(n, p = m/N)$, as we do know how many edges we want the graph to contain, but the distribution function we have depends on p and not on m).

We compared the obtained distribution with the one computed on the Levenshtein graphs built from the shuffled sequences. This is achieved by first computing the distribution for a shuffled version of *Mycoplasma genitalium* for a given threshold, and then computing the theoretical distribution of a random graph having the same number of edges, together with the corresponding binomial distribution. In figure 1.4 the curves for a threshold $\Gamma = 0.74$ are

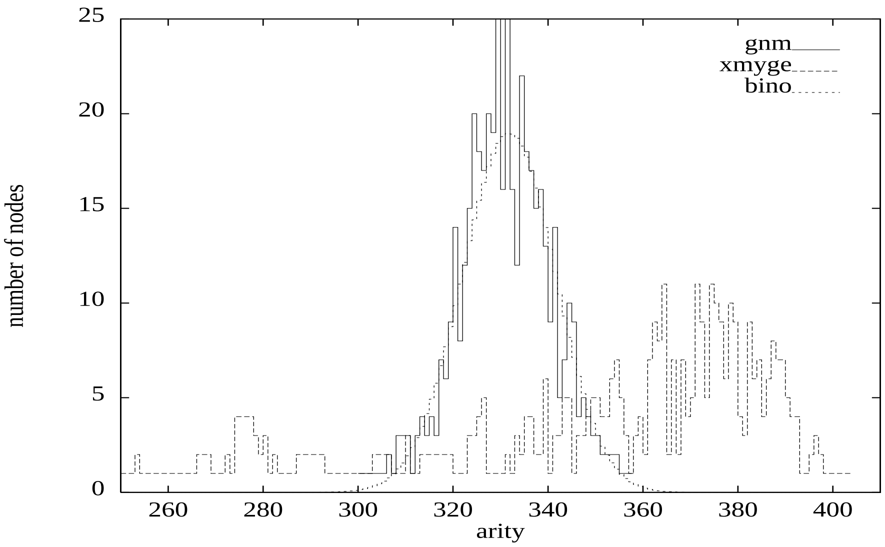


Figure 1.5: Degree distributions for a random graph of the model $\mathcal{G}(468, m = 77323)$ (curve gnm) and for a shuffled *Mycoplasma genitalium* (curve xmyge) with a threshold $\Gamma = 0.81$. They both have 77323 edges. Moreover, the curve bino shows the binomial distribution of parameters $n-1 = 467$ and $p = 77323/N$.

shown, while in figure 1.5 we have $\Gamma = 0.81$.

In figure 1.5, it is clear that the behaviour of the random graph $\mathcal{G}(n, p = m/N)$ is too different from the corresponding Levenshtein graph (that is, having the same numeber of edges) built from the shuffled sequences. Indeed, for the highest value of the threshold (see figure 1.5) it is clear that the degree distribution of the Levenshtein graphs does not even look like a binomial one. Increasing the threshold (and then the number of edges), the curve of the binomial distribution (as well as the one of the random graph) simply moves to the right, detecting higher degrees for the nodes, while the shuffled data maintains lower degrees and then the curve assumes a different shape, showing maximum values for higher degrees. These results make the simple model $\mathcal{G}(n, m)$ of random graphs not suitable for our aim, as it is so different from the shuffled biology that it cannot be used for a comparison. This of course has a reason that resides in the nature of Levenshtein graphs and we will explain it in the next chapter.

Chapter 2

The model $\mathcal{G}(n, m, f)$

2.1 Why $\mathcal{G}(n, m)$ does not work

2.1.1 Levenshtein graphs are not random

We have seen in section 1.2.4 how a simple random graph having the same number of nodes as the Levenshtein graph built from the shuffled biological data does not behave at all like this latter. It seems that in the shuffled data, once the threshold is increased, there remain nodes with low degrees that do not appear anymore in the random graph. The reason for this resides in the nature of the Levenshtein graphs: in fact, if our random model has a fixed number of edges, they are spread evenly in the graph because the probability p of any of the edges is assumed to be independent from the presence of any other edge. This does not hold in Levenshtein graphs. In fact, assuming that two sequences i and j are similar (i.e., the edge (i, j) appears in the graph), and that j is also similar to k (i.e., the edge (j, k) is also there), then a similarity between i and k has a quite high probability to appear, at least higher that the average (that would correspond to the presence of the edge (i, k) in the graph), while the random graph gives this event the same probability as any other edge. This property appears in the graph built from the shuffled sequences, and appears even more does in the $\mathcal{G}_\Gamma$ built from the pure (not shuffled) biological data, since we know that duplications did take place.

The model of random graphs must hence be modified in order to resemble Levenshtein graphs more closely so as to be comparable to them. What is the nature of the modification we need? The graphs derived from biological sequences have the property that some edges are more probable than others. This corresponds to a kind of (partial) transitive closure of the similarity relation: (i, j) and (j, k) imply a high expectation for (i, k) to be present as well. This is the reason for our claim that Levenshtein graphs differ from random graphs in the sense that they present a certain degree of a partial transitive closure for edges. We explicitly mention it is a *partial* closure because a total closure would affect the data too much. We will see in section 2.2 that a certain degree of closure is necessary to approximate similarities to the biological data.

2.1.2 Partial transitive closure

In order to address transitive closure properties of Levenshtein graphs, it is worthwhile to discuss some theories of molecular evolution and corresponding analyses of the Levenshtein distance. For this purpose we remind the reader that a distance has the following properties:

- *Nonnegative property:* $d(i,j) \geq 0$ for all i and j.
- *Zero property:* $d(i,j) = 0$ if and only if $i = j$.
- *Symmetry property:* $d(i,j) = d(j,i)$ for all i and j.
- *Triangle inequality:* $d(i,k) \leq d(i,j) + d(j,k)$ for all i,j and k.

Where $d(i,j)$ indicates the distance between i and j. The relation we are interested in (and that corresponds to the existence of edges in Levenshtein graphs) is the following:

$$iRj \text{ if and only if } d_L(i,j) \leq \Gamma$$

where Γ is the threshold and $d_L()$ denotes the Levenshtein distance. The metric properties we just listed do not imply any form of transitive closure of the relation, but another class of metrics with a stronger version of the triangle inequality property does: a distance is said to be an *ultrametric* if and only if it has the following properties:

- *Nonnegative property:* $d(i,j) \geq 0$ for all i and j.
- *Zero property:* $d(i,j) = 0 \Leftrightarrow i = j$.
- *Symmetry property:* $d(i,j) = d(j,i)$ for all i and j.
- *Ultra triangle inequality:* $d(i,j) \leq max\{d(i,k), d(j,k)\}$ for all i,j and k.

It is easy to see that ultrametrics lead the relation R to be transitively closed. In fact, we have

$iRj \Leftrightarrow d(i,j) \leq \Gamma$ and $jRk \Leftrightarrow d(j,k) \leq \Gamma$.

Since $d(i,k) \leq max\{d(i,h), d(k,h)\}$ for all h, in particular we have for $h = j$ that

$d(i,k) \leq max\{d(i,j), d(k,j)\} \leq \Gamma$, thus implying iRk.

We have thus proved that iRj and jRk implies iRk, i.e., the definition of transitive closure.

Our aim is now to show that the Levenshtein distance partially behaves as an ultrametric. As explained in section 1.1.2, genes duplicate during evolution and the copies then mutate successively in different ways so that new proteins become available to the organism. If mutations always took place in different amino acids, then the number of mutations could easily be counted: it would be the distance in relation to the ancestor sequence. But since mutations can happen more than once in the same amino acid, a more recent mutation could overwrite an older one and thus deleting its trace. In the molecular evolution literature, this effect is named *saturation* and it is assumed that, for the first 20% of the total number of differences that a sequence cumulates with respect to its ancestor, mutations do not overwrite themselves; after that the number of such differences is linear with respect to the number of mutations (and hence to the time if we assume a constant mutation rate).

If we go back to the Levenshtein distance among genes of a same organism, we observe that, when this linearity property holds, the distance is an ultrametric (see below for a brief explanation of this) and then our relation is transitive. But since this linearity property holds only up to a certain point, the result is something we assumed to be a partial transitive closure only. This is the biological explanation of the model we supposed to be a better candidate for the comparison with Levenshtein graphs, and that we will detail in the next sections.

We just stated that when the number of differences among genes is proportional to the number of mutations (which itself is proportional to time), then the ultra triangle property holds for the Levenshtein distance. In short, the reason is the following: if we imagine the classical representation of evolution as a tree where each node is the ancestor gene of its sons, then the distance of two nodes appearing at the same level (an organism possesses at any time only genes that appear at the same level of such a tree) is equal to the sum of their distances to the lowest common ancestor, which is also the maximum distance that any other node of the same level may have to any of the two of them.

2.1.3 $\mathcal{G}(n, m, f)$

Once the partial transitive closure of Levenshtein graphs is assumed, we conceived a new model of random graphs that reflects this property. We say that a *transitive closure of degree* f of a graph G is the graph G^f whose set of nodes is the same as that of G, and the set of edges is the one of G plus all edges linking nodes that in G were connected by a path of length at most f. For example, for $f=2$, we would add to G all edges (i,k) where in G we had the edges (i,j) and (j,k), but not (i,k).

Hence, our new model is indeed $\mathcal{G}(n,m,f)$; a graph belonging to this set is a random graph having n nodes, m edges, and having been transitively closed with a degree f. We stress that m is the number of edges obtained *after* the partial transitive closure applied to a *normal* random graph having a suitable number m_0 of edges. For this purpose, we had to generate graphs in $\mathcal{G}(n,m,f)$ using a trick. In fact as one could expect, once n, m and f are known, the value of m_0 such that the f*–closure* of $\mathcal{G}(n,m_0)$ has m edges needs to be captured, as otherwise we would be obliged to find it with an expensive dichotomy (the closure algorithm has a time cost of $O(fn^2)$). We then used the following result which holds for graphs in $\mathcal{G}(n,p)$:

$$P_{n,p}(k) \approx 1-(1-p^k(1-p)^{((k+1)(k-2)/2)}(k-1)!)^{\binom{h-1}{n-2}},$$

where $P_{n,p}(k)$ is the probability that any two distincts nodes are connected by at least one path of length k. From this formula, which holds for $k \geq 2$ (for $k=1$ we have trivially $P_{n,p}(1)=p$), we obtain the following:

$$P_{n,p}^{\mathrm{MIN}}(k) \approx P_{n,p}(k)\prod_{i=1}^{k-1}(1-P_{n,p}(i)),$$

where $P_{n,p}^{\mathrm{MIN}}(k)$ denotes the probability that the minimal path is of length k. It is clear that, for $p_0=m_0/N$ and $p=m/N$ (where m is the number of edges after the $f-$*closure* and m_0 the one before), we have;

$$p \approx \sum_{h=1}^{f} P_{n,p_0}^{\mathrm{MIN}}(h).$$

Using this property, our algorithm can reach the requested m_0 in just one step and then the suitable graph of $\mathcal{G}(n,m,f)$ can be obtained by the $f-$*closure* of one in $\mathcal{G}(n,m_0)$.

The result is that what we compare with the Levenshtein graph is now a random graph having the same number of nodes and edges, but moreover partially transitively closed. As we will see, the results are better than the basic model $\mathcal{G}(n,m)$, for degree statistics as well as for path ones. Regarding these last ones, we have to say that not every parameter among those we listed in section 1.2.3 is equally significant. The number of minimal paths, e.g., does not change in a graph once we add edges among nodes that already were connected (by a path of f or less than f edges), while this is what we do when performing the partial closure. The same holds for the number of connected components and the reasons are analogous. Consequently, in a graph G of the set $\mathcal{G}(n,m,f)$, these values are the same as those of a random graph of

$\mathcal{G}(n, m_0)$ whose f−*closure* results in G. All the other parameters in the path statistics, mostly regarding the lengths of the paths, however are definitely significant.

2.2 Behaviour of the new model

In this section we will show results about the model $\mathcal{G}(n, m, f)$. In section 2.2.1 everything concerning degree statistics will be reported, while in section 2.2.2 we will present path statistics. In both sections, histograms are showed for three values of the threshold:[1] 0.75, 0.81, and 0.86 that well represent the behaviour. Moreover, data is computed, for random graphs, both for $f = 2$ and $f = 3$.

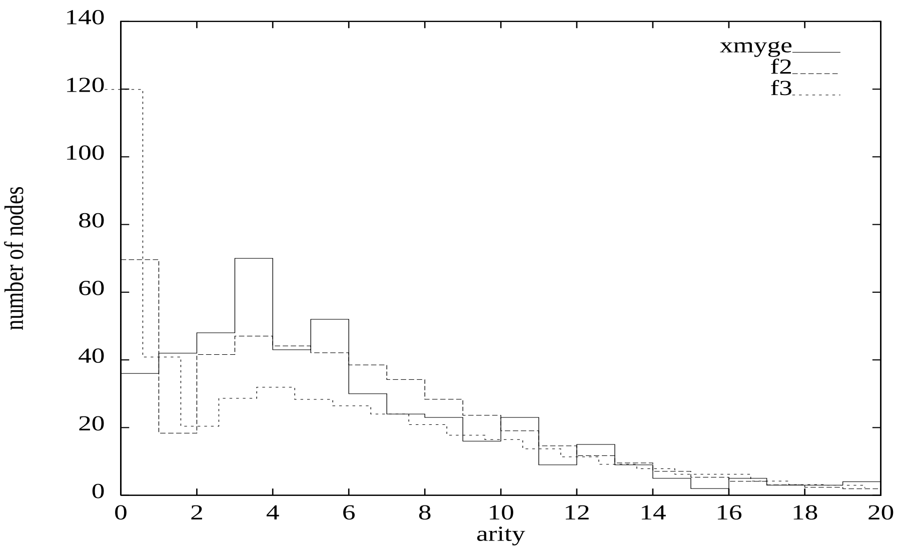


Figure 2.1: Degree distributions. The curve xmyge refers to the graph built for the shuffled sequences (1287 edges), while f2 and f3 are the distributions of the corresponding graphs in $\mathcal{G}(n, m, 2)$ and in $\mathcal{G}(n, m, 3)$, respectively.

Since graphs in $\mathcal{G}(n, m, f)$ are randomly generated, in order to guarantee the significance of the data, we generated 100 graphs for each case. The statistic we report in the next sections, are then simply the average of all those

[1]Respectively 1287, 77323, and 101431 edges for the strongly shuffled version (xmyge) of the genome of *Mycoplasma genitalium*. The requested number of edges for the graphs in $\mathcal{G}(n, m, f)$ are these ones, but an approximation of 5% will be allowed.

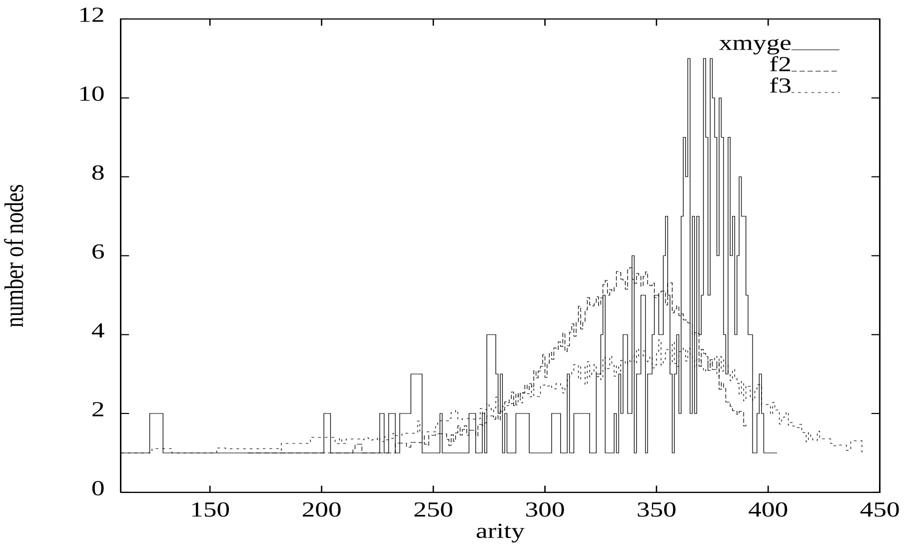


Figure 2.2: Degree distributions. The curve xmyge refers to the graph built for the shuffled sequences (77323 edges), while f2 and f3 are the distributions of graphs in $\mathcal{G}(n, m, 2)$ and in $\mathcal{G}(n, m, 3)$.

obtained with these generations (and in particular the average number of edges will be very close to the requested one). Moreover, for these 100 representants of $\mathcal{G}(n, m, f)$, we compute the variance so that we will understand what we can expect to observe when working with generations of random graphs.

2.2.1 Degree statistics

Figures 2.1, 2.2 and 2.3 represent histograms of the degree distributions for three different thresholds. All of them show the distributions for the shuffled genome and the corresponding graphs in $\mathcal{G}(n, m, 2)$ and $\mathcal{G}(n, m, 3)$ having the same number of edges. Figure 2.1 refers to graphs having all 1287 edges, while in figure 2.2 graphs have 77323 edges, and finally graphs whose degree distributions are shown in figure 2.3 have 101431 edges (fully saturated graphs will have $468 * 467/2 = 109278$ edges).

It is immediately evident that the curves related to the graphs in $\mathcal{G}(n, m, 3)$ are further away from the biology than those of $\mathcal{G}(n, m, 2)$ are. This is due to a phenomenon that is also causes an undesired behaviour of $\mathcal{G}(n, m, 2)$. The problem is that, in order to obtain a number m of edges, the closure is performed on a random graph in $\mathcal{G}(n, m_0)$ with m_0 considerably smaller than m,

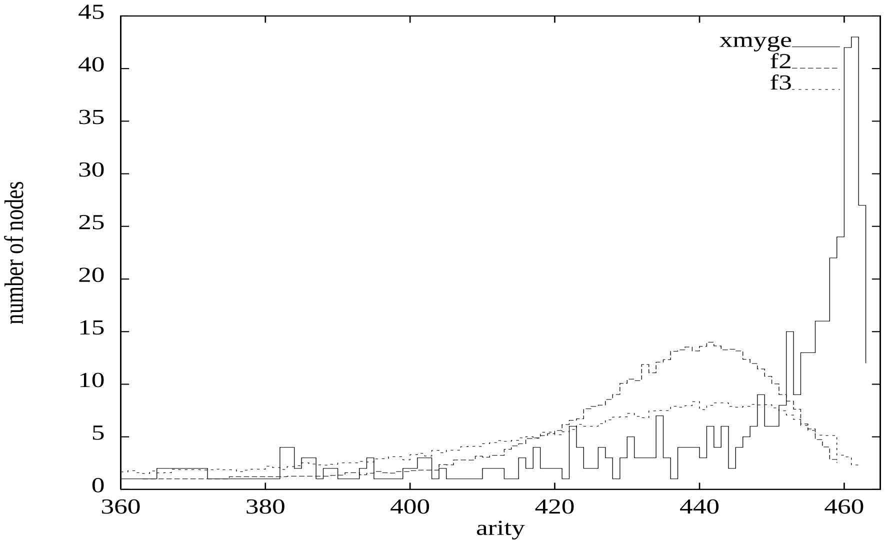


Figure 2.3: Degree distributions. The curve xmyge refers to the graph built for the shuffled sequences (101431 edges), while f2 and f3 are the distributions of graphs in $\mathcal{G}(n, m, 2)$ and $\mathcal{G}(n, m, 3)$.

especially for a high m and above all especially for a closure of degree 3. With such a low value of m_0, it is easy for some nodes not to be connected with any other one and the transitive closure cannot change this. In section 2.2.2 data about the model $\mathcal{G}(n, m, 3)$ is still reported, so that we will also see its behaviour concerning paths, but for the aforementioned reasons, we will then abandon the closure of degree 3.

When computing degree statistics for $\mathcal{G}(n, m, f)$, we have presented results that are the average over 100 different random generations of graphs, but actually for degrees the variance appeared to be very low and therefore in this case even less random generations would have already given significant results.

2.2.2 Path statistics

Similarly to section 2.2.1, figures 2.4, 2.5 and 2.6 represent histograms of the path distributions for the three different thresholds. In each one we give the distribution for the shuffled genome and those of the corresponding graphs in $\mathcal{G}(n, m, 2)$ and $\mathcal{G}(n, m, 3)$. Figure 2.4 refers to graphs having all 1287 edges while in figure 2.5 graphs have 77323 edges. Finally graphs whose path distri-

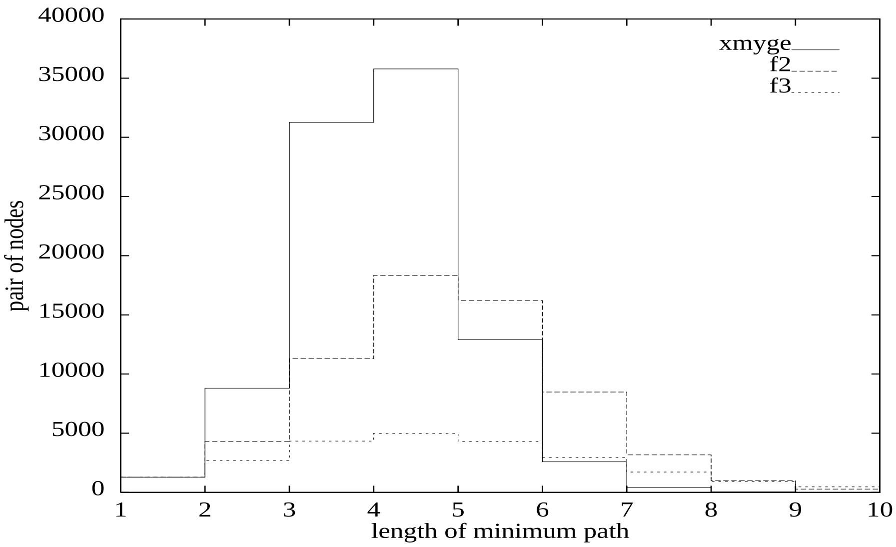


Figure 2.4: Path distributions. The curve xmyge refers to the graph built for the shuffled sequences (1287 edges), while f2 and f3 are the distributions of, respectively, the corresponding graphs in $\mathcal{G}(n, m, 2)$ and $\mathcal{G}(n, m, 3)$.

butions are shown in figure 2.6 have 101431 edges.

While computing data for the random graphs belonging to $\mathcal{G}(n, m, 2)$ and $\mathcal{G}(n, m, 3)$, we also computed the variance resulting from 100 generations of graphs. This was much higher for all the statistics regarding paths. This makes expensive or dangerous the use of random generators for a comparison with biology. Consequently, the generation of a big number (> 100) of different random graphs appeared to be here a necessary task.

Thus, the knowledge of the probability results for random graphs plays an important role as it would avoid the necessity of generating several random graphs. In fact, if we knew the statistics about path distributions, we could avoid generating random graphs and then computing paths (which is also quite expensive in time) and just compare biological graphs with expected values of the statistics. We do have such statistics for degree distributions in $\mathcal{G}(n, p)$, but not yet for $\mathcal{G}(n, m, f)$, and this is the reason for the multi-generation of random data in this work.

One can easily see that the partially closed random graphs behave better than the simple random graphs, but despite encouraging results concerning

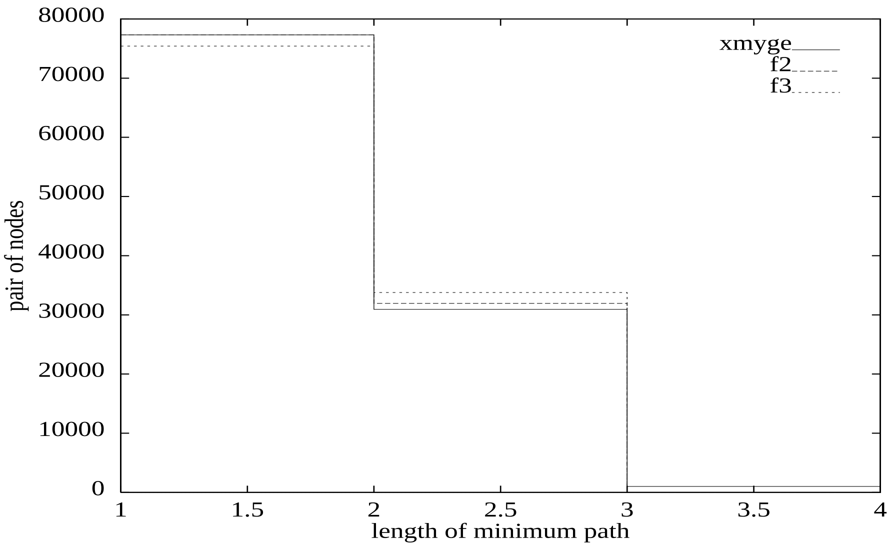


Figure 2.5: Path distributions. The curve xmyge refers to the graph built for the shuffled sequences (77323 edges), while f2 and f3 are the distributions of the corresponding graphs in $\mathcal{G}(n, m, 2)$ and $\mathcal{G}(n, m, 3)$.

path statistics for the two highest values of the threshold, they in general still differ too much from Levenshtein graphs. The reason was already mentioned in section 2.2.1: in order to obtain the desired number of edges after the closure we work with graphs that are disconnected before, and thus also after, the closure. This strongly affects the data related to $\mathcal{G}(n, m, 3)$, but also that of $\mathcal{G}(n, m, 2)$. We then tried to correct this by conceiving a new model. Actually, the new model is a soft version of $\mathcal{G}(n, m, 2)$. It takes as an input, instead of f, a parameter α, $(0 \leq \alpha \leq 1)$, which indicates the probability with which the transitive closure takes place among nodes. Details and results appear in the next section.

2.3 The model $\mathcal{G}(n, m, \alpha+1)$

This new model builds graphs having n nodes, m edges, and transitively closed with a degree $1 \leq \alpha+1 \leq 2$. In fact, these graphs are built similarly to those in $\mathcal{G}(n, m, 2)$ except that in the closure, once we have the edges (i, j) and (j, k), we add the edge (i, k) only with a probability α, unless i and k were already connected in $\mathcal{G}(n, m)$.

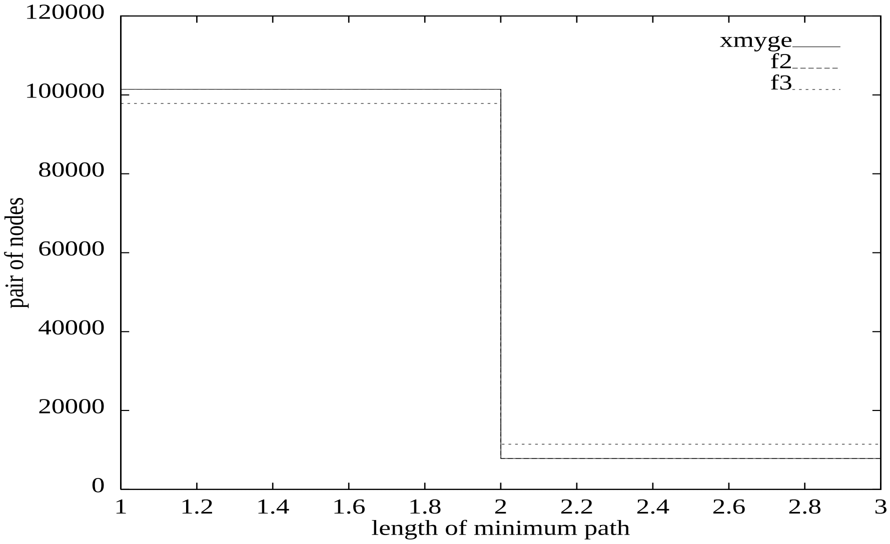


Figure 2.6: Path distributions. The curve xmyge refers to the graph built for the shuffled sequences (101431 edges), while f2 and f3 are the distributions of the graphs in $\mathcal{G}(n, m, 2)$ and $\mathcal{G}(n, m, 3)$.

It is clear that for $\alpha = 0$ we obtain simple graphs in $\mathcal{G}(n, m)$ (no closure at all has taken place), while for $\alpha = 1$ we are back to the model $\mathcal{G}(n, m, 2)$.

With such a modification, we can handle a degree of closure lower than 2 and thus operate this closure starting from a graph with a higher number of edges than in the case of $\mathcal{G}(n, m, f)$, avoiding partly the problem of unsolvable disconnection of nodes.

In figures 2.7, 2.8, and 2.9 we show the distributions (both for degrees and paths) for such graphs where $\alpha = 0.5$, together with (for a comparison) the already known curves obtained with $\mathcal{G}(n, m, 2)$ and the strongly shuffled genome. Again, we report histograms for 1287 edges (figure 2.7) in the shuffled genome and corresponding random graphs, 77323 edges (figure 2.8) and finally for 101431 edges (figure 2.9).

It is evident that the model $\mathcal{G}(n, m, 1+0.5)$ is better than $\mathcal{G}(n, m, 2)$ and goes in the good direction. But, unfortunately, it is not good enough to be used for a safe application to the true (not shuffled) biological data. In fact, the graph built from the shuffled genomes are still quite better than our models of random graphs and there is no way to reach its values (especially for

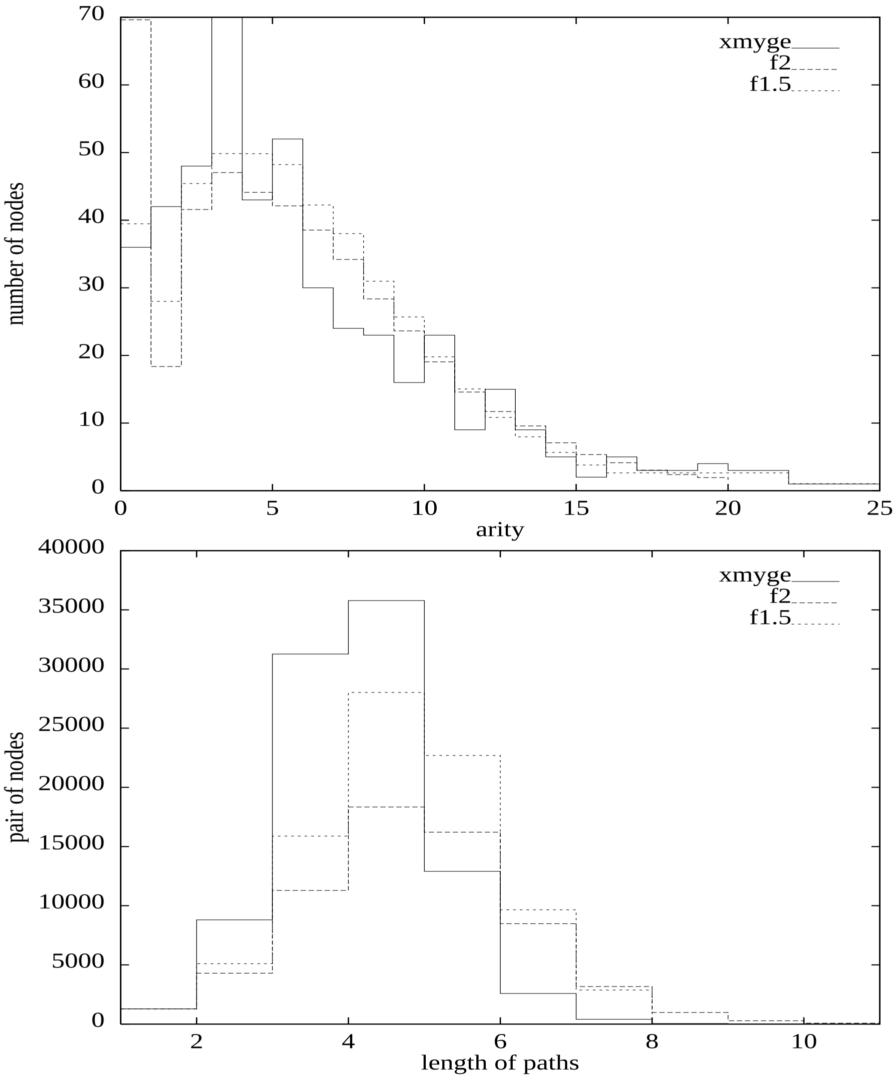


Figure 2.7: Degree and (length of) path distributions (respectively at the top and at the bottom). The curve xmyge refers to the graph built for the shuffled sequences (1287 edges), while f2 and f1.5 are the distributions of, respectively, the corresponding graphs in $\mathcal{G}(n, m, 2)$ and $\mathcal{G}(n, m, 1+0.5)$.

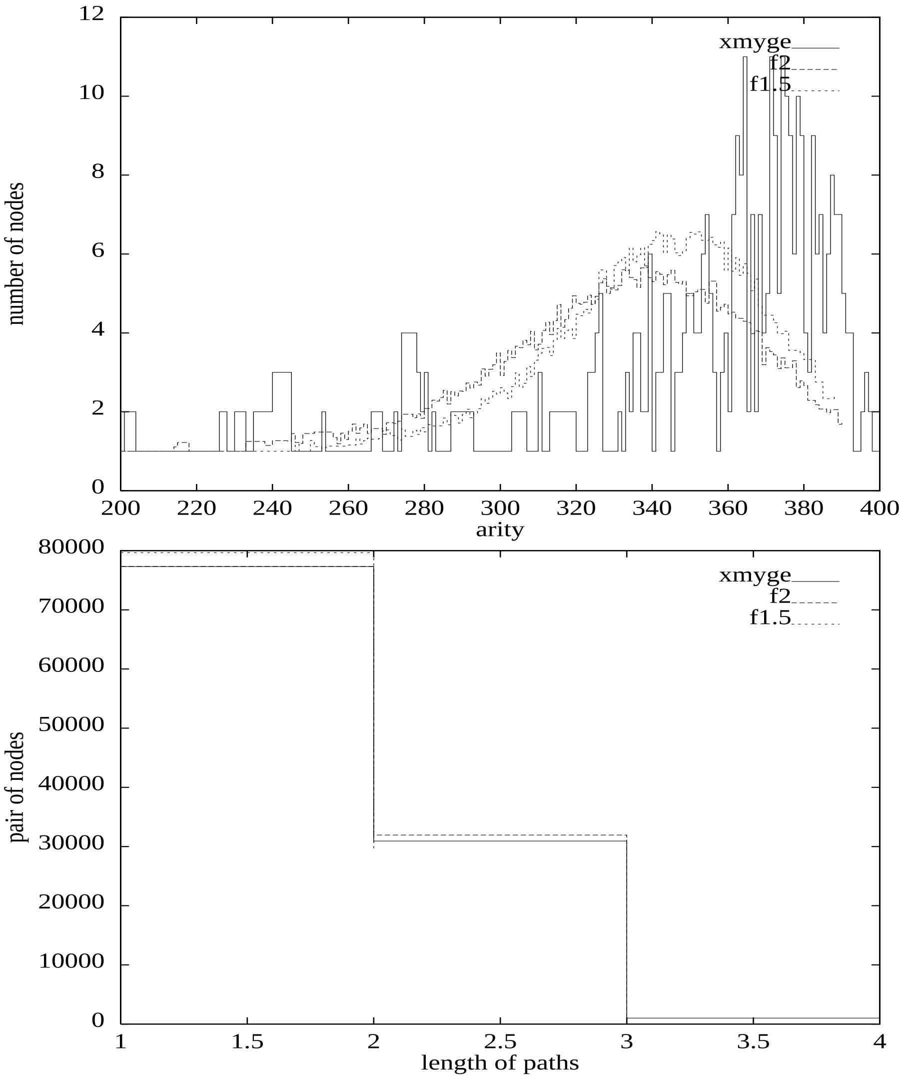
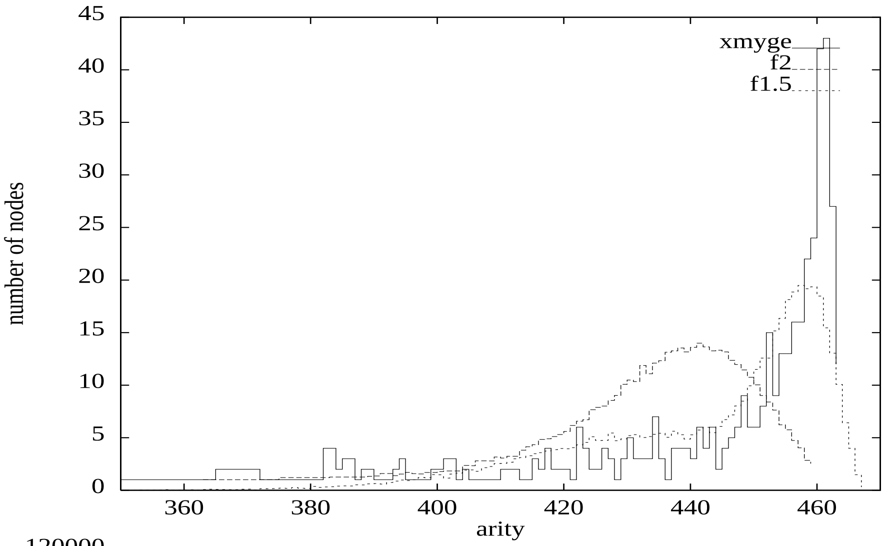


Figure 2.8: Degree and (length of) path distributions (respectively at the top and at the bottom). The curve xmyge refers to the graph built for the shuffled sequences (77323 edges), while f2 and f1.5 are the distributions of, respectively, the corresponding graphs in $\mathcal{G}(n, m, 2)$ and $\mathcal{G}(n, m, 1+0.5)$.



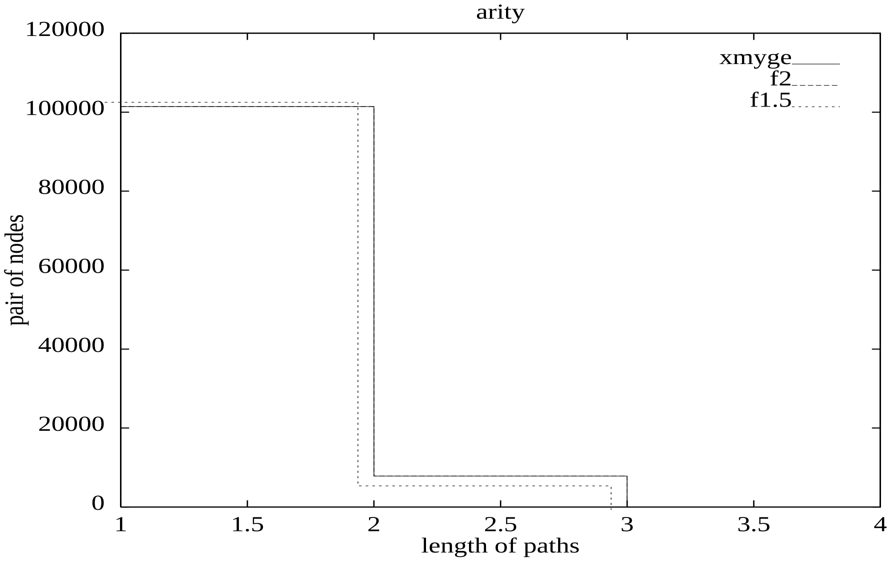


Figure 2.9: Degree and (length of) path distributions (respectively at the top and at the bottom). The curve xmyge refers to the graph built for the shuffled sequences (101431 edges), while f2 and f1.5 are the distributions of, respectively, the corresponding graphs in $\mathcal{G}(n, m, 2)$ and $\mathcal{G}(n, m, 1+0.5)$.

low thresholds and degree statistics) with similar models of random graphs.

Hence, in next chapter we will compare the graphs built from the two genomes of *Mycoplasma genitalium* for all the thresholds with the corresponding Levenshtein graphs built from the shuffled sequences and, at the same time, just to test the random models, we will compute also the differences among the biology and the random data of the different models we conceived so far.

Chapter 3

Applications to a bacterium

3.1 Description of the organism

We give here a brief description of the bacterial genome on which we test our approach. More information than the content of this section can be found in [4].

Mycoplasmae are members of a genus belonging to the class *Mollicutes* (soft skin), due to their complete lack of cell wall components. They have the smallest and simplest genome of all free living-species. At least 24 separate species have been characterised so far. These bacteria exist in nature in relationship with some specific higher eukaryotic plants or animals. *Mycoplasma genitalium* and *Mycoplasma pneumoniae* are the only two ones associated with the human species. Since they are pathogenic to humans, these two are the most studied among mycoplasmae. It is for this reason that the complete genome is already (although only recently) available for both of them.

3.2 Comparing histograms

If we had had a theoretical model (that is one with statistical results about mean and variance that could avoid the random generation of data) to be compared with the graphs representing the genomes, we would have easily detected values of the threshold for which biology sensibly differs from the expected values with respect to the variance, and consequent detailed analyses for such values. For example, supposing that we knew that the degree of a node has a mean value $<a>$ and a variance σ^2, then we could compute, for nodes in $\mathcal{G}_\Gamma$ whose degree is a, the value

$$z = \frac{a - <a>}{\sigma}$$

that gives a measure of how much the observed value differs from the expected one with respect to the variance. When we detect nodes for which z

is larger than (say) 3, we may then claim that the gene represented by that node has unexpected similarities with another gene and then a more detailed biological analysis could investigate in more depth the reasons for this.

Unfortunately, we do not have here such a theoretical model and we will work by comparing degree distributions (that is the histograms) of Levenshtein and randomly generated graphs.

In the previous chapters, for a preliminary test of the models, we just compared histograms by plotting the curves and looking at the shape and values. Of course, an application to the true biological issue a more detailed comparison. In fact, we will compute a numerical difference among each pair of histograms we will compare. Let us suppose that we want to compare the histogram B representing biological data with the random one R, where both have values for all $i = 1..n$. We will measure the difference among them by using the chi-square:

$$\chi^2 = \sum_{i=1}^{n} (B[i] - R[i])^2 / R[i].$$

The histograms of degree distributions are computed for the biological data for each threshold and then compared, for all of them, to the corresponding random graphs. In fact, using the previous formula we will compute the difference with (graphs built from) the shuffled sequences, the graph in $\mathcal{G}(n,m)$, the one in $\mathcal{G}(n,m,2)$, and finally the one in $\mathcal{G}(n,m,1{+}0.5)$. Results will appear in the next sections.

It should be pointed out that when comparing the graph representing the genome ($\mathcal{G}_\Gamma$) with the one built from the shuffled sequences for each threshold of the graph $\mathcal{G}_\Gamma$, we do not use the same threshold Γ on both graphs. We rather search for the corresponding threshold of the shuffled version such that the two graphs have the same number of edges. Most of the time the threshold happens to be the same, but there appear slight differences sometimes. With the same request (the same number of edges of the $\mathcal{G}_\Gamma$ for each one of the Γ) we generate the three different kind of random graphs.

3.3 Degrees in *Mycoplasma genitalium*

Figure 3.1 shows the χ^2 for all the thresholds (only relevant data, that is from $\Gamma{=}0.7$ to $\Gamma{=}1.0$, is shown). As we expected, the lowest values appear for the shuffled genome, as it is the most similar to *Mycoplasma genitalium.* After it, there comes the curve of $\mathcal{G}(468, m, 1{+}0.5)$, better than the $2-$*closed* one, while the basic random graph is confirmed as a bad model. In particular, one can see

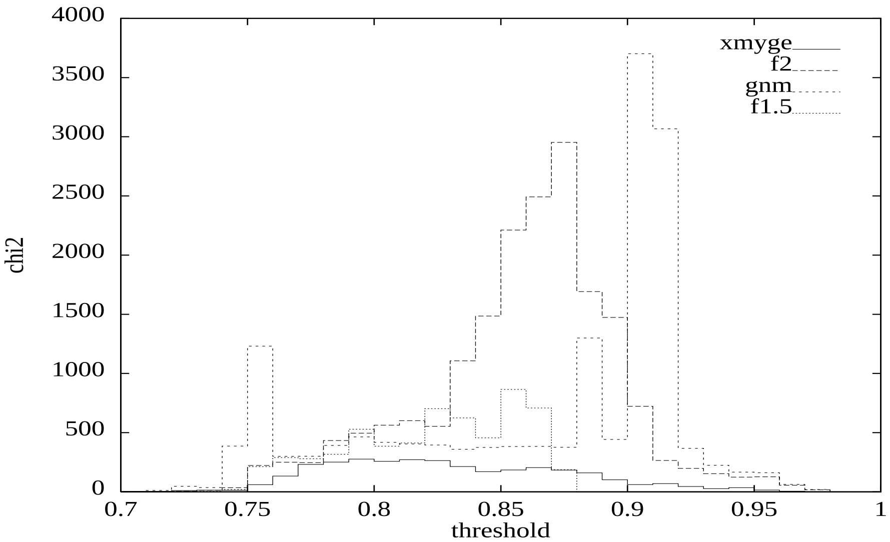


Figure 3.1: χ^2 of degree histograms for all the thresholds (differences from degree distributions of the Levenshtein graph built from *Mycoplasma genitalium*). The curve xmyge refers to the shuffled genome, f1 is the one corresponding to random graphs of the basic model $\mathcal{G}(468, m)$, f2 corresponds to graphs in $\mathcal{G}(468, m, 2)$, and finally f1.5 corresponds to graphs of $\mathcal{G}(468, m, 1+0.5)$.

that for the lowest thresholds (among those shown in fig. 3.1), our theoretical model $\mathcal{G}(468, m, 1{+}0.5)$ does not behave much worse than the shuffled genome.

We went then to detail the data detected by $\mathcal{G}(468, m, 1+0.5)$ for low thresholds. We do not claim here that this is the perfect model for the comparison and that the following results are biologically significant. We just want to investigate the kind of *unusual* properties that such a model detected in the graph $\mathcal{G}_\Gamma$.

3.4 Back to biology

We focused our attention on values of the threshold up to 0.73, that are those for which the model $\mathcal{G}(468, m, 1{+}0.5)$ still remain close to the shuffled data. In particular, for such values of the threshold, we looked at which genes happen to be similar to others and may then form a group.

In *Mycoplasma genitalium*, even with a threshold $\Gamma = 0.0$ we have an edge in $\mathcal{G}_\Gamma$ as there are two perfectly identical genes. Raising the threshold, new edges appear resulting in new groups of similar genes. For values of the threshold up to 0.67, the groups that appear are not such news as the corresponding genes are so similar to each other that *classical* alignment techniques could easily detect them. We list below all groups appearing for these lower thresholds:

0.00	Unknown_261	polI_262		bb
0.30	Unknown_413	Unknown_414		b
0.55	Unknown_96	Unknown_288		b
0.55	gyrB_3	parE_202		bb
0.57	Unknown_185	Unknown_259		bb
0.61	Unknown_68	Unknown_395		b
0.63	SPase_67	Unknown_68		b
0.63	SPase_67	Unknown_68	Unknown_395	bb
0.64	glpQ_293	Unknown_385		b
0.64	hlyB_179	glnQ_180		b
0.65	Unknown_438	Unknown_439		b
0.66	Unknown_25	rfbV_60		bb
0.67	Unknown_224	aroP_225		bb
0.67	Unknown_431	Unknown_442		bb
0.67	gyrA_4	parC_203		bb
0.67	hlyB_179	cbiO_304		b
0.67	hlyB_179	glnQ_180	cbiO_304	b

Each line starts with the threshold and then lists the group of genes that resulted similar. For each Γ, we listed only new groups, avoiding to list, for example, the pair of two identical genes Unknown_261 and polI_262 that appear already for $\Gamma = 0.0$.

The 'bb' and 'b' at the right side of each line have the following meaning: we just stated that these similarities are detectable with the usual techniques. In fact, all of this similarities are detected by the well known program *BLAST*[1]. The sign 'bb' denotes that a running of BLAST detected exactly the same group of genes as being similar to one another. The sign 'b' indicates that, next to the genes listed, BLAST also detected similarities with other sequences (with a $pvalue < 10^{-3}$). The reason for this is that BLAST does not use the Levenshtein distance and thus catches similarities among genes even when they have much different lengths. These are missed by our method in its current state.

The range of Γ where our $\mathcal{G}(468, m, 1+0.5)$ detects something and where the comparison with *Mycoplasma genitalium* seems to be still feasible, is the one that comes right after the above data and up to $\Gamma = 0.72$. Here is the complete list of our results:

0.68	rpL35_196	rpL34_465		n
0.68	rpL36_174	rpL35_196		n
0.68	rpL36_174	rpL35_196	rpL34_465	n
0.69	hlyB_179	glnQ_180	glnQ_303	b
0.70	rpL24_162	Unknown_393	Unknown_448	n
0.70	rpL36_174	rpL35_196	BS18_423	
	rpL34_465			n
0.71	Unknown_267	Unknown_319	P69_406	n
0.71	glnQ_180	Unknown_290	glnQ_303	b
0.72	Unknown_198	Unknown_242	nodF_287	
	Unknown_337	Unknown_389		n
0.72	Unknown_55	ptc1_108	Unknown_221	
	Unknown_239	pip_310	gidB_380	
	Unknown_415	Unknown_418		n
0.72	amiE_79	hlyB_179	glnQ_180	
	cbiO_304			b
0.72	dnaE_10	Unknown_57	nodF_287	
	Unknown_337			n
0.72	rpS19_155	rpL36_174	rpL33_325	
	BS18_423	rpL28_425		n

[1] BLAST is probably the fastest database scanning technique available at the current time ([14])

As we can see, only a few of them are detected by BLAST (the sign 'n' means that BLAST did not detect any significant similarity within the group). Most of them are only suggested by $\mathcal{G}(468, m, 1+0.5)$. The first three groups, appearing for $\Gamma = 0.68$, correspond indeed to families of ribosomal proteins, as well as the last group of 5 genes at $\Gamma = 0.72$. We had a more detailed look at the second group appearing at $\Gamma = 0.72$ that consists of 8 genes. We ran BLAST for the gene gidB_380 (in the graph $\mathcal{G}_\Gamma$, it is this gene whose node has the seven edges linking whole the group), and we observed that none of the similarities detected here are found by BLAST. What it did detect are similarities (even strong ones) with genes of very different size, for the reasons we already explained. Thus, since we had a similarity detected by $\mathcal{G}(468, m, 1+0.5)$ and not by BLAST, we went to investigate one of them at the level of the sequences. Here is the alignment of the sequences gidB_380 and Unknown_418:

```
Unknown_418   MIETLNFEKQHYAFLIKAIEENTNFGLSQLTLIDRLKA
gidB_380                           MNNANF          EKYVD
                                    :*:**           ::

Unknown_418   IVISYNEFFNQKPLTISSPSNEKSLHLETEYLEKKKIK
gidB_380      LVFEANKNFN                  LTGFKTKEAIY
              :*:- *:  **                  *  :   *: *

Unknown_418   KSNHKQDQKHFSLFEKSFIDKSEKTPKNDEVTNNKFLD
gidB_380      QNLVIEILTLFKGYEKFFIDKTVADLGSGNGSPGIILK
              :-    :   - *- :** ****:      --:  : -  :*

Unknown_418   TSKLNLANIALAINAFNDNKWINHFQNLLSVFQTKFNDKD
gidB_380      LLFQKIKKLVLIDSKHKKISFLNKLTKQLNLEKTVAICER
                  ::  ::-*  -  -:-  -: :*::  : *-:  :*      :

Unknown_418   KQNNLSYFNNFIDKYSARDIVKATKIVKASSFGIVILFE
gidB_380          IEVHKNHYDVICSRGLS     TIIKVNDLAFSLLNS
                  :-  -:*-  *   -:*-:      -*:*---:-: :* -

Unknown_418   DQKIAMRLWKEAIEEGNVQATIFQIFNQNLFLASFSEHQ
gidB_380                     KG      IIFHIKQS
                             :*       **:* :-
```

```
Unknown_418      YKTTITEETKNQKYQTEVLNLTQLENLAKPFLKEKKRSLS
gidB_380         LDQYIEFEKSNQKNQFNLLFIKHFTSQNKKLILIALQKND
                  -  *  *--*** * ::* :-:: -  * ::    :- -
```

```
Unknown_418      QKMVDKYFKGLFEEK
gidB_380
```

The sign '*' denotes a perfect matching, while ':' indicates that the two amino acids are very similar. A less strong similitude is denoted by '-'.

This alignment shows that, indeed, there exists a similitude between Unknown_418 and gidB_380. Moreover, we can notice that a particular subsequence cause a perfect matching: FIDK. It appears twice in Unknown_418. This pattern is not very common in protein data banks. The genome of *Mycoplasma genitalium* presents an unusual occurrence of it. We made a query into the Swissprot data bank (release 35), asking for the pattern FIDK. We first searched for it in the entire genome of *Mycoplasma genitalium* only and we found that it appears 6 times. On the contrary, in the whole Swissprot it resulted to appears only 292 times. Since Swissprot has 69113 sequences and only 468 of them belong to *Mycoplasma genitalium*, then we should have observed $6 * 69113/468 = 886$ hits in Swissprot instead of only 292. From this we infer that this particular pattern could be important in *Mycoplasma genitalium* and, probably, that Unknown_418 and gidB_380 are somehow related genes.

Conclusions

Although many approximations were made along the way, eventually our rough model detected some interesting and unusual properties of *Mycoplasma genitalium.*

This work should be considered as a preliminary study of a new method. There are several further work and improvements that came in mind while designing and testing this new approach.

First of all, another notion of distance among biosequences could be used. For example, a more suitable one could be what is often denoted as the *best fit* distance that could better detect events of duplications in bacteria. In a few words, such a distance between two genes, having in general different sizes, detects the best alignment of the smaller sequence into the larger one. In fact, duplications in bacteria consist mostly of two events: firstly, a gene could truncate and a thus subsequence of it becames a new shorter gene; secondly, two genes could concatenate and give rise to a new longer one. It is easy to see that both these events are missed by the Levenshtein distance but could be detected by a *best fit.*

Another possible improvement concerns the different rates of duplications in different families of genes. It is documented ([8]) that some genes duplicate more often that others. In our model, this could be represented by a larger probability of the presence of certain edges: genes of the same and quickly duplicating family, should correspond, in the random graphs, to nodes that have a larger probability of being connected by an edge. This point is related to another one: we have seen how even the best model of random graphs $\mathcal{G}(n, m, \alpha{+}1)$ is not yet good enough to be used for a comparison, and we have also stated that both a lower and a higher degree of closure cannot make things better. Thus, the only possible improvement that can be done is the one that closes with a certain probability as well, but whose probability is suitably chosen according to the specific organism's properties. This would imply a loss of generality, but actually, if a theoretical random model did exist, it would not be that dramatic to set some parameters for each organism to whcih we apply the method. What we mean (what follows is just an idea that has not been tested yet) more in detail, is that we could perhaps fix different probabilities p_i for different set of edges (say, pairs of genes). This values could probably be set observing the distance distribution, which is immediately available at the beginning of the analysis and not yet depending on the threshold).

Another further work is the application of the approach to the recently sequenced genome of *Bacillus subtilis* ([9]) as well as to the other Mycoplasma whose genome has already been sequenced (that is *Mycoplasma pneumoniae*). *Bacillus subtilis* has 4099 genes, and consequentely, since some of our algorithms require $O(n^3)$ time (with $n = 4099$) we opted for testing the method with smaller, equally interesting, genomes. For analogous time complexity reasons, we did not include yet among the parameters of comparison, statistics about cliques in the graphs. It is clear that this data also has an interesting biological meaning: nodes that belong to a clique would correspond to families of edges that are all similar to one another. Moreover, a special interest could be addressed to *Mycoplasma pneumoniae* from the fact that, since it is, as well as *Mycoplasma genitalium*, pathogenic for humans, these two bacteria are the first two completed genomic sequences belonging to the same genus, and thus their analysis provides a strong foundation for comparative genomics.

Finally, as we saw in the tests of the model, we can reasonably hope, for the model $\mathcal{G}(n, m, \alpha{+}1)$, to obtain better results when comparing data concerning paths in the graphs. Hence, such an application is another possible further work.

Acknowledgements

There are many people that I have to acknowledge for this time I spent in France. This year of DEA, in fact, for me coincided with one year spent abroad, with all the difficulties and the great experiences that this means.

Thus, I thank whoever allowed me to live one year in Paris: my family that helped me economically, my italian advisor prof. F.Luccio and the director of Pisa's P.h.D. program prof. A.Maggiolo Schettini that allowed me to terminate this experience before starting my P.h.D. in Italy, as well as the friends I met here that was so kind with me.

Special thanks to all the people of the Atelier de Bioinformatique, where I did my stage and where they made me feel at home. In particular, thanks to prof. A.Viari, my advisor, that was always kind and patient with me, and to E.Rocha and G.Achaz that helped me with duplicating genomes.

This work received an useful *maquillage* thanks to the suggestions of the prof. M.F.Sagot and M.ter Beek, that also encouraged me a lot.

Finally, all this would have never been possible without the help, the hospitality, and the friendship of the family Longo that suggested and made possible this year in Paris. This memoire is definitely dedicated to them.

Bibliography

[1] A.R.Mushegian and E.V.Koonin. A minimal gene set for cellular life derived by comparison of complete bacterial genomes. *Proc. Natl. Acad. Sci. USA*, 93:10268–10273, September 1996.

[2] A.V.Aho, J.E.Hopcroft, and J.D.Uhlman. *Data structures and algorithms.* Addison-Wesley Publishing Company, 1982.

[3] B.Bollobás. *Random Graphs.* Academic Press, 1985.

[4] F.J.de Bruijn, J.R.Lupski, and G.M.Weinstock. *Bacterial genomes. Physical structure and analysis.* Chapman & Hall, 1997.

[5] C.Ouzounis, G.Casari, C.Sander, J.Tamames, and A.Valencia. Computational comparisons of model genomes. *Bioinformatics*, 14:280–285, 1996.

[6] C.Seoighe and K.Wolfe. Extent of genomic rearrangement after genome duplication in yeast. *Proc. Natl. Acad. Sci. USA*, 95:4447–4452, April 1998.

[7] D.Sankoff and J.K.Kruskal. *Time warps, string edits, and macromolecules: the theory and practice of sequence comparison.* Addison Wesley, 1983.

[8] E.Coissac. *Analyse structurale et fonctionelle du génome de la levure* Saccharomyces cerevisiae. PhD thesis, Université Pierre et Marie Curie (Paris VI), 1996.

[9] F.Kunst et al. The complete genome sequence of the Gram-positive bacterium *Bacillus subtilis. Nature*, 390:249–256, 20 November 1997.

[10] E.V.Koonin, R.L.Tatusov, and Kenneth E.Rudd. Sequence similarity analysis of *Escherichia coli* proteins: Functional and evolutionary implications. *Proc. Natl. Acad. Sci. USA*, 92:11921–11925, December 1995.

[11] J.B.Walsh. How Often Do Duplicated Genes Evolve New Functions? *Genetics*, 139:421–428, January 1995.

[12] J.C.D.Hinton. The *Escherichia coli* genome sequence: the end of an era or the start of the FUN? *Molecular Microbiology*, 26(3):417–422, 1997.

[13] J.C.Setubal and J.Meidanis. *Introduction to Computational Molecular Biology.* PWS Publishing Company, 1997.

[14] M.S.Waterman. *Introduction to Computational Biology. Maps, sequences and genomes.* Chapman & Hall, 1995.

[15] Tomoko Ohta. Simulating Evolution by Gene Duplication. *Genetics*, 115:207–213, January 1987.

[16] P.E.Hansche. Gene duplication as a mechanism of genetic adaptation in *Saccharomyces Cerevisiae. Genetics*, 79:661–674, April 1975.

[17] P.Slonimski, M.O.Mosse, P.Golik, A.Henaut, J.L.Aude, A.Wozniak, E.Glemet, and J.J.Codani. The First Laws of Genomics. In Tokyo Universal Academy Press, editor, *Genome Informatics*, pages 9–10, 1997.

[18] R.L.Tatusov, E.V.Koonin, and D.J.Lipman. A Genomic Perspective on Protein Families. *Science*, 278:631–637, 24 October 1997.

[19] W.H.Press, S.A.Teukolsky, W.T.Vetterling, and B.P.Flannery. *Numerical recipes in C.* Cambridge University Press, 1992.