CZECH TECHNICAL UNIVERSITY IN PRAGUE
Faculty of Nuclear Sciences and Physical Engineering

DIPLOMA THESIS

Slow evolutionary dynamics of RNA structures

2009 Petr Šulc

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta jaderná a fyzikálně inženýrská
Katedra fyziky

DIPLOMOVÁ PRÁCE

Pomalá evoluční dynamika struktur RNA

Slow evolutionary dynamics of RNA structures

Posluchač:	Petr Šulc
Školitel:	Dr. Olivier C. Martin
Akademický rok:	2008/2009

Název práce: **Pomalá evoluční dynamika struktur RNA**

Autor: Petr Šulc

Obor: Matematické inženýrství: Matematická fyzika

Druh práce: Diplomová práce

Vedoucí práce: Dr. Olivier C. Martin, LPTMS, Université Paris-Sud, Orsay, Francie

Abstrakt: Zabýváme se evoluční dynamikou sekundární struktury molekul RNA. Nejprve uvedeme motivaci tohoto problému z hlediska biologie a poté provedeme numerické simulace vývoje a vyhodnotíme výsledky. Zajímáná nás zejména, zdali evoluční proces RNA vyznačuje stejnou relaxační dynamiku jako např. spinová skla. Rovněž diskutujeme matematický model procesu a studujeme další vlastnosti neutralních sítí sekundární struktury RNA. Dále uvádíme model pro výpočet času stráveného na neutrální síti. Používáme model náhodné procházky po náhodném grafu s absorbujícím uzlem. Výsledky našeho modelu testujeme na numerické simulaci.

Klíčová slova: evoluce RNA, pomalá dynamika, neutrální sítě, náhodná procházka, grafy

Title: **Slow evolutionary dynamics of RNA structures**

Author: Petr Šulc

Abstract: The dynamics of RNA secondary structure evolution is examined. After introducing biological motivations for our problem, we perform numerical simulations of the secondary structure evolution and interpret the results. We are especially interested whether the relaxation dynamics expresses the same relaxation behavior as other slow dynamics relaxation systems such as spin glasses. Mathematical model of the process is also examined. We further examine several properties of RNA secondary structure neutral networks. We then present a model to estimate the time spent on a neutral network which is studied using a random walk on a random graph with an absorbing node and we test our results by numerical simulations.

Key words: RNA evolution, slow dynamics, neutral networks, random walk, graphs

Acknowledgements

I would like to thank my supervisor Olivier Martin for his help and support during the internship as well as his consultations throughout the writing of this diploma thesis. His attitude was very motivating and enriching and has made the internship a truly enjoyable experience. I am also grateful to his colleague prof. Andreas Wagner from the University of Zurich, who helped us to define the concrete objectives of our work and provided valuable pieces of advice concerning the biological aspect of the problem. He also enabled us to use the computing resources of his research group in Switzerland, where we performed part of our computer simulations. I also thank the other researchers and employees of LPTMS who helped me with mathematical or computer problems that arose during the course of our work.

I would also like to express gratitude to prof. Jiří Tolar for his guidance throughout my studies at the Faculty of Nuclear Sciences and Physical Engineering.

Contents

Introduction

This diploma thesis is based on results obtained during a research internship which was held at the *Laboratoire de Physique Théorique et Modélisation Statistique* at the *Université Paris-Sud, Orsay* during the period from early April to the end of July 2008. The internship was carried out under the supervision of Olivier Martin. We studied the dynamics of RNA structure evolution by the means of Monte Carlo simulation and then interpreted the results. In the following chapter, we first introduce and describe the problematics of RNA structure evolution and explain its interdisciplinary relation to physics and biology. We then describe the actual simulation algorithm and discuss the results produced. The mathematical aspect of the problem is also examined. Finally, the simulation code is briefly described in the appendix. The full source code together with simulation results are provided on the enclosed CD-ROM. A brief glossary of biological terminology is included for easier orientation throughout the text.

Chapter 1

Motivation and goals

In many systems, dynamics undergo an anomalous slow-down, a feature characteristic of “complex” landscapes. This arises for instance in physical systems (thermal relaxation), in optimization problems (diminishing returns on search efforts), and in evolutionary dynamics (long periods of stasis in the evolutionary records). In this work, inspired by theoretical frameworks from statistical physics for glassy systems [1], we reconsider a toy model of evolutionary stasis, namely RNA secondary structure evolution.

The relation between genotype and phenotype in biology is generally very complex. The genotype of an organism is its genetic information while the phenotype is associated with measurable properties of the organism (such as gene expression level or proteome for single cells, size, longevity etc. for a pluricellular organism). This “mapping” from genotype to phenotype can be extended to the molecular level where it remains a difficult problem; for instance one doesn't know how to reliably predict the structure of a protein from its sequence of amino acids. One of the simplest yet realistic maps however is the relation between RNA primary and secondary structures. This simplicity is relative because the relation remains nevertheless nontrivial. The relevant information about RNA secondary structure and its relation to the genotype will be given in chapter 2.

Evolutionary pressures act on the phenotypes and so a natural question is whether the mapping from genotype to phenotype allows for efficient “optimization” of genotypes when putting selection pressures on phenotypes. Indeed, if the map is “encrypted”, it is impossible to produce a continuous path in genotype space which will lead to successive improvements in the phenotype. The analogy with physical systems is quite direct and is best given in the context of a disordered system such as a glass: by incrementally changing a configuration (positions of atoms), can one reach low energies or even the ground state of the system? The microstate (configuration) plays the role of the genotype, while the energy is related to the phenotype. Thus just as one studies energy landscapes in disordered system, it is natural in evolution to study fitness landscapes.

We shall focus in this work on how evolutionary dynamics can lead to improved phenotypes in the context of RNA. To specify the fitness (energy) of a genotype, we compute the corresponding phenotype and measure the "distance" of this secondary structure to that of some target structure. This last structure is considered to be "optimal" i.e. preferred by the selection process because of an underlying enzymatic function for instance. How does the fitness evolve with time? Specifically, we would like to examine the following questions:

- During the evolution towards a target secondary structure, do the RNA dynamics slow down or does one have "standard" exponential approach to the target?

- Does the speed of the dynamics vary a lot with the initial condition or with the target structure?

- During the evolutionary process, does the trajectory encounter atypical genotypes or structures? Indeed, the non-equilibrium process may favor transitions to structures that are anomalously abundant or robust or evolvable.

- Finally, how do these different properties depend on the size of the RNA molecule?

We used a computer Monte Carlo simulation to model the process of approaching the optimal structure as a stochastic walk through the phase space of all possible genotypes.

Inspired by the diffusion process in neutral network, we then examined random absorbing walks on random graphs and estimated the mean time spent on such graphs.

Chapter 2

RNA folding and neutral networks

We briefly present basic facts about RNA molecules and use these to illustrate the relation between genotypes and phenotypes in evolutionary dynamics. This will bring us to the notion of neutral networks and neutral sets which play an important role in the process of searching for the optimal phenotype. They will be described in more detail in the later paragraphs.

2.1 RNA and its secondary structure

2.1.1 Primary structure

RNA molecules are organic molecules found in all living organisms. They consist of nucleotides that are joined together to form a chain. There are four different bases associated with these nucleotides in RNA molecules: adenine, cytosine, guanine and uracil, which are usually denoted by the letters A, C, G, U respectively (whose chemical structure is shown in figure 2.1). Together they polymerize to form a single chain. The sequence of bases that composes the RNA molecule is called its primary structure. It is believed that RNA arose before DNA and that it was originally RNA that carried the genetic information in cells. In our simulation, the RNA primary structure will play the role of genotype.

The composition of a chain is always described in terms of bases, for example $ACGGGUA$ is a chain of length 7. The distance between two primary structures (genotypes) is defined as the number of positions where their bases are different; it is thus a Hamming distance.

2.1.2 Secondary structure

The secondary structure of RNA refers to the shape of the folded molecule; different bases can be paired through chemical bonds and therefore it is more advantageous to make such bonds to minimize the total energy of the molecule. The hydrogen bonds that appear in RNA are A-U and G-C, to lesser extent also G-U. Because of the

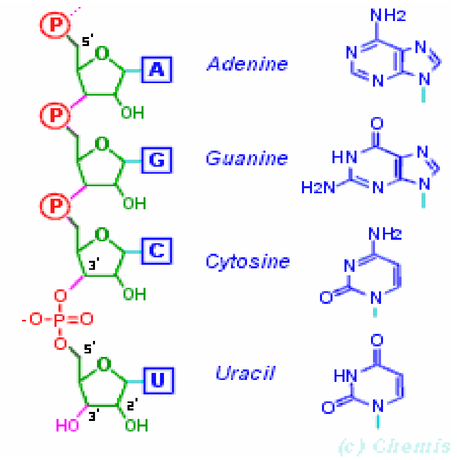

Figure 2.1: An example of a short part of RNA molecule. The chemical structure of the bases is shown on the right side. Source: `chemis.org`

interaction between bases, the RNA chain is neither linear nor coiled as a polymer but is "folded" as a result of the pairing between different pairs of bases. Figures 2.2 and 2.3 illustrate secondary structures of RNA molecules. Note that the secondary structure arises from the pairing of many bases. The secondary structure can be effectively represented by a pairing in the plane as long as no "knots" are allowed. Then one can specify such a folding by introducing a chain in which unpaired bases are represented by dots while paired bases are represented by oriented parentheses. The task of assigning a secondary structure to a given RNA chain is still an open problem in bioinformatics [2]. Different algorithms have been developed and there are several dynamical programming tools that compute a secondary structure of an RNA chain. For our work, we used the ViennaRNA package [3], which uses Zuker and Stiegler's algorithm [4] to find an energetically optimal structure. Although the current bioinformatic tools do not achieve 100% accuracy in predicting the secondary structure of RNA molecules, the performance of ViennaRNA tool is sufficient for our purposes.

A secondary structure of RNA molecules is often represented using a bracket-dot notation. For example, a short chain AAAAAACCCCCGGGGGUUUUUU has a secondary structure "(((((((((....)))))))))" where bracket "(" means that a corresponding base is paired with a base corresponding to bracket ")". The dot "." means that the corresponding base is not paired. ViennaRNA package also provides a visualization tool for a given bracket-dot chain. The secondary structure corresponding to our example is shown in figure 2.2. A more complicated example of bracket-dot structure "(.((((....)))))......((((((((...........(((((((((.....))))))))).(((.((......)))))........)))).)))).." is shown in figure 2.3. We say that an RNA chain *folds* into its secondary structure.

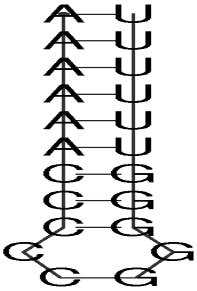

Figure 2.2: Secondary structure of an RNA chain AAAAAACCCCCGGGGGUUUUUU, corresponding to bracket-dot notation (((((((((....)))))))))

It is important to note that two different genotypes (i.e., RNA chains that differ in at least one base) can still have the same secondary structure. The secondary structure denotes the spatial organization of the molecule and therefore determines many of its properties. It is this spatial conformation of the chain that is usually relevant for enzymatic properties, and thus is usually taken to be the molecule's phenotype.

When we mention distance between two RNA structures (or distance between two phenotypes, since the RNA structure is the only phenotype that we will further consider in the following chapters) we always mean the Hamming distance between the two "bracket-dot" chains that specify the secondary structures. The possibility for different genotypes to have the same phenotype leads quite directly to the notion of a neutral network which we will discuss in next section.

2.1.3 Tertiary structure

The tertiary structure of a molecule refers to its three-dimensional structure, i.e., the spatial coordinates of all of its atoms. The tertiary structure is usually determined by the means of X-Ray crystallography or nuclear magnetic resonance spectroscopy. Determining tertiary structures is still an outstanding problem today in spite of years of efforts on the part of the research community.

For RNA, the tertiary structure involves non-planar pairings and thus complicates very seriously the mapping from genotype to phenotype. The situation is nevertheless simpler than in the protein world where the secondary structure cannot be unambiguously determined without also obtaining the tertiary structure; furthermore, the role of a protein is directly associated with its full 3D structure so secondary structure provides no clue of biological function. Fortunately, in the case of RNA, numerous enzymatic activities (e.g., in interfering RNA) do follow from the secondary structure. In line with standard practice of RNA studies, we will give up a bit of realism and will ignore effects due to tertiary RNA structures in our study.

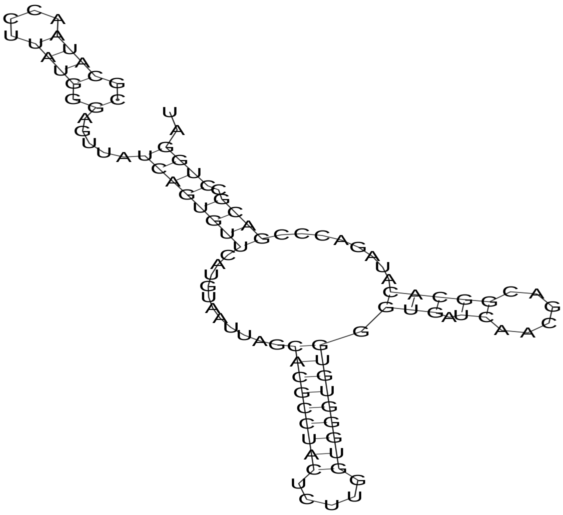

Figure 2.3: Secondary structure of an RNA chain of length 100

2.2 Neutral networks

To describe a neutral network, some elementary definitions from graph theory will be used [5]. Here we just briefly present terminology used in the following chapters and we give more details in section 5.1. By the term neutral network, we mean a graph whose nodes (vertices) correspond to the different genotypes having a given phenotype. Nodes are joined by an edge if they differ in one base, that is their Hamming distance is equal to one and they have the same phenotype. In other words, two nodes are adjacent if we can move from one to the other by a single mutation. It is important to note that mutations which don't leave the neutral network do not change phenotype. It is therefore possible for evolution to perform a diffusion through the neutral network without changing the phenotype. Thus it can significantly modify the genetic sequence while keeping the same phenotype. The number of neighbors is a called a *degree* of the node.

The *fitness* is an important notion in the theory of evolutionary dynamics. It

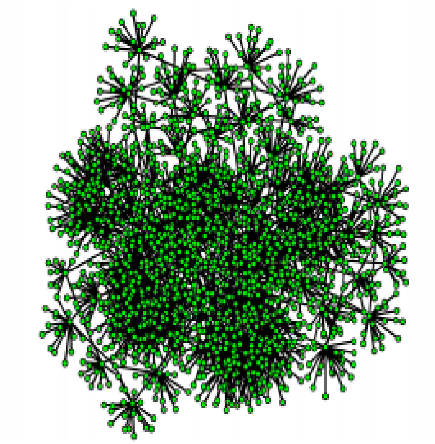

Figure 2.4: An example of neutral network. Each node corresponds to a specific genotype. All genotypes within the network have the same phenotype.

corresponds to a number (typically a positive one between 0 and 1) that is assigned to a genotype which gives the probability that an offspring it generates will survive; in general survival depends on the strength of natural selection which is applied on the phenotype. All genotypes in a neutral network have the same fitness. In computer simulations of evolutionary dynamics, the next generation is produced by generating offspring, some of which survive the selection process.

Network "neutrality" or mean degree, denoted $< d >$, refers to a mean numer of neighnors of a node in the neutral network. This notion can be extended to the context of an evolving population, in which case the population neutrality is the mean degree of the individuals in that population. It can be shown [6] that in the limit of large populations, the population neutrality is equal to the spectral radius of the adjacency matrix[1] of the graph, given that the neutral network is irreducible (meaning that we can reach any node from any arbitrarily chosen node).

Neutral network are of great interest because they allow one to address the relation between evolvability and robustness [7, 8]. RNA structures and chains are often used to test the theories and hypotheses related to evolution; they will also play a central role in our work. To be precise, we shall have to deal with neutral sets really rather than neutral networks, but the framework will be nearly the same. By neutral set we mean a set of neutral networks that correspond to phenotypes with the same Hamming distance to the target structure. Since the distance to a target structure plays the role of fitness in our model, it is therefore possible to change a phenotype while keeping the same fitness.

The mutations that keep the fitness unchanged correspond to a random walk

[1] Element A_{ij} of adjacency matrix is equal to 1 if i-th and j-th nodes are adjacent, otherwise it is 0. We will discuss adjacency matrix in section 5.1

through the neutral network (or set in our case). We distinguish between two possible kinds of random walks: myopic ant and blind ant (meaning that we imagine an ant that performs the random walk). The blind ant random walk means that in each step, we choose randomly with equal probability one of the possible mutations of the genotype. If the mutation leads to a genotype that belongs to the same neutral set (i.e., the randomly selected mutation turns out to be neutral), we move onto this new genotype, otherwise we stay at the node and do nothing (note that the neglected mutations could have been either deleterious or advantageous, leading thus to neutral networks / sets with different fitnesses, as illustrated in figure 2.5). The second type of random walk, the myopic ant random walk, considers only mutations that are neutral and chooses randomly one of them in each step.

Determining the size of a neutral network for a given phenotype is generally an arduous task that we don't have the time nor computational resources to address in our work. However, we will be able to obtain some statistical information on the neutral sets by the means of random walks.

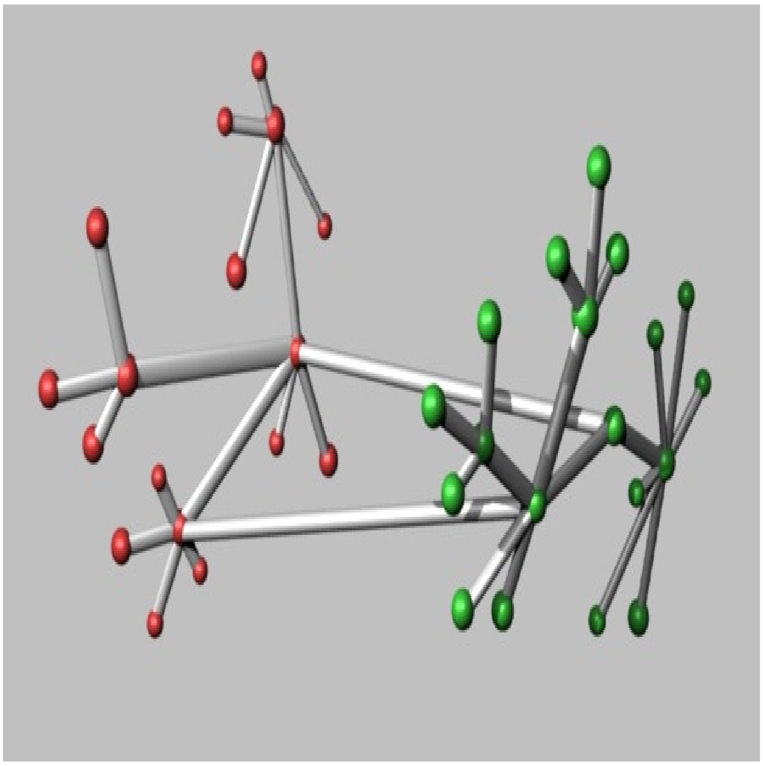

Figure 2.5: The connections between two neutrals sets. Some of the nodes are connected to the nodes from other neutral sets (with smaller distance to the target, thus greater fitness), while others have connections only to the nodes within the same set. As it turns out from our simulation, a majority of the possible mutations are deleterious, thus leading to phenotypes with smaller fitness. When such mutations arise in our simulation, they are not accepted and the genotype is not changed. The above image is informative only, the actual neutral sets in our simulations are much larger.

Chapter 3

The evolutionary model

3.1 Simulation algorithm

We simulate the process of RNA evolution towards a target structure by a random walk algorithm allowing a point mutation (i.e., a mutation of one base) at each step. After the mutation is applied to the genotype, we determine the distance between phenotype of the new genotype and the target phenotype. If a mutation on the current genotype increases the distance to the target, it is refused, otherwise it is accepted, corresponding to the blind ant dynamics. We quantify the fraction of the deleterious, neutral and advantageous steps as a function of the distance to the target. The trajectory is stochastic and is influenced by the initial genotype (an arbitrary RNA sequence) and the target (an arbitrary secondary structure). Only the standard bases A, C, G and U are used.

In a more general context, if μ is the mutation rate and N the population size, the effective number of genotypes in a population scales as $N\mu$. When $N\mu \ll 1$, the population is essentially monoclonal and that is the case we focus on here for simplicity. Another possible approach would be to simulate a population of some number of structures; at each round we would increase in relative proportion those structures that are more advantageous (in our case closer to the ideal structure). However, a simulation of a large population is more expensive computationally and the analysis is simpler when using our random walk framework. (Note that in the literature, the random walk approach is referred to as an “adaptive walk”.)

Our algorithm can be summarized as follows:

1. Generate a random RNA chain.

2. In each mutation step, mutate randomly the chain at a random position. If the (phenotypic) Hamming distance between the new secondary structure and the target structure is higher than previously, the mutation is not accepted.

If the distance to the target has not increased, then we continue with the new chain.

3. Repeat the step 2 until the target structure is reached or until a maximum number of mutations is reached.

The target structure is specified at the beginning of the program. In some of the simulations, we generated randomly the target structure and the simulation data were averaged over multiple runs of the algorithm. We developed different variants of the algorithm that provided different statistical information about the dynamics of the approach to the optimal structure and also about the neutral sets.

To test whether the (non-equilibrium) evolutionary dynamics leads to atypical structures produced during the trajectory, we must obtain a statistical description of structures arising at each given distance to the target. This can be done by sampling uniformly the fitness landscape at each such distance. We do that by Monte Carlo with importance sampling using the Metropolis algorithm; it corresponds simply to using the blind ant dynamics, accepting only mutations that do not change the distance to the target. From this sampling, we can obtain the (equilibrium) mean mutational robustness in this space. One can also consider statistical properties of structures encountered, namely the number of stems or other indices. At the same time, we measure the fraction of deleterious, neutral and advantageous mutations for random genotypes in this space.

3.2 Mathematical model

The dynamics corresponds to a discrete Markov process [11], where the dimension of the vector space is equal to size of the genotype space, 4^L, where L is the length of the RNA molecule. A unique vector corresponds to each of the 4^L possible genotypes, with entries all 0 except for one which has the value 1. More generally, the entries of the state vector give the probabilities of having the system in a given state (given state corresponds to a particular sequence) at the time of interest. The initial state is a vector $\mathbf{v}^0$ with one entry equal to 1 (which corresponds to the initial sequence or genotype) and all others equal to 0. The evolution matrix M is a matrix with non-negative entries with the sum of numbers in a column equal to 1. The probability of occupying any of the genotypes evolves from step to step and we have for the n-th step:

$$\mathbf{v}^n = M^n \mathbf{v}^0. \tag{3.1}$$

The average distance distribution for a given iteration (time) n is given by the inner product of the vectors $\mathbf{d}$ and $\mathbf{v}^n$. We define $\mathbf{d}$ as follows: The i-th entry of vector $\mathbf{d}$ is the Hamming distance of the RNA structure that corresponds to the i-th sequence in genotype (sequence) space. Therefore, the mean distance at iteration n is

$$\langle d\rangle(n) = \sum_{i=1}^{4^L} d_i \alpha_i \lambda_i^n \tag{3.2}$$

where λ_i are the eigenvalues and α_i are the coordinates of $\mathbf{v}$ in the basis composed of the eigenvectors of M, d_i being the coefficients that come from the inner product with $\mathbf{v}$.

An important theorem describes properties of stochastic matrices [12]:

Theorem 1 (Perron-Frobenius theorem for nonnegative matrices)**.** *Let A be real $n \times n$ matrix whose elements $A_{ij} \geq 0$. Then*

1. *There exists a real eigenvalue r such that $|\lambda| \leq r$ for all other eigenvalues λ.*
2. *There is non-negative eigenvector associated with the eigenvalue r*
3. $\min_i \sum_{j=1}^n a_{ij} \leq r \leq \max_i \sum_{j=1}^n a_{ij}$

Therefore, according to the Perron-Frobenius theorem, the eigenvalues of a stochastic matrix will satisfy

$$|\lambda_i| \leq 1 \quad \forall i.$$

The Perron-Frobenius theorem for non-negative irreducible matrices[1] is stronger in a sense that it guarantees that all elements of the eigenvector $\mathbf{g}$ corresponding to r will be positive and that r will be a simple root of the characteristic equation of matrix A.

It is clear that the size of the matrix makes it impossible to estimate its eigenvalues and eigenvectors by standard deterministic methods. But the Perron-Frobenius theorem guarantees that only one eigenvector corresponding to eigenvalue 1 exists if the matrix is irreducible (meaning it cannot be put into block diagonal form by any permutation of rows and columns).

In the case of our evolutionary dynamics, the eigenvector corresponding to eigenvalue 1 is the uniform occupation of the genotypes at (phenotypic) distance 0 to target, that is precisely the neutral network of the target. For our simulations at a fixed distance to the target, the eigenvector is analogously the uniform occupation of the neutral set, i.e., the whole space sampled at given distance to the target.

The components along eigenvectors whose eigenvalue is smaller than one in absolute value approach zero for large times. However, since the expression contains 4^L

[1] Irreducible matrices cannot be transformed into block diagonal form by any permutation of rows and columns. A non-negative matrix A is irreducible if for any two indexes i, j, there exists m such that $(A^m)_{ij} > 0$.

eigenvalues and the Perron-Frobenius theorem says nothing about their distribution (they can be actually very close to 1), we will seek a method to approximate the dependence of mean distance on time by a simpler function.

Chapter 4

Results for the evolutionary dynamics

4.1 The approach to the optimal structure

We now present numerical simulation results. The relaxation curves that are shown below were obtained by the algorithm described in section 3.1. Most of the results presented in this section were obtained by averaging the simulation curves over multiple runs (the number of the actual runs for the shown graphs varies between 1000 and 6000 for different graphs shown). The target structure was randomly generated and 5 runs were performed with given initial and target structure, then a new pair was randomly generated for the next 5 runs, etc.

The corresponding source code that performed simulations is described in appendix A.1 as well as the code that was used to evaluate the data from the simulations. The shown plots were produced with program Gnuplot [13] which was also used perform the fits.

4.1.1 Trajectories and slow dynamics

The y-axis of the following graphs shows the relative distance from the optimal structure (Hamming distance divided by the length of the chain) while x-axis shows the number of mutations. The graph then shows the average distance (taken over several runs) after a given number of attempted mutations has taken place. For comparison, figure 4.1 shows a typical trajectory produced by a single run (no averaging), while the average over 6000 runs is shown in 4.2. We see that in the slow dynamics part of the evolution (not for the first part of the curve where transitions are frequent), the distance shows long plateaus separated by small changes (mainly of 1, 2 and rarely more) in the Hamming distance to the target.

As figure 4.3 illustrates, the approach to the optimal structure is slower for longer chains. The curves obtained from the simulation are smooth and demonstrate that

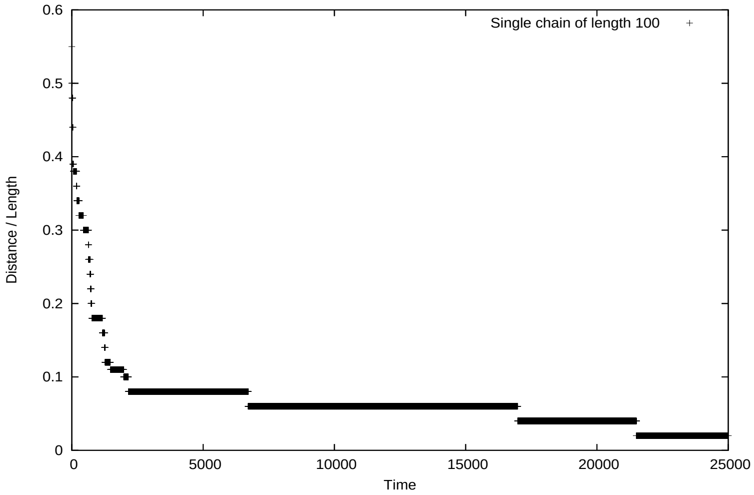

Figure 4.1: Plot of distance to target structure as a function of the number of mutation steps for a single simulation with an RNA chain of length 100.

the dynamics is slower than exponential, as can be seen from nearly all the plots. In the next section, we will determine functions that fit well the obtained data.

4.2 Effective power-law decay for the average trajectories

4.2.1 Plots

Figures 4.4 and 4.5 show semi-log and log-log plots respectively. Figure 4.5 also shows the power law fit function.

4.2.2 Power law fits

As was mentioned previously, we try to approximate the equation (3.2) by a continuous function. As semi-log plots suggest, the relaxation dynamics does not lead to an exponential decay of the distance to the target. We were therefore motivated by the trap model from the glass phenomology approaches that deal with relaxation of a system to its equilibrium distribution [1, 14], where the glass system approaches the equilibrium state as a power function of time. Although there are significant differences between our evolutionary model and trap models, the power law fit worked well with most of the simulations we performed.

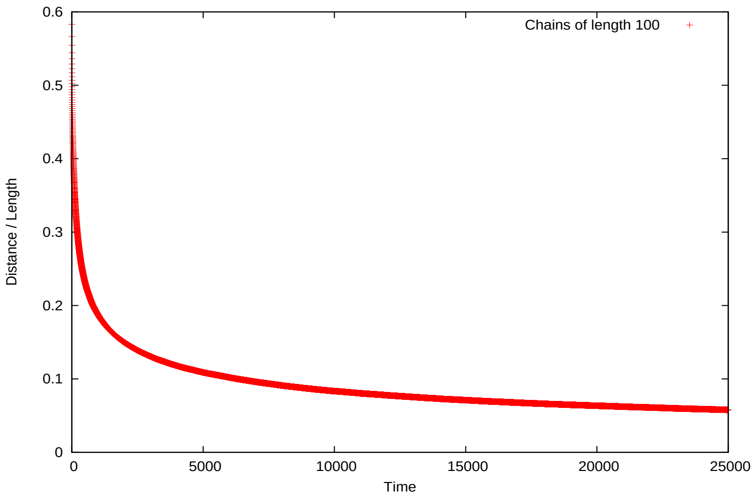

Figure 4.2: Plot of dependence of distance to target structure as a function of the number of mutation steps averaged over 6000 different simulations of relaxation of chains of length 100.

The power law fit function used for the graphs was

$$d(t) = \frac{A}{(t + t_0)^\beta} \tag{4.1}$$

and figures 4.5 and 4.6 provide examples of using such fits to match the evolutionary dynamics.

The values of parameters for function (4.1) that were obtained by fitting the relaxation curves for different lengths of chain are shown in the following table:

Chain length	A	t_0	β	Maximal t
40	36.0	299	0.89	5000
60	50.0	835	0.81	10000
80	14	964	0.6	25000
100	3.7	484	0.41	25000
180	0.95	-685	0.20	70000
200	1.9	-140	0.21	70000
300	0.7	-2641	0.15	140000

One can see several trends, the most important one being a general decrease of the exponent β as the length L of the RNA molecule grows. The quality of the fits is good, but the behavior of the fitting parameters is not easily interpreted.

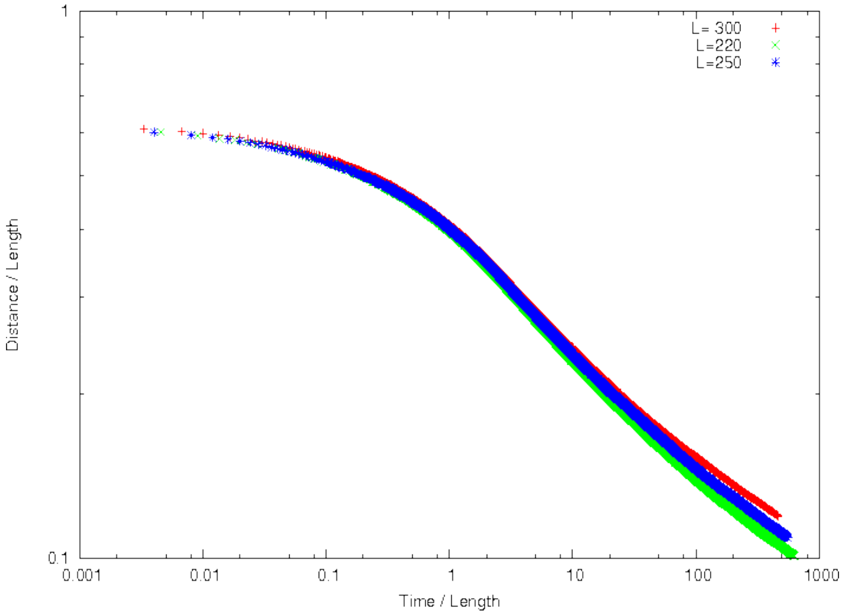

Figure 4.3: Plot with both axes on logarithmic scale of three different simulations of chains of lengths 250, 280 and 300 respectively. The graphs show data averaged over 1000 simulations of relaxation from randomly generated chain to a randomly generated phenotype. The longer the chain, the slower the relaxation process.

4.3 Relaxation towards different targets

To further analyze the slow process of relaxation towards a target structure, we investigated the dependence of the dynamics on the target structure.

We therefore ran simulations of a chain of length 100 with two different targets. In these simulations, we kept the targets different and averaged the relaxation curves over the randomly generated initial chains. The target structure of one of the simulations is shown in figure 2.3, which turned out to be much easier to achieve: relaxation dynamics was much faster than in the case of the structure which is shown in figure 4.7.

Figure 4.8 illustrates how the relaxation speed is different for the two target structures. We conclude that there is slow dynamics whose speed depends on the target structure. This conclusion holds for all the L values we have investigated (see for illustration figure 4.9). An indicator of what makes reaching a target difficult is perhaps the size of the neutral network of the target phenotype. However, this hypothesis is very difficult to test and is beyond our computational capacity.

We ran multiple simulations with fixed targets for different lengths of RNA chains in order to determine whether the process is self-averaging, i.e. whether the relaxation curves have smaller dispersion for different targets when the chain length increases. We ran simulations for tens of different targets and the relaxation curve

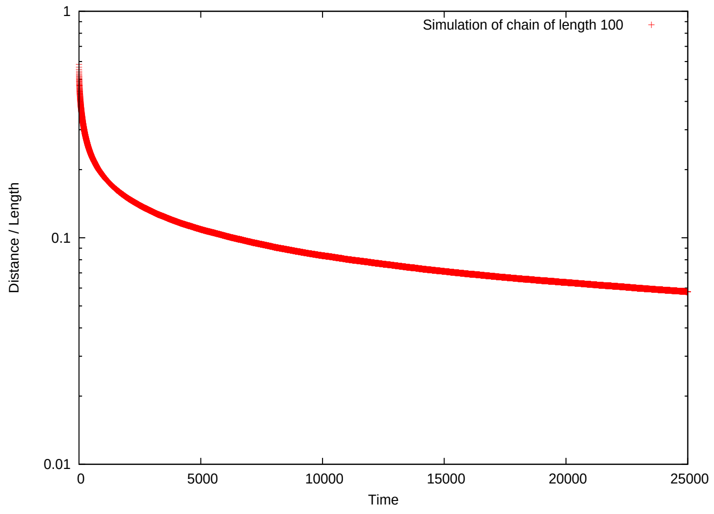

Figure 4.4: Semi-log plot of dependence of mean distance to target structure as a function of the number of mutation steps for chains of length 100, when averaged over all initial genotypes and target structures.

for each respective target was averaged over 1000 random walks with arbitrary initial chain and fixed target phenotype. We then measured the standard deviation of the relative distance to the target phenotype at times where the mean distance of all curves was 10% of the chain length from the target. Data are summarized in the following table:

Chain length	$T_{0.1}$	σ
40	499	0.0346
60	1439	0.02306
80	3349	0.02475
100	6933	0.02683
120	8953	0.02626

where $T_{0.1}$ denotes the time where average of all selected target phenotypes reaches distance $0.1L$, where L is the chain length. The simulations were quite time-consuming which prevented us from testing larger lengths than 120. However, the dispersion decreases initially, but then remains practically unchanged. The relaxation process towards different targets therefore shows a non self-averaging behavior.

4.4 The neutral sets

During the simulation, we gathered data that can be helpful in understanding the role of neutral sets in the approach to the optimal structure. In particular, we

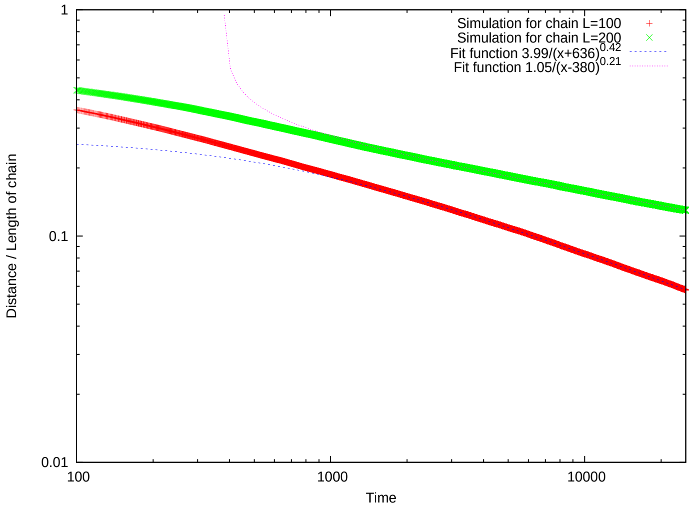

Figure 4.5: Log-log plot of mean distance to target structure as a function of the number of mutation steps for chain of length 100 and 200, when averaged over all initial genotypes and target structures. Also shown are the fits to a shifted power law (see text).

measured additional information concerning the properties of neutral sets, namely, for each neutral set encountered during the simulation (a neutral set is characterized by its distance from the target structure), we measured the fraction of mutations that are deleterious, neutral and advantageous; from this one has a measure of the rate at which one can jump closer to the target. We also determined the distribution of times between beneficial mutations, a quantity that will be useful for our modeling of the dynamics.

Further statistics include waiting times on a given neutral set (figure 4.10), number of different phenotypes visited during the random walk in the neutral set (figure 4.11) and mean distance between the consecutive neutral sets in the random walk. (figure 4.12). All plotted data refer to simulations where we compared relaxation towards two different targets of length 100 (shown in figure 4.8).

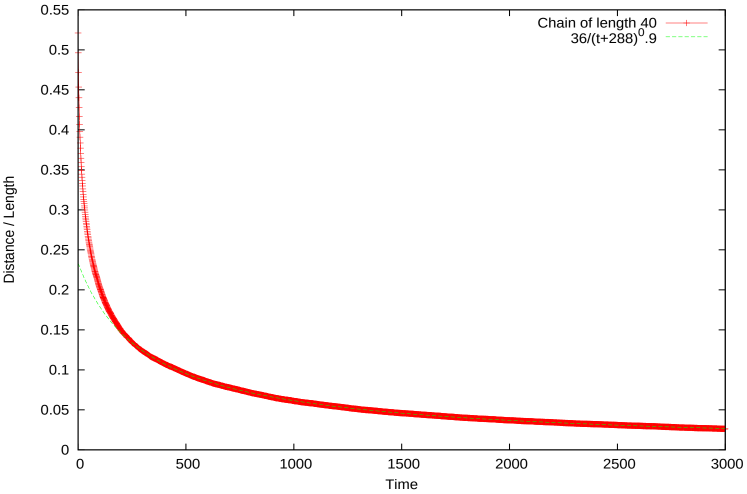

Figure 4.6: Plot of mean distance to target structure as a function of the number of mutation steps for chain of length 40 together with the shifted power law fit function.

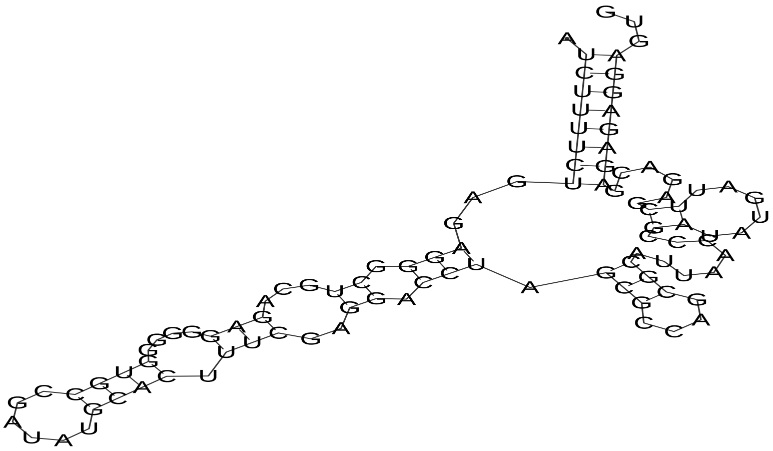

Figure 4.7: Secondary structure of an RNA chain of length 100.

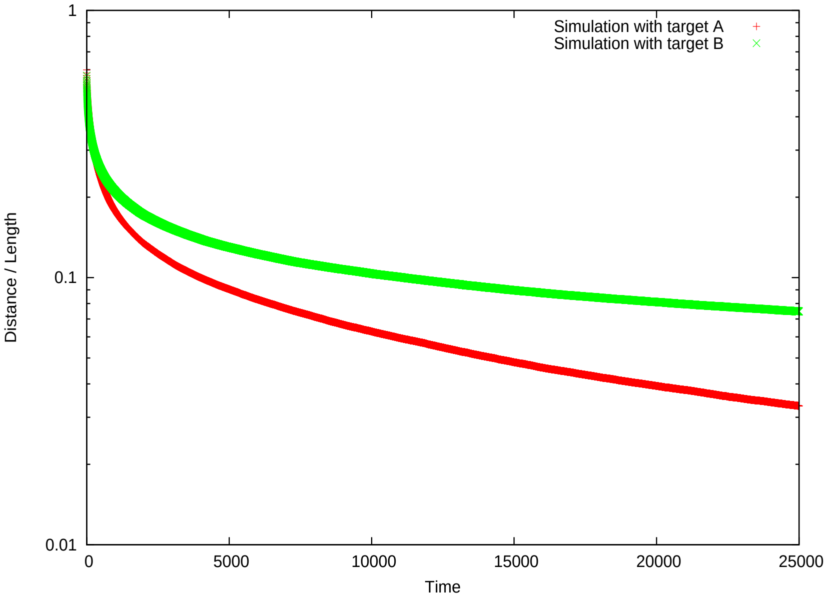

Figure 4.8: Semi-log plot for simulations of chains of length 100, for two different target structure. The curves are averaged over random walks from randomly generated initial sequences.

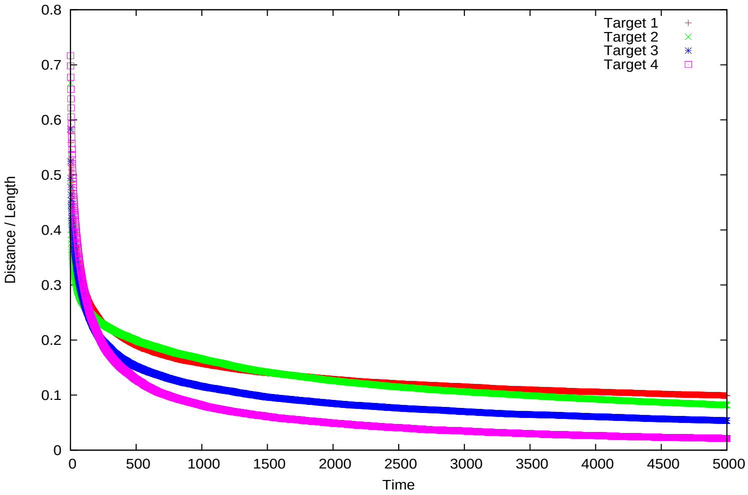

Figure 4.9: Plot for simulations with fixed targets (all of them are of length 60). For each simulation, the plotted curve is an average over 1000 different initial chains.

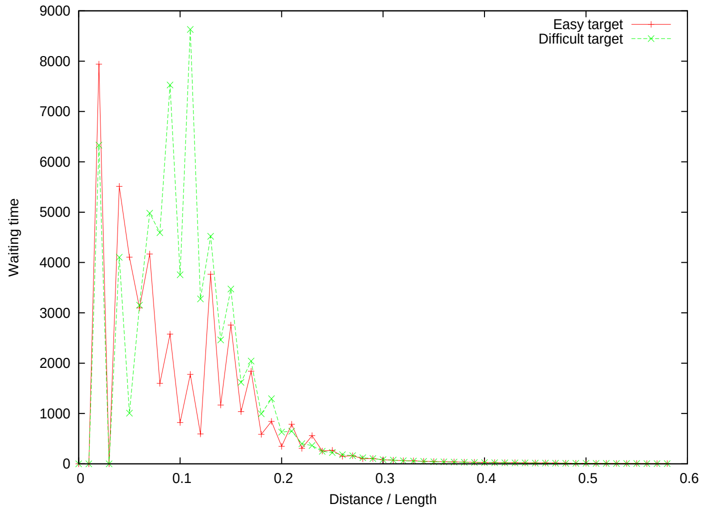

Figure 4.10: Plot of waiting times (= mean time spent on neutral set before a beneficial mutation is found) for a simulation of chains of length 100 with two different targets (see figure 4.8).

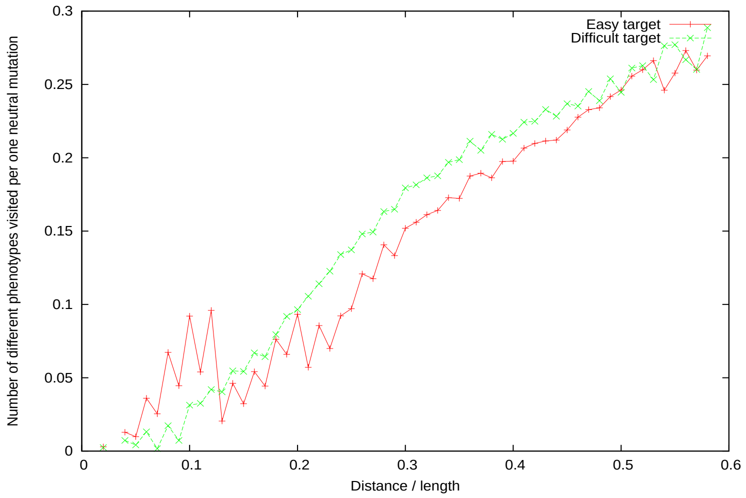

Figure 4.11: Number of different phenotypes visited per number of neutral mutations effected at a given distance for two different targets.

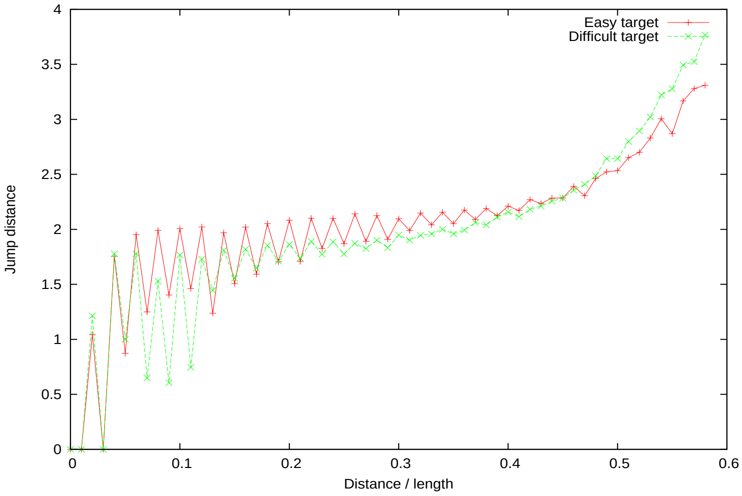

Figure 4.12: The mean jump distance, that is the distance by which we approach the target when we leave the neutral set whose distance is shown on x-axis (normalized).

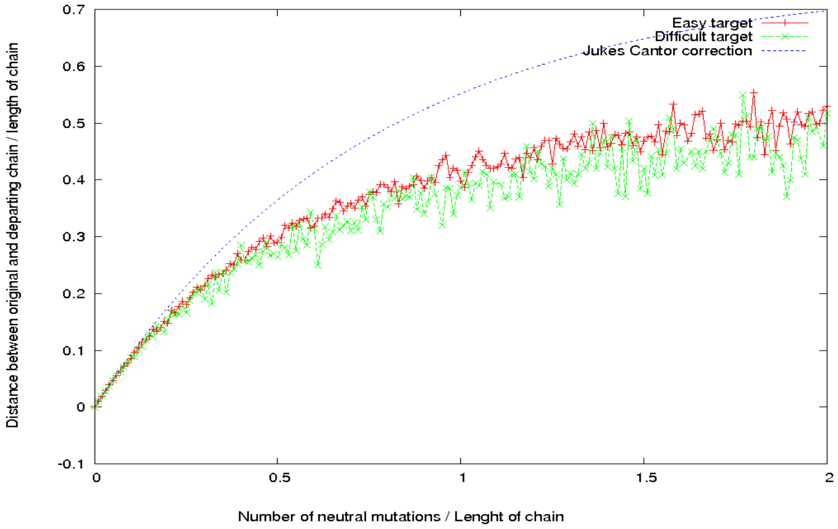

Figure 4.13: Mean normalized genotypic distance to genotype arising at the beginning of the current stasis period as a function of the number of accepted mutations for chains of length 100 (shown for two different targets) exhibiting fast diffusion in genotype space. Also shown is Jukes-Cantor correction, which is the function $f(x) = \frac{3}{4}\left(1 - e^{-\frac{4}{3}x}\right)$ used in evolutionary biology [15] to estimate the distance from the initial phenotype after x mutations have taken place. Our diffusion curves are slower which is not surprising, because the diffusion is bound to the neutral set.

It should also be noted that we observed variations between proprieties of neutral sets that have odd distance from the target and neutral sets that have even distance from the target. In general, it is more difficult to find a beneficial mutation on neutral set with odd distance and the waiting times are longer.

4.5 Normal diffusion during periods of stasis

Are the stasis periods associated with slow changes of genotype? It turns out that they are not, instead genotypes diffuse normally. The periods of stasis are characterized by slow phenotypic dynamics and during these periods many independent genotypes are visited before the walk finds a mutation that brings one closer to the target.

To test for this diffusion in genotype space, we have followed the distance between the current genotype and the one at the beginning of the stasis plateau. Because L is large, the initial growth in distance is linear in the number of accepted mutations (see figures 4.13 and 4.14).

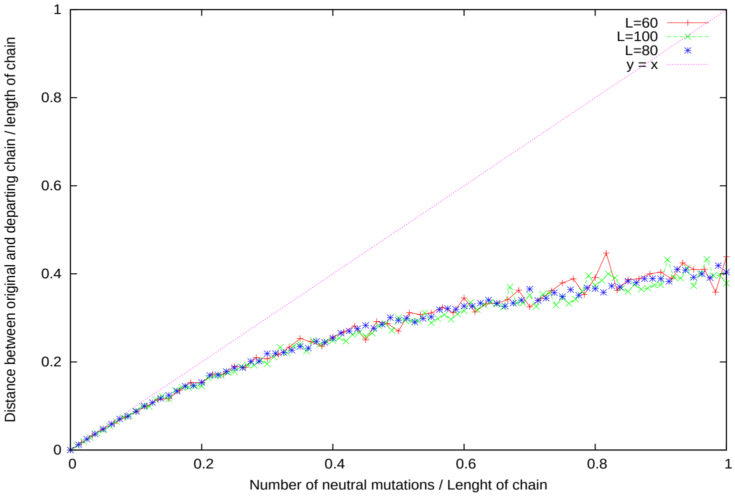

Figure 4.14: Mean genotypic distance to genotype arising at the beginning of the current stasis period as a function of the number of accepted mutations for chains of length 60,80 and 100, exhibiting fast diffusion in genotype space. The distances both on the x-axis and y-axis were normalized for each chain (i. e. divided by the chain's length). Also shown is the parameter-free linear law.

4.6 A mean field modeling

The objective here is to use modeling to obtain insights into the slow dynamical processes that we have observed. For this, we will replace the actual adaptive walk by a different system which is easier to understand and then compare to the true relaxation dynamics in the RNA fitness landscape.

The nested neutral sets allow for diffusion inside each set and for transitions to other sets closer to the target. Each neutral set is a union of neutral networks; the arguments developed in [16] for neutral networks apply also to neutral sets. These sets are thus expected to be connected and heterogeneous, and we have confirmed that they are so. This makes modeling the dynamics rather challenging as heterogeneous networks are always subtle objects.

The simplest modeling consists of neglecting the memory of the walk in each neutral set and simply considering that at each mutational step there is a given probability to perform a transition to a neutral set closer to the target. Input to our mathematical model are the fractions of deleterious, neutral, and advantageous mutations. Only these last ones are associated with transitions to better neutral sets. We have measured these transition rates for each distance to the target (cf. the uniform sampling of these sets as previously described). These rates define an effective Markov process towards the target space which we have encoded into Mathematica. Note that in such a modeling (where the memory of the walk inside a neutral set is neglected), the waiting times in each neutral set are exponentially distributed, a waiting time being precisely the period of stasis. As shown in figure 4.15, this is not too far from being the case in the real dynamics, at least for times shorter than the mean. When we use this "well mixed" process to reconstruct the relaxation to a given target, we find a discrepancy between the actual dynamics and the simplified dynamics. Upon investigation, we found that the reason for this was that the predicted waiting times were much smaller than the actual ones. This means that the memory of the walk is in fact very important, a feature which requires that the space in which the random walk occurs be irregular. In our context, this means that the neutral set is highly inhomogeneous: the vast majority of the nodes do not provide any exit transitions, and those that do are probably clustered and have many favorable mutations. The main consequence of this heterogeneity is that the walk wastes much of its time in unfavorable regions, and so the mean exit probability (when averaged over the whole neutral space) is not relevant: this "mean-field" approximation is too simplified.

4.7 Empiricial observation

The simplest approach to interpret the power law function is to find it as a solution of a Langevin equation for Brownian motion. For the average motion, we omit the

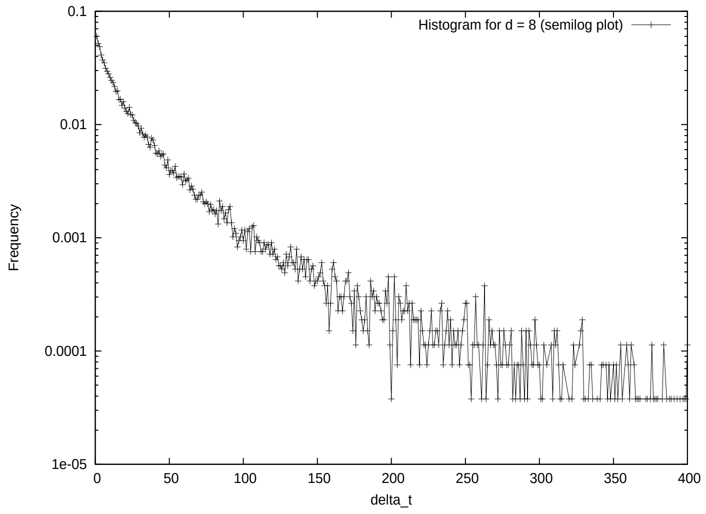

Figure 4.15: The distribution of waiting times between finding two beneficial mutations that lead to a phenotype of distance closer by 2 to the target phenotype. The distribution was obtained by several random walks that were performed in the neutral sets of phenotypes of distance d=8 from the target. The length of the RNA chain was 40. The plot is semilogarithmic.

random terms in the equation and therefore limit it only to an equation of a particle that is moving in a potential field:

$$\dot{d} = -f(d) \tag{4.2}$$

where $d(t)$ is a function of mean distance at a given time. The above equation cannot explain the shift t_0 nor the value of the parameter A in equation (4.1). $f(x)$ can be chosen empirically to give a value of the exponent that was observed in our fit. If we put

$$f(d) = d^{\alpha}, \quad \alpha > 1$$

we get a solution with integration constant C

$$d(t) = \left(\frac{1}{(t-C)(\alpha-1)} \right)^{\frac{1}{\alpha-1}} \tag{4.3}$$

given that α was chosen so that $\frac{1}{\alpha-1} = \beta$ where β is the value from the power-law fit. We see that this approach is not able to sufficiently explain all the parameters of the fit function (4.1) and is based only on an empirical observation that the closer we get to the target (the smaller d), the longer it usually takes to find a mutation

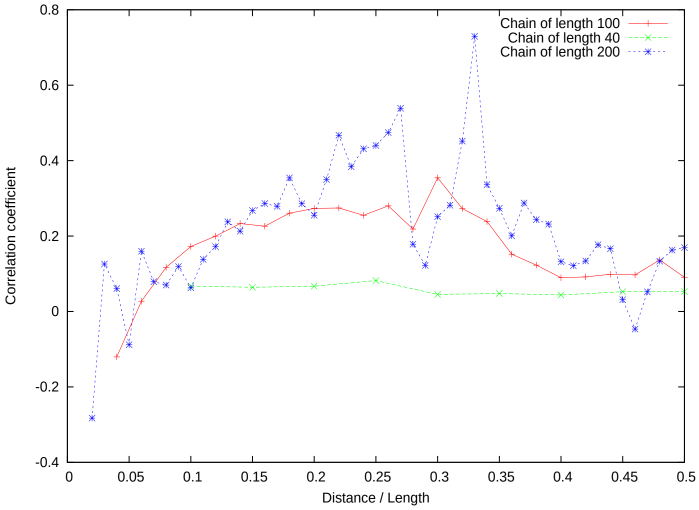

Figure 4.16: Correlations of waiting times between two favorable mutations, displaying nearly independent times. Longer chains have some regions with higher correlations. The correlation coefficients in the graph were calculated for favorable mutations arising at the distance shown on the x-axis and the following one at the distance smaller by 2.

that would approach the target structure.

4.8 A random connection approximation

We are confronted with non-trivial random walk effects in each neutral set. One could take a random graph modeling of these sets, but this would not lead to fat tails in the distribution of waiting times. The properties of random walks on heterogeneous graphs has been considered in the literature, for instance in scale free cases. For our purposes here, we shall simply take as given these distributions rather than try to approximate them. What then is left to quantify? We must consider that after a transition has arisen, a new period of stasis begins. Are the different times of stasis statistically independent? We have seen in figure 4.16 that their linear correlation coefficient is small, at least for short chains; this leads us to consider that an acceptable modeling might be obtained by assuming that the transitions from one neutral set connect to random nodes of the closer neutral sets. This assumption is particularly easy to analyze as it means that the beginning genotype in each period of stasis is taken uniformly in the associated space. Thus the approximation scheme in this part is to treat the random walk in each neutral set exactly, but to randomize the *landing* sites of the transitions.

To test our model, we created a histogram of times δt for a given distance d. The δt were obtained by performing 10 random walks through the neutral set and counting the times (number of mutations) between finding two beneficial mutations. Each random walk consists of 30000 steps through the neutral set. We then suppose that the landing on the neutral set is random (uniform). The random walk then proceeds until the first beneficial mutation is found. There are δt "places" between two beneficial mutations where the landing of the random walk can arise. Given such a position, we mark the time to start counting the number of mutations before reaching a beneficial one as zero; then if the random walk starts at position i, the time spent on the neutral set will be $\delta t - i$. Therefore, we have to average over all possible times i where the random walk can start and average over all the sampling random walks that we have performed. The "Monte-Carlo estimate" for waiting time T_w at given distance d is then obtained as

$$\begin{aligned} T_w &= \frac{(\delta t_1 + (\delta t_1 - 1) + (\delta t_1 - 2) + \ldots + 1) + \ldots + (\delta t_n + (\delta t_n - 1) + (\delta t_n - 2) + \ldots + 1)}{\delta t_1 + \delta t_2 + \ldots + \delta t_n} = \\ &= \frac{1}{2}\left(\frac{\langle \delta t^2 \rangle}{\langle \delta t \rangle} + 1\right) \end{aligned}$$

where δt_i runs over all times that were measured during our random walk sampling.

How well does this approximation work? We have constructed the mean waiting time as a function of distance to target for both the actual evolutionary dynamics and for this random connection approximation. It is important to note that in the simulation as well as in our random sampling that was used to construct the model, we allowed only those beneficial mutations that decrease the distance by two. This was done to simplify the otherwise complex dynamics when longer jumps appear as well as to avoid the problems that would have arisen if we had to deal with neutral sets with odd distances to the target (where a generally longer time is needed to find a beneficial mutation). The resulting curve is shown in figure 4.17. We see that there is a quantitative disagreement while the qualitative aspects of the predictions are good. Of course deviations are clear; these show that for the evolutionary dynamics the landing genotype is not uniformly distributed in the neutral space. We find in fact that the evolutionary dynamics is slower, which may mean that the non-equilibrium processes lead to genotypes at the beginning of the stasis periods that are less evolvable than random.

4.9 Are the generated phenotypes atypical?

Given the long diffusion times at a given distance realized in the periods of stasis, the genotypes are expected to be randomized and lose memory of the entry genotypes

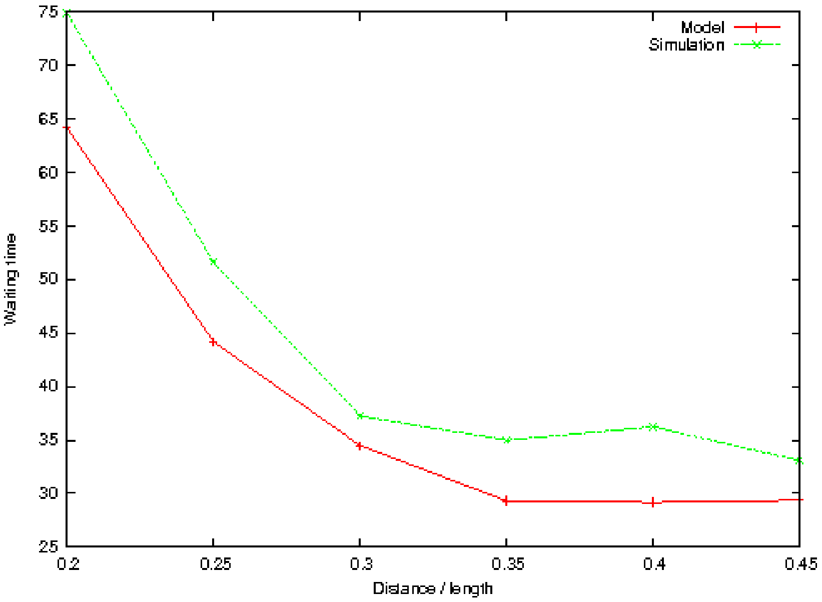

Figure 4.17: The points labeled as "Simulation" refer to mean waiting times at a given distance for a simulation of a chain of length 40 (with fixed target structure and averaged over random initial chains). The curve labeled as "model" refers to the quantity $\frac{1}{2}\left(\frac{\langle\delta t^2\rangle}{\langle\delta t\rangle}+1\right)$ where δt refer to the time between finding two beneficial mutations during a random walk on the neutral set at given distance (see text).

(the genotype occurring at the beginning of the stasis period). However the exit genotype is the one which produces the end of the stasis and often the single mutation which takes one closer to the target initiates another period of stasis. Thus it is plausible that the initial genotype and phenotype of a period of stasis is atypical. In the previous section we used an approximation of the dynamics where this initial genotype was uniformly taken from the space at a given distance to the target; we found that this hypothesis led to an underestimation of the time to approach the target. We thus hypothesized that the initial genotype has the property of being less "evolvable" than a random one at the same distance to the target. This certainly means that the corresponding phenotype is also atypical (note that the phenotype evolves very slowly, so that the atypical behavior is not restricted to just the first few steps in the stasis plateau.)

To quantify this property, we have studied the *short term evolvability* of the initial genotype in a period of stasis and compared it to that of random genotypes at the same distance to the target. We define the short term evolvability as the fraction of mutations that lead to a higher fitness. We found that the fraction of beneficial mutations is actually lower for the first genotype on the neutral set than for the set average, as can be seen in figure 4.19. We also find that the first genotype has

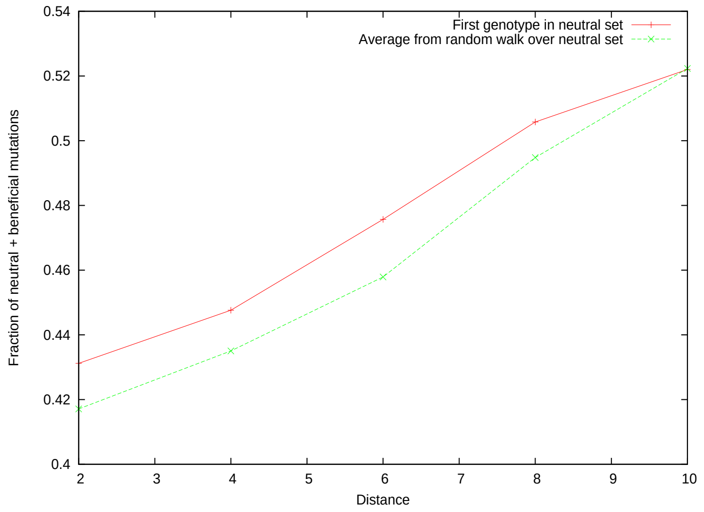

Figure 4.18: The fraction of beneficial plus neutral mutations. The statistics for the first genotype of the neutral set was averaged over 4000 random walks that lead from a randomly generated phenotype to the fixed target structure. The simulation was stopped when a neutral set of given distance was reached and statistics for fractions of beneficial / neutral mutations was then computed for the first genotype of the neutral set. The statistics for the neutral set was averaged over 10 random walks on the neutral set, each containing 30000 mutations. All simulations were performed with chains of length 40.

a smaller number of deleterious mutations than the neutral set average, as shown in figure 4.18. Our sampling of the neutral set also revealed a high dispersion of the number of beneficial mutations in each genotype. This further confirms the heterogeneity of the neutral sets and complicates the goal of finding a model that would be able to adequately describe the evolutionary dynamics of RNA evolution.

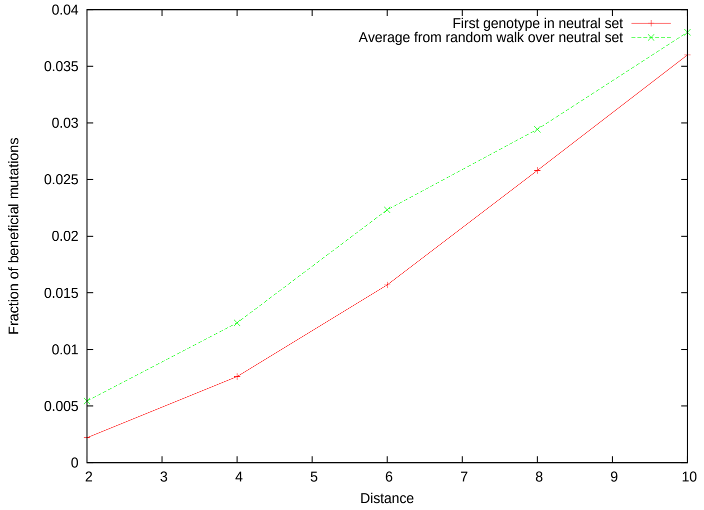

Figure 4.19: The data shown were obtained from the same set of simulations as data shown in figure 4.18. Here, we display only the fraction of beneficial mutations.

Chapter 5

Mean first passage time on a random graph

In this chapter, motivated by the walks on RNA neutral sets, we introduce a model of walks on random graphs with an absorbing node. We begin by introducing terminology from graph theory [5] that will be used in this chapter.

5.1 Graph theory preliminaries

A graph is given by two sets V and E, where the elements of set V are called *vertices* (or *nodes*) and the elements of the set E are *edges* (or *lines*). An edge is a pair of two nodes and denotes that the two nodes are connected. For example, a graph with $V = \{1, 2, 3\}$ and $E = \{\{1, 2\}\}$ denotes a graph with nodes 1 and 2 connected by the edge $\{1, 2\}$ and node 3 isolated. We will be dealing only with finite undirected graphs, meaning that edges $\{i, j\}$ and $\{j, i\}$ are equivalent and V and E are finite sets. An undirected graph such that there is at most one edge connecting two nodes and no node is connected to itself [1] is called *simple*. We will only deal with simple finite graphs.

Nodes i and j are called *adjacent* (neighbors) if there is an edge $\{i, j\}$ in the set E. The *adjacency matrix* of a simple graph is a symmetric matrix A whose elements A_{ij} are equal to 1 if the vertices i and j are adjacent, otherwise A_{ij} is equal to zero.

The *degree* of a node is the number of its neighbors. If every node has d neighbors, the graph is called d-regular.

A *connected* graph is a graph such that any two nodes i, j are linked by a path. A *path* in a graph is a list of nodes $x_1, x_2, \ldots, x_n$ such that $\forall i < n : x_i$ and $x_i + 1$ are adjacent. We say that a graph is k*-connected* if in order to disconnect it, the smallest number of nodes you have to remove is k.

A *cycle* in a graph refers to path in the graph with no repeated nodes except the

[1] This means that for all edges $\{i, j\} : i \neq j$

starting and final one which are identical. A graph that doesn't contain any cycles is called *acyclic.* A connected acyclic graph is called a *tree.*

Cayley tree is a connected acyclic regular graph.

Random graphs are graphs that are generated by a random process. In a case of a d-regular random graph, we suppose that each node is connected to other d randomly chosen nodes. The most commonly studied random graph is an Erdös-Rényi graph, denoted as $G(n, p)$, where n is the total number of nodes and each pair of nodes has probability p to be connected.

5.2 Neutral networks of the RNA secondary structures

To characterize the neutral networks of the secondary structures of RNA, we first performed several simulations to better understand the structure of the graph with nodes that correspond to a given phenotype. That means that we considered only neutral networks, not neutral sets. To estimate the structure of the neutral networks, we generated a random chain of a given length L and then considered all possible $3L$ mutations (since each base can mutate into 3 other bases). The number of mutations that leave the phenotype unchanged is then the degree of the node, i.e. the number of its neighbor genotypes in the neutral set. We generated 1000 different genotypes for each chain length. The distribution of degrees for different chain lengths is shown in figure 5.1. We see that the mean degree as well as the degree dispersion increases with higher chain length.

Another important property that is used in assessing networks [17] is assortativity. The assortativity of a node i is defined as

$$k_i = \frac{1}{d_i} \sum_{j=1}^{N} A_{ij} d_j \tag{5.1}$$

where A_{ij} are the coefficients of the graph's adjacency matrix and d_j is the degree of the j-th node. The assortativity of the node is then a sum of degrees of all neighbors divided by the degree of the node. An assortative network is such a network where the node's assortativity increases with increasing degree, meaning that nodes with high number of neighbors are connected to nodes which have many neighbors themselves (a property found for example in social networks [17]). Figure 5.2 shows assortativity as a function of the node's degree for chains of length 80 and 100. The neutral network of an RNA phenotype is therefore assortative, as the plots show increasing functions. A neutral network is also heterogeneous, as we already observed during our simulations of RNA chain evolution.

Another property of a network is provided by its clustering coefficient. The

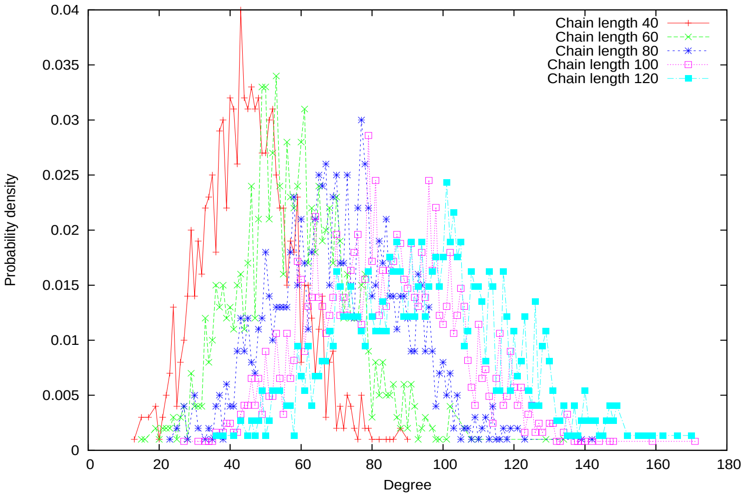

Figure 5.1: The histograms of degree of a node in a neutral network of a phenotype of a given length.

clustering coefficient of the i-th node is

$$c_i = \frac{1}{d_i(d_i-1)} \sum_{j,h=1}^{N} A_{ij}A_{ih}A_{jh} \tag{5.2}$$

where A_{ij} again denotes an element of adjacency matrix. The product $A_{ij}A_{ih}A_{jh}$ equals 1 if j-th and h-th nodes are connected and they are both neighbors of the i-th node. Coefficient c_i is then the number of triangle cycles that include the i-th node divided by $d_i(d_i-1)$ so that

$$0 \leq c_i \leq 1.$$

Our simulations show that clustering coefficients (figure 5.3) for neutral networks are small and therefore triangle cycles are rather rare.

We decided to approximate the structure of a neutral network by a random graph. For simplicity, we studied graphs with each node having the same degree d (d-regular graphs). Moreover, we will consider only connected graphs. It can be shown that the probability of such a graph not being d-connected is $O(1/N^{d-2})$ for $d \geq 3$ [18] and therefore by accepting only the connected graphs with degree $d \geq 3$, we do not exclude any important class of d-regular random graphs.

Although this random graph approximation is unable to describe heterogeneity of the neutral network, it is a model that allows us to estimate the first passage time,

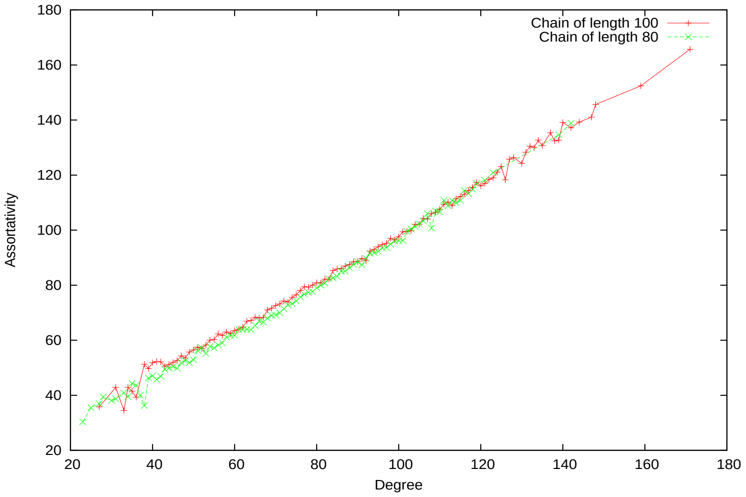

Figure 5.2: The assortativity of a node as a function of its degree. The plot shows the relation for chains of length 80 and 100 respectively.

that is a mean time needed to visit a particular node for the first time. We call such a node “absorbing” because visiting such node results in leaving the network and moving to a different phenotype.

5.3 Neutral networks and modeling with random graphs

The time spent on a neutral network is linked to a first passage time $H(s,t)$ (sometimes also called hitting time), which denotes a mean number of steps on the network (mean time) that is necessary to get to a target node t while starting in a departing node s. We will present two different approaches to calculate the first passage time $H(s,t)$ for a random graph, which we take to be an approximation of RNA neutral networks. We start by presenting a relation between the value of $H(s,t)$ averaged over all nodes of the graph and the spectrum of its adjacency matrix as calculated in [19] and apply it to the case of a d-regular random graph. Then, we show an alternative approach, based on the diffusion equation on the graph. We compare the prediction of the two approaches with numerical simulations of random walks on random regular graphs.

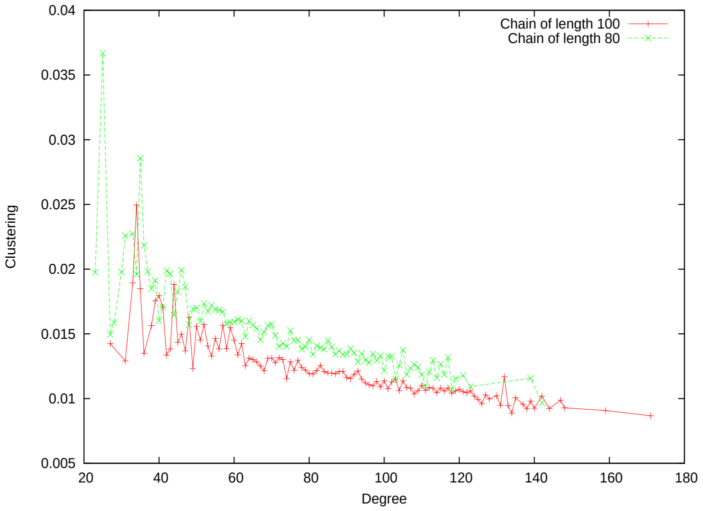

Figure 5.3: The clustering coefficient of a node as a function of its degree. The mean number of degrees for chains of length 80 is about 71. The mean degree of a chain of length 100 is about 84, therefore the clustering coefficient is rather small, meaning that triangle cycles appear rarely in neutral networks.

5.4 First passage time and spectrum of the adjacency matrix

5.4.1 A general relation

The topology of a graph is described by its adjacency matrix A. Another matrix that is often studied in relation to the structure of a graph is

$$N = D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \tag{5.3}$$

where D is a diagonal matrix such that its i-th diagonal element D_{ii} is equal to the degree d_i of the i-th node. A random walk on a graph is described by a stochastic matrix $M = AD^{-1}$, since the evolution of vector $\mathbf{v}$ whose i-th element corresponds to the probability of occupying i-th node is

$$\mathbf{v}_i = \sum_{<ji>} \frac{1}{d_j} \mathbf{v}_j = \left(AD^{-1}\mathbf{v}\right)_i \tag{5.4}$$

where the sum is taken over all nodes j that are adjacent to the node i. For the case of a regular (i.e., all nodes having the same degree) graph, matrix M is equal to N, but in a general case (i.e., variable number of edges in each node), M is not a symmetric matrix. In the case of a d-regular graph, the eigenvalues of matrix N are just $\frac{1}{d}$ multiples of eigenvalues of the adjacency matrix A and N is just

$$N = \frac{1}{d}A. \tag{5.5}$$

According to the Perron-Frobenius theorem [12], the largest eigenvalue of the matrix N is $\lambda_1 = 1$ and all other eigenvalues satisfy

$$|\lambda_k| \leq 1.$$

If the adjacency matrix is irreducible (which is equivalent to the graph being connected, that is any node is reachable from any departing node), all other eigenvalues are strictly smaller than λ_1. We will consider only the case of connected graphs in the rest of the chapter.

It can be shown [19] that the mean first passage time $H(s,t)$, when averaged over all possible target nodes, can be expressed in terms of the spectrum of the matrix N as follows:

$$H(s) = \sum_t \pi(t)H(s,t) = \sum_{k=2}^{n} \frac{1}{1-\lambda_k} \tag{5.6}$$

where λ_k are the eigenvalues of matrix N and $\pi(x)$ is a stationary probability distribution on the nodes of the graph, $\pi(t)$ being the probability to occupy node t. One can check that on a regular graph, $\pi(t) = \frac{1}{n}$ where n is the number of nodes. Note that the eigenvalue $\lambda_1 = 1$ is not included in the sum (5.6). Since $H(s)$ in equation (5.6) does not depend on the departing node s, it follows that $H(s)$ is equal to the mean first passage time averaged over all departing nodes.

5.4.2 Spectrum of random graphs

As we mentioned before, the Perron-Frobenius theorem implies that the largest eigenvalue is equal to one and that the other ones are smaller. However, in order to estimate the value of first passage time (5.6) we need to know all eigenvalues. We are therefore interested in estimating eigenvalues of a randomly generated d-regular graph. The ensembles of random matrices have been closely studied with relation to numerous problems in quantum physics and chaotic dynamics. It has been proved by Wigner that the distribution of eigenvalues of the Hermitian matrices whose elements follow a Gaussian distribution follows a semicircle law

$$f(\lambda) = \frac{2}{\pi R^2}\sqrt{R^2 - \lambda^2}, \quad |\lambda| \leq R \tag{5.7}$$

as the dimension of the matrix approaches infinity [20]. However, randomly generated adjacency matrix contains only 1 and 0 elements and in the case in which we are interested is sparse.

We will now consider the spectrum of matrix N as defined in (5.5). We know that its largest eigenvalue λ_1 is equal to 1. Concerning the other eigenvalues, it has been proved by F. Chung et al. [21] that the distribution function of eigenvalues $\lambda_i \neq 1$ converges to a semicircle law for large dimension of the adjacency matrix (i. e. large number of nodes of the graph) given the condition that

$$\langle d \rangle \gg \sqrt{\langle d \rangle}$$

where $\langle d \rangle$ is the average degree of a node. For the case of a d-regular graph, $\langle d \rangle = d$. The estimate of the radius of the semicircle distribution function (5.7) is then

$$R \approx \frac{2}{\sqrt{\langle d \rangle}}. \tag{5.8}$$

To obtain a mean value of H, we compute the mean value of the sum (5.6) using the semicircle probability distribution of its eigenvalues. Note that the trace of the matrix N is equal to zero, which means that the sum of the eigenvalues that are distributed according to the semicircle law is

$$\sum_{i=2}^{n} \lambda_i = -1 \tag{5.9}$$

The mean value of the semicircle distribution function therefore has to be shifted by $\frac{1}{n-1}$. For our calculation of mean first passage time H for a d-regular random graph, we will use a distribution function

$$f_n(\lambda) = \begin{cases} \frac{2}{\pi R^2}\sqrt{R^2 - (\lambda + x_n^0)^2} & \text{if } |\lambda + x_n^0| \leq R \\ 0 & \text{otherwise} \end{cases} \tag{5.10}$$

with

$$x_n^0 = \frac{1}{n-1} \quad \text{and} \quad R = \frac{2}{\sqrt{d}} \tag{5.11}$$

The mean value of H (averaged over the space of all graphs) is then given by

$$\langle H \rangle = \sum_{k=2}^{n} \int\limits_{[-R-x_n^0, R-x_n^0]^{n-1}} \prod_{i=2}^{n} \left(d\lambda_i f_n(\lambda_i)\right) \frac{1}{1-\lambda_k} \delta\left(\sum_{j=2}^{n} \lambda_j + 1\right). \tag{5.12}$$

For a numerical calculation, we simplify the above equation by omitting the Dirac function δ which should introduce no error in the large n limit:

$$\langle H \rangle = (n-1) \int_{-R-x_n^0}^{R-x_n^0} d\lambda \frac{f_n(\lambda)}{1-\lambda}. \tag{5.13}$$

Inserting (5.10) into the previous equation, we get

$$\langle H \rangle = \frac{2(n-1)}{\pi} \int_{-1}^{1} dx \frac{\sqrt{1-x^2}}{1+x_n^0 - Rx} \tag{5.14}$$

Integration gives

$$\langle H \rangle = 2(n-1) \frac{1+x_n^0 - \sqrt{(1+x_n^0-R)(1+x_n^0+R)}}{R^2} \tag{5.15}$$

where $x_n^0 = \frac{1}{n-1}$ and $R = \frac{2}{\sqrt{d}}$. We can express equation (5.15) in terms of d as

$$\langle H \rangle = (n-1)\frac{d}{2}\left((1+x_n^0) - \sqrt{(1+x_n^0)^2 - \frac{4}{d}} \right), \tag{5.16}$$

which could be further simplified in limit of large number of nodes n and high degree d as

$$\langle H \rangle \approx n-1. \tag{5.17}$$

Figure 5.4 shows the comparison between the spectrum distribution of 500 randomly generated 10-regular graphs with 400 nodes compared with a semicircle distribution (5.10) for $d = 10$ and $n = 400$. A small systematic disagreement is visible which is due to the fact that d is finite, i.e., the graphs considered are sparse rather than dense. Thus equation (5.17) is exact only for dense graphs, while the next sections will provide an exact answer for sparse graphs. A comparison with a numerical simulation of first passage time will be provided in section 5.7. The estimation of the radius of the semicircle distribution is given by (5.11).

5.5 First passage times on random regular graphs

An exact analytic solution of the mean first passage time on random regular graphs can be obtained by realizing that such graphs are locally tree-like. More explicitly, for any given degree d, loops can arise in random regular graphs but their typical length is $O(\ln(n))$. Thus it is expected that most properties can be obtained by studying what happens locally, as long as boundary conditions at "infinity" are properly handled. Such procedures have been used in many contexts with a high level of success [22].

For a given random regular graph, of fixed degree d, we consider a node t and ask

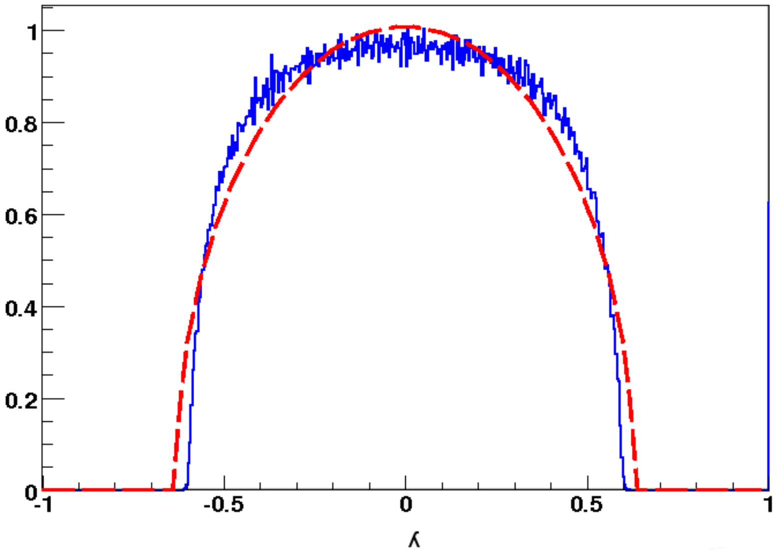

Figure 5.4: A semicircle distribution (shown in red) of eigenvectors as presented in (5.10) (with $d = 10$ and $n = 400$) compared with a histogram of eigenvalues of matrix $\frac{1}{10}A$, where A is a adjacency matrix of 10-regular graph of size $n = 400$. The histogram (shown in blue) was obtained by numerical diagonalization of 500 randomly generated 10-regular graphs.

what is the mean of $H(s,t)$ when averaging over all s. This can be formulated in terms of a diffusion problem where at time 0 a walker is equi-distributed amongst the $n-1$ nodes s ($t \neq s$) and all walkers hitting t get absorbed (disappear). The evolution with time of the probability to find the walker on a given site allows one to obtain $\langle H(s,t)\rangle_s$ by considering the influx into node t as a function of time. If one denotes by $F(\tau)$ the flux into node t at time τ, then the first passage time averaged over all s is given by the first moment of τ distributed as $F(\tau)$.

In the neighborhood of t, the graph is a Cayley tree with probability one at large number of nodes n and thus does not depend on t in the large n limit. We thus study the diffusion on this tree with the walker initially distributed uniformly. The boundary condition is that the probability of finding the walker at t is zero at all times (absorbing site).

The vector of probabilities on each site quickly converges to the dominant eigenvector of the evolution equation (that with the largest eigenvalue, decaying the slowest). In the limit of large n, the decay rate goes to zero and all the transient behavior (associated with the other eigenvectors) becomes irrelevant. When $n \to \infty$, it is then enough to consider the solution to the discrete evolution equation on the tree, where one has zero boundary conditions at t and the probability goes to $\frac{1}{n-1}$ for the far away nodes.

The recurrence equation that is then satisfied by the vector's elements is

$$A_{n+1} = \frac{1}{d} A_{n+2} + \frac{d-1}{d} A_n \tag{5.18}$$

where A_n is a sum of all probabilities on all nodes that have distance n from the root node:

$$A_n = \sum_{\text{distance}(j,0)=n} a_j \tag{5.19}$$

where a_j are elements of the state vector and their values correspond to the probabilities of occupying the j-th node. The general solution of (5.18) is

$$\alpha(d-1)^n + \beta. \tag{5.20}$$

Given the conditions of absorption at the root node

$$A_0 = 0$$

and the total probability of occupying the nodes of graph equal to one (where l is total number of layers):

$$\sum_{k=0}^{l} A_k = 1$$

we obtain a solution for A_1

$$A_1 = \frac{(d-2)^2}{(d-1)^{l+1} - 1 - (l+1)(d-2)}. \tag{5.21}$$

The rate at which one falls into the absorbing node $a_0 = A_0$ is then $p = \frac{1}{d} A_1$.

Note that since at large n only this eigenvector matters, the first passage time τ is exponentially distributed with a mean given by the inverse of this rate. This gives then for the mean first passage time

$$H = \frac{1}{p} = \frac{d-1}{d-2} n + \frac{1}{(d-2)} - \frac{d}{(d-2)} (l+1). \tag{5.22}$$

The relation between the number of layers l and total number of nodes is

$$n = \frac{d(d-1)^l - 2}{d-2} \tag{5.23}$$

so we can express l as

$$l = \frac{\ln\left(\frac{n(d-2)+2}{d}\right)}{\ln(d-1)}. \tag{5.24}$$

For large n, the dominant term in equation (5.22) is proportional to n. Then on a random regular graph of connectivity d, the mean first passage time behaves at large n as

$$H = \frac{d-1}{d-2}n + O(\ln n)\ . \tag{5.25}$$

5.6 Numerical analysis

To compare the analytical calculations for the estimate of the first passage time, we ran a numerical simulation of the d-regular graph for different sizes n.

5.6.1 Absorbing walk

One of the possible approaches to calculate the mean first passage time is to perform an iteration of the probability distribution vector whose entries correspond to the probability of occupying a selected node. The iteration of the vector then corresponds to the stochastic process described by equation (3.1). We calculate the first passage $H(s,t)$ time by multiplying the probability of occupying the target node at i-th iteration by i:

$$H(s,t) = \sum_i ip(t). \tag{5.26}$$

After each iteration, we set the probability to occupy the target node equal to zero. This simulation process is equivalent to applying matrix TM on the probability distribution vector

$$\mathbf{v}^k = (TM)^k\,\mathbf{v}^0, \tag{5.27}$$

where matrix T is equal to identical matrix with zero on diagonal element that corresponds to the absorbing node. M is stochastic matrix for random walk on graph as seen in equation (5.4). At each step, we measure the probability to remain on the graph (which is equal to the sum of the elements of the vector $\mathbf{v}$) and once it is lower than a given bound (we used 10^{-8}), the calculation is terminated. We run the simulation multiple times, each time choosing an arbitrary departing and absorbing node (separated at least by distance $\ln(n)$) and then average the first passage time over all graphs produced in the runs.

5.6.2 Iteration method

Since the absorbing walk algorithm was too much time-consuming, we decided to complement the absorbing walk simulation by the fixed point method for calculation of the eigenvector corresponding to the largest eigenvalue.

The stochastic matrix M has the leading eigenvector that corresponds to the uniform distribution $\mathbf{v}_i = \frac{1}{n}$ with eigenvalue 1. However, the largest eigenvalue of the matrix TM will be smaller than one.

We calculate the eigenvector corresponding to the largest eigenvalue by the iteration method

$$\mathbf{v}^{k+1} = \frac{TM\mathbf{v}^k}{\left\|TM\mathbf{v}^k\right\|_1}$$

where the norm used is l_1:

$$\|\mathbf{v}\|_1 = \sum_{i=1}^{N} |\mathbf{v}_i|. \tag{5.28}$$

The departing node is chosen arbitrarily. The simulation is terminated when the maximal difference

$$\delta_{max} = \max_{i \in \widehat{n}} \left|\mathbf{v}_i^{k+1} - \mathbf{v}_i^k\right|$$

is smaller than a desired threshold (10^{-8} in our simulations). We then calculate the probability of falling into the absorbing node j given the probability distribution vector v as

$$p_j = \sum_{i \in <ij>} \frac{1}{d}\mathbf{v}_i \tag{5.29}$$

where the sum is taken over all neighbors of the absorbing node j. The mean first passage time is then estimated as $H = \frac{1}{p_j}$.

We supposed that for the matrix TM in normal Jordan form, the Jordan blocks corresponding to eigenvalues that are smaller than the maximal one will become negligible after a few iterations and therefore it will be only the leading eigenvector that will determine the flow of probability. Thus, we can replace the calculation of the first passage time from the previous section by the value calculated from the probability flow (5.29). The simulation data indeed show that this approach gives predictions compatible with those of the absorbing run approach. The iteration method is much faster than the absorbing walk method, which allows us to perform relatively fast simulations also for the graphs having several thousand nodes. The values predicted by both methods are plotted in figure 5.5.

5.7 Comparison of models

Hereafter, all numerical simulations were obtained using the iteration method (described in section 5.6.2) to calculate the first passage time. A brief description of simulation code is given in appendix A.2. Figure 5.6 shows the ratio between the first passage time calculated by the iteration method and the prediction of Cayley tree calculation in (5.22). We see that the ratio approaches one for large size graphs.

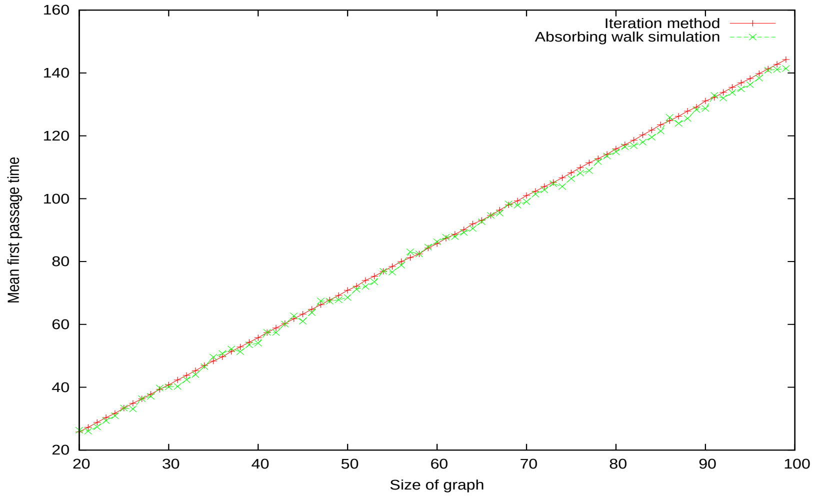

Figure 5.5: Comparison of the iteration method and the absorbing method for 4-regular graphs of different sizes.

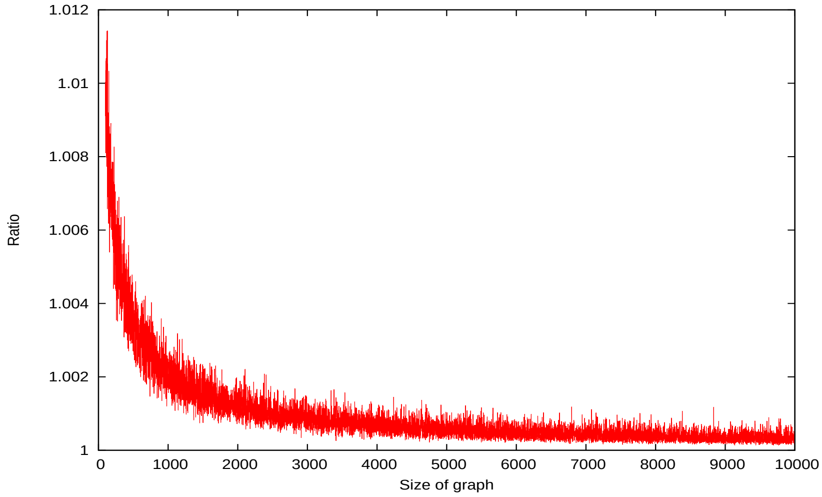

Figure 5.6: The ratio between the first passage time calculated by the iteration method simulation and that using the Cayley tree approach (equation (5.22)) for 10-regular graphs of different sizes. The first passage time calculated by the iteration method was averaged over 20 simulations of different randomly generated 10-regular graphs of the same size.

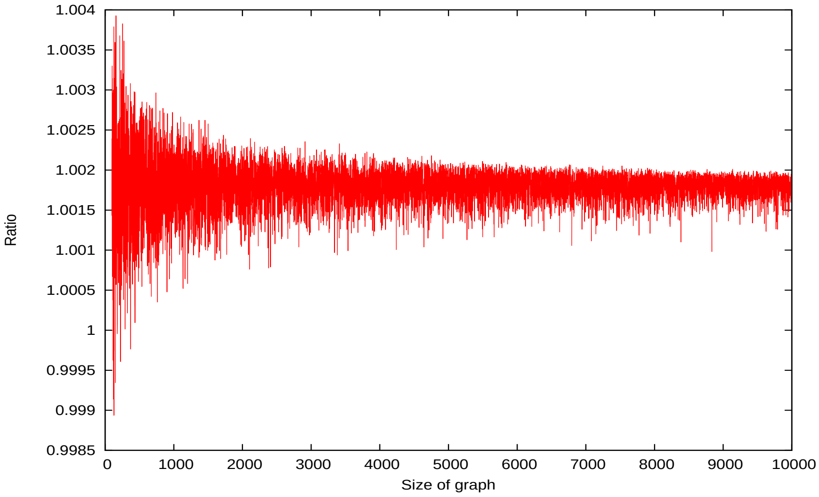

Figure 5.7: The ratio between the mean first passage time as estimated by the calculation of spectrum of random graph (equation (5.16)) and by a value obtained from a numerical simulation by iteration method for a graph of given size n (x-axis). The data were produced for 10-regular graphs.

The ratio between the first passage time estimated by calculation of the spectral distribution of random 10-regular graphs and the first passage time obtained by iteration method simulation is shown in the figure 5.7. Although the ratio doesn't seem to converge towards one, the difference between the two models is smaller than one percent. Considering that the semi-circle law we used is only valid in the limit of large degree d and large n, the results are satisfying.

5.8 Multiple absorbing nodes

Since our study of first passage time was inspired by the study of the time spent on a neutral network, we should also discuss the case when multiple absorbing nodes exist, as is often the case of a neutral network. We therefore consider here that we have in total m possible sites where we can leave the graph of size n. We will show that the mean time spent on the graph before visiting any of the m absorbing nodes is equal to $\frac{1}{m}$ of the mean first passage time of the graph with one absorbing node.

We use the fact that the probability distribution to stay on the graph (i.e. not visit the absorbing node) follows an exponential distribution (cf. previous results).

The probability to leave the graph through the i-th node later than at time t_0 is

$$\int_{t_0}^{\infty} P(t)dt \qquad (5.30)$$

with

$$P(t) = e^{-\frac{t}{\tau}} \tag{5.31}$$

where τ is equal the mean first passage time H for a graph with one absorbing node. The probability of not being absorbed by any of the m independently selected absorbing nodes before time t_0 is

$$\prod_{i=1}^{m} \int_{t_0}^{\infty} P(t)dt = \left(\int_{t_0}^{\infty} P(t)dt \right)^m = e^{-\frac{t_0 m}{\tau}} \tag{5.32}$$

The probability to leave the graph later than after t_0 is

$$\int_{t_0}^{\infty} Q(t)dt \tag{5.33}$$

where $Q(t)$ is a probability distribution that we want to determine. Comparing (5.33) with (5.32) we get

$$\int_{t_0}^{\infty} Q(t)dt = e^{-\frac{t_0 m}{\tau}} \Rightarrow Q(t) = \frac{m}{\tau} e^{-\frac{mt}{\tau}} \tag{5.34}$$

from which it follows that the mean leaving time of the graph τ_L is

$$\tau_L = \frac{\tau}{m} = \frac{H}{m}. \tag{5.35}$$

This conclusion is in agreement with results that we have obtained when performing a simulation of absorbing walk (as described in section 5.6.1) with multiple absorbing nodes.

5.9 First passage times and return probability on random graphs

In this section, we would like to estimate the mean first passage time on a random graph. We will generalize the model of finite Cayley tree from section 5.5 and show how the mean first passage time is linked to the probability of return to an origin. We will assume that the graph is a tree, i.e., no loops are present. Furthermore, we consider large number of nodes $n \to \infty$. While in the previous model, we considered only regular graphs, we will also treat graphs where the degree of a node follows a probability distribution with mean $\langle d \rangle$. We assume Poisson distribution of the degree of each node. Our model is sufficient to describe a walk on a random Erdös-Rényi graph, since the distribution of degree of a node approaches Poisson distribution for large n and there are no cycles of given size with probability 1. In the following paragraph, we establish a connection between the first passage time and probability of return to the departing node and then describe an algorithm used to compute the distribution of return times on such a graph. In the rest of the section, we will suppose that the first node (meaning that the first element of the probability distribution vector corresponds to this node) is the absorbing node.

We suppose we have a tree with n nodes and $n \to \infty$. We want to estimate the steady state flux of probability into the absorbing node which gives the inverse of the mean first passage time. We begin with an initial vector $\mathbf{v}^0$ whose i-th element corresponds to the probability of occupying i-th node. The initial condition is that every node is occupied with the same probability $\frac{1}{n-1}$ except for the absorbing node which has the probability 0. During the absorption process (that is subsequent application of matrix $TM = TAD^{-1}$, defined in section 5.6.1, on the state vector), the probability of occupying node i after time k (k iterations) is

$$\mathbf{v}_i^k = \left((TM)^k \, \mathbf{v}^0 \right)_i . \tag{5.36}$$

In our case, T is a diagonal matrix

$$T = \operatorname{diag}(0, 1, 1, \dots, 1) . \tag{5.37}$$

We will denote $\mathbf{s}$ the normalized eigenvector of stochastic matrix M corresponding to eigenvalue 1. It is

$$\mathbf{s}_i = \frac{d_i}{\langle d \rangle n} \tag{5.38}$$

where d_i is the degree of the i-th node. We introduce vector $\mathbf{b}^k$ that represents the difference between the stationary distribution of matrix M and the vector $\mathbf{v}^k$:

$$\frac{1}{n}\mathbf{b}_i^k = \frac{d_i}{\langle d \rangle n} - \mathbf{v}_i^k . \tag{5.39}$$

Since after each application of the matrix TM the probability of occupying the absorbing node is zero, that is $\mathbf{v}_1^k = 0$, we get

$$\mathbf{b}_1^k = \frac{d_1}{\langle d \rangle} \quad \forall k . \tag{5.40}$$

Since we treat the graph size $n \to \infty$, after an initial relaxation the probability distribution vector $\mathbf{v}^k$ satisfies

$$\mathbf{v}_m^k = \frac{d_m}{\langle d \rangle n} \tag{5.41}$$

where the index m corresponds to nodes whose distance from the absorbing node dist$(1, m) \to \infty$. From equation (5.41), we obtain boundary conditions for $\mathbf{b}$

$$\mathbf{b}_m^k = 0 \tag{5.42}$$

where again the index m corresponds to nodes with large distance from the node 1. Considering the conditions (5.40) and (5.42), we can interpret the evolution of the vector $\mathbf{b}^k$ as diffusion of probability with fixed source at the departing node 1. This interpretation will allow us to construct connection between the mean first passage time and the probability of return to the origin.

After several applications of matrix TM, the vector $\mathbf{v}^k$ will converge to the eigenvector $\tilde{\mathbf{v}}$ of matrix TM that corresponds to the largest eigenvalue $\lambda_0 < 1$. We again define the corresponding steady state vector $\tilde{\mathbf{b}}$ as

$$\frac{1}{n}\tilde{\mathbf{b}}_i = \frac{d_i}{\langle d \rangle n} - \tilde{\mathbf{v}}_i. \tag{5.43}$$

The flow p of probability to the absorbing node is then the inverse of the mean first passage time. We will express this quantity in terms of vector $\tilde{\mathbf{b}}$:

$$p = \sum_{<i1>} \frac{\tilde{\mathbf{v}}_i}{d_i} = \frac{1}{n} \sum_{<i1>} \frac{1}{d_i} \left(\frac{d_i}{\langle d \rangle} - \tilde{\mathbf{b}}_i \right) = \frac{1}{n} \sum_{<i1>} \left(\frac{1}{\langle d \rangle} - \frac{\mathbf{c}_\mathbf{i}}{d_i \langle d \rangle} \right) \tag{5.44}$$

where the sum is taken over all neighbors of absorbing node. We also introduced a new vector $\mathbf{c}$ which corresponds to $\tilde{\mathbf{b}}$ up to a scaling factor

$$\mathbf{c} = \langle d \rangle \tilde{\mathbf{b}}. \tag{5.45}$$

The equation (5.44) then becomes

$$p = \frac{1}{n\langle d \rangle} \sum_{<i1>} \left(1 - \frac{\mathbf{c}_i}{d_i} \right) = \frac{1}{n\langle d \rangle} \sum_{<i1>} (1 - r_i). \tag{5.46}$$

The quantity r_i is equal to $\frac{\mathbf{c}}{d_i}$, which is the probability flow back to the origin of diffusion, i.e., the probability of coming back to the departing node from the adjacent node i. Vector $\mathbf{c}$ satisfies the following condition at the origin of diffusion

$$\mathbf{c}_0 = \langle d \rangle \tilde{\mathbf{b}}_0 = d_0 \tag{5.47}$$

which means that the probability flow to each of the neighbor nodes is equal to one. In order to calculate the return probability r_i in (5.46), we will consider a model of an Erdös-Rényi graph where the departing node that has only one edge from which the flow of probability into the rest of the graph is equal to 1. We will calculate numerically the distribution of the return probability r for such graph. In contrast to the fixed connectivity case, r is a random variable. In the case when the departing node is connected to a d-regular graph, the return probability satisfies the following relation:

$$r = \frac{1}{d}\left(1 + \frac{d-1}{d}r + \left(\frac{d-1}{d}r\right)^2 + \dots\right) = \frac{1}{d\left(1 - \frac{(d-1)r}{d}\right)} \tag{5.48}$$

where $\frac{1}{d}$ is the probability of coming back to the departing node, while $\frac{d-1}{d}$ is the probability of choosing different neighbor and making a step further from the original node. The k-th term in the geometric series (5.48) then corresponds to the probability of visiting the node adjacent to the departing node k times before coming back to the origin.

However, we are interested in the case where the degree of each node in the graph follows a Poisson distribution $p(d)$ with mean $\langle d \rangle$. For the numerical calculation of the distribution of return probability, we construct a histogram of distribution of return probabilities. This histogram is then iterated using the formula:

$$r = \left\langle \frac{1}{d} \frac{1}{1 - \frac{\sum_{j=1}^{d-1} r_j}{d}} \right\rangle_d = \sum_{d=1}^{d_{\max}} p(d) \left(\frac{1}{d} \frac{1}{1 - \frac{\sum_{j=1}^{d-1} r_j}{d}} \right) \tag{5.49}$$

which is an implicit formula for the distribution of r. The term $\sum_{j=1}^{d-1} r_j$ is constructed in our numerical calculation by composing the histograms of distribution of r. Note that our calculation assumes an infinite graph.

The numerical solution for a distribution of return probability using formula 5.49 is shown in figure 5.8.

To obtain the mean first passage time, we calculate the mean value of quantity $\frac{1}{p}$ from (5.46) using the obtained probability distribution for r. In the case of a random graph with absorbing node having only one neighbor (which translates into the departing node having only one neighbor for the return probability approach) we have

$$\langle H_1 \rangle = \langle d \rangle n \left\langle \frac{1}{1-r} \right\rangle_r . \tag{5.50}$$

For example, considering the case of $\langle d \rangle = 5$, the value calculated using the distribution shown in figure 5.8 is

$$\langle H_1 \rangle = 6.72n \tag{5.51}$$

The comparison with a numerical simulation of absorption is shown in figure 5.9.

To obtain a solution for for mean first passage time on a node whose degree is given by a Poisson distribution (i.e. the general case for calculation of mean first

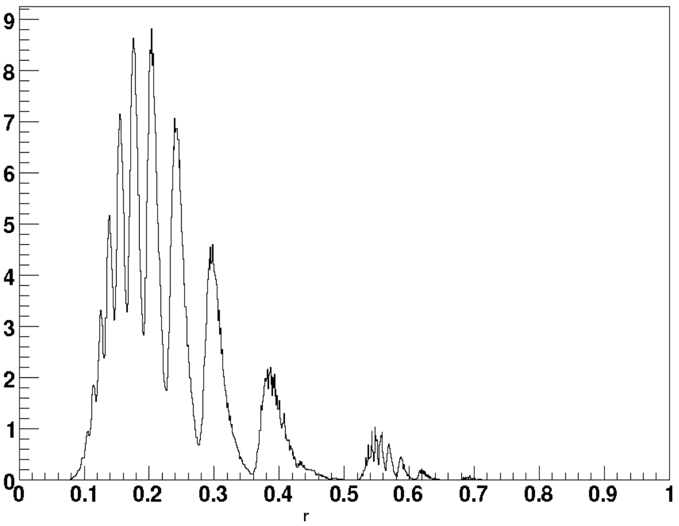

Figure 5.8: The probability distribution of return probability r on a random graph with Poisson distribution of a degree of a node with mean degree equal to 5. The distribution was obtained by iterating the equation (5.49) until a stable solution was obtained. The value of $d_{\max}$ in (5.49) was set to 16 for this calculation. The program used to produce the distribution is available on the enclosed CD-ROM and its use is described in A.2.4.

passage time on an Erdös-Rényi graph), we average the quantity $\langle H_1 \rangle$ for different number of neighbors of the absorbing node

$$\langle H \rangle = \sum_{i=1}^{d_{\max}} \frac{\langle H_1 \rangle}{d_i} P(d_i) \tag{5.52}$$

which gives for the Poisson distribution with mean degree 5

$$H = 1.73n \tag{5.53}$$

The comparison with the simulation is shown in figure 5.10.

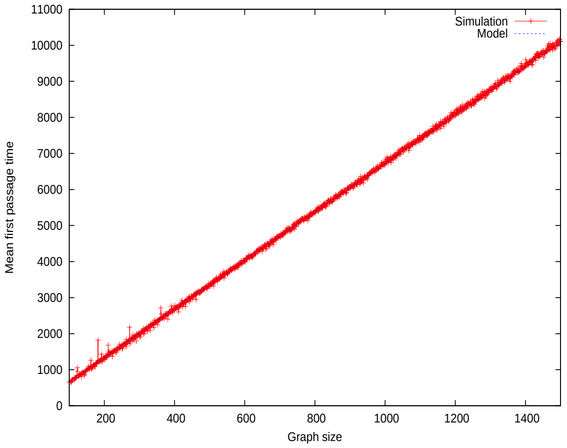

Figure 5.9: The comparison of the numerical simulation with calculation of mean first passage time for Erdös-Rényi graph with absorbing node having only one neighbor (equation (5.51)). The mean degree in the rest of the graph is 5.

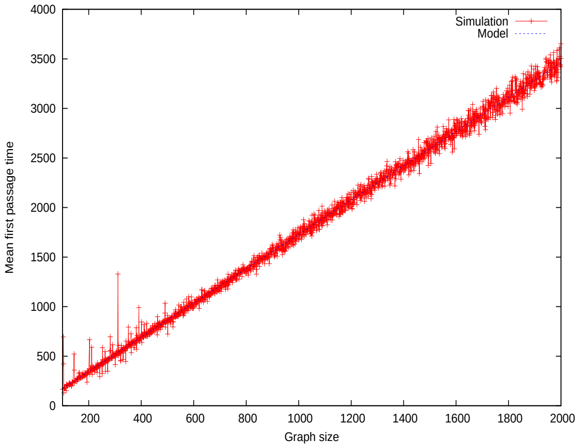

Figure 5.10: The comparison of calculation of mean first passage time for Erdös-Rényi graph with mean degree equal to 5 (equation (5.53)) and the numerical simulation.

Chapter 6

Discussion and conclusions

Mappings from genotypes to phenotypes play a central role in biology, from the molecular scale up to whole organisms. Working at the level of RNA allowed us to use a framework for such mappings that is accepted as “relevant” in the biological community while being at the same time relatively tractable from the computational point of view. We showed a rich phenomenology of the associated evolutionary dynamics: not only does the “relaxation” towards the optimum undergo severe slowing down, but also it seems possible that this slowing down remains sensitive to the quenched disorder in the thermodynamic limit (long chain sizes). Thus there are both analogies and differences with what happens in physical systems undergoing ageing.

What causes this slowing down? We saw that the fraction of favorable mutations decreases severely as one approaches the optimum, a property of the fitness landscape itself. Nevertheless, the major feature of the slow-down is the long waiting times encountered by random walks on neutral sets, far larger than expected from the frequency of favorable mutations. These neutral sets are strongly heterogeneous, leading to walks that spend much time in unfavorable regions. Understanding where these heterogeneities come from would be of major interest, just as in the case of neutral networks where the same problem remains open today in spite of the many groups investigating RNA neutral networks. Such heterogeneities seem to be general features of complex landscapes.

The process of random walk through neutral sets has lead us to the study of random walks on graphs, particularly to estimate the first passage time. We studied several approaches to estimate the mean value of first passage time on random graphs and supported our calculations by numerical simulations.

This work opened up for me the frontier between physics, biology and computer science. Beyond the large amount of biological content I had to digest, I learned about diverse mathematical and physical methods (random walks, Markov processes, trap models, graph theory, random matrices ...). I also got a view of research in an interdisciplinary field where quantitative tools are sure to be in high demand in the foreseeable future.

Naturally, for this project, I had to develop many computational codes, shell scripts for managing the long runs and to perform the analysis. Note that these studies are particularly intensive from the computational point of view. For a molecule of length L, several times L mutations need to be applied to get to an independent genotype. Each mutational step is followed by the folding of the molecule which takes $O(L^3)$ operations. Thus the CPU time is dominated by these folding operations, implemented in already optimized software, thus no real speed-ups are possible. The C++ codes performing the evolution simulation are available on the enclosed CD-ROM.

Appendix A

The simulation code

A.1 The simulation code for RNA evolution

The source code was written in C++ programming language. The source code together with the results of our simulations is available on the enclosed CD-ROM. In this section, we describe the structure of the code and explain how to the data were produced. The code for RNA chains requires standard C++ library and ViennaRNA package [3] and ANSI C++ standard compatible compiler. The simulation code for random graphs requires standard C++ library. The code is written using object-oriented programming techniques. All source code concerning RNA evolution simulation is in directory `sourcecode/RNA`.

A.1.1 RNA manipulation and simulation interface

The program for simulation of RNA chains consists of the following files:

1. `rnass.h` and `rnass.cpp` contain an interface for manipulation of RNA sequences and structures. It contains two classes, `RNASequence` and `RNAStructure` that provide interface for manipulation with RNA chains and sequences in an object-oriented program. `RNASequence` class encapsulates manipulation with sequence which is represented as a string in the class. Class RNAStructure encapsulates manipulation with secondary structure which is represented as a string of brackets and dots.

2. `randomwalk.h` and `randomwalk.cpp` contain the simulation algorithm itself. They contain declaration and implementation of random walk towards a given target structure. The algorithm is encapsulated in class `RandomWalk`, which comprises functions for saving the information about the evolution process. The method `RandomWalk::Walk` starts the evolution simulation of random walk from a given sequence to target secondary structure. It ends if the

target structure is reached or the number of maximal allowed mutations is reached, whatever comes first. Method `RandomWalk::SaveWalk` saves information about the evolution process to file that can be later evaluated to obtain statistics about the process. The information saved to file include time spent on a given neutral set before finding a new beneficial mutation, number of different structures encountered during the diffusion in neutral sets and Hamming distance between the first and last RNA sequence encountered in the set. The functions that are used to evaluate these data are declared in the file `simulation.h`.

3. `simulation.h` and `simulation.cpp` contain declaration and implementation of class `Simulation` which provides interface that either generates randomly or loads from a file a departing chain and than starts the random simulation by calling a method `RandomWalk::Walk` implemented in file `randomwalk.cpp`. Method `Simulation::StartRandomSimulation` generates different target phenotypes and runs a simulation from a random departing sequence given number of times. File also contains method `Simulation::Statistics` that performs the statistical evaluation of the files produced during the evolution simulation.

A.1.2 Programs

Two main programs using the interface described in A.1.1 were used throughout the simulation. Program `walk` produces simulation data while program `analyze` reads the output files from the simulation and produces files that can be later processed by data-analysis software such as Matlab or Gnuplot. Both programs use methods from class `Simulation`.

Program `walk`

In order to compile the program, type `make walk` in the `sourcecode/RNA` (you need to have make utility installed in your system. Note that ViennaRNA package has to be installed on the system as well.). This command compiles file `walk.cpp` and produces a binary file `walk`. The usage of the program is as follows:

```
walk -o data.out -m 1000 -t 20 -s 10 -n 20 -l 100 -r 7
```

The program generates randomly several secondary structures (as specified by parameter `-n`) and then randomly chooses the departing and target structure. It then performs random mutations until the target structure is reached or until the number of mutations reaches value specified by `-m` parameter. The whole random walk for the same pair of structures is repeated as many times as specified by `-s` parameter. The whole process is repeated `-t` times. The parameter `-r` serves to initialize the random number generator. The length of the RNA molecules in the simulation is specified by the `-l` parameter. Note that the time needed for the simulation increases as $O(l^3)$ where l is the length of the molecule. Finally, the `-o` parameter specifies the name of the file where the simulation output will be saved.

There are several other variants of the simulation program `walk` and whose usage is very similar. Program `uniquetargetwalk.cpp` performs only evolution towards a target structure that is saved in a file (in a bracket-dot notation) specified by the `-i` parameter. Programs `evenuniquetargetwalk.cpp` and `evenwalk.cpp` perform the same functions as programs `uniquetargetwalk` and `walk.cpp`, but allows only beneficial mutations that bring the phenotype closer to target structure by the distance equal to two.

Program `analyze`

This program analyzes data produced by program `walk`. To compile the program `analyze.cpp`, type `make analyze` in the directory with source files. The usage is as follows:

```
analyze -i data.out -m 1000 -o graphs.dat
```

The program takes as an input (specified by the `-i` parameter) file produced by `walk`. The maximal number of mutations with which the program `walk` was executed must be specified as well. The output is saved to file specified by `-o` parameter. The data are saved into columns, the first column is the number of mutations, second column is average Hamming distance to target phenotype (divided by the length of the RNA molecule) after the given number of mutations, averaged over all simulation data. The third column is the standard deviation. Note that the input file can be in fact a result of several independent simulations merged together. In that case, all of them have to be executed with the same parameters except for the random number generator, which has to be different.

Program `fullanalyze`

Program `fullanalyze` (source code `fullanalyze.cpp`) extracts information from simulation as a function of distance from the target structure. It has one additional parameter `-t`, compared to program `analyze`, that specifies the distance to the target. For each distance from the target, the program saves into the output file the following information: distance from the target structure, time spent on a neutral sent before finding a beneficial mutation, number of different phenotypes encountered in neutral set, number of neutral mutations performed before finding a beneficial mutation, drift distance within the neutral set, length of jump (how much closer is the new neutral set compared to the original one). To compile the program, type `make fullanalyze`.

Other programs

Additional programs are included in the source code directory. Their compilation and use is the same as for the programs described above. Typically, there is always one program that performs a desired simulation and another program that creates statistics from the produced data. Program `neutral_walk.cpp` has one additionally parameter `-d` compared to `uniquetargetwalk.cpp` that specifies the distance from target structure at which the simulation of evolution stops and then only neutral mutations are accepted. It then accepts only neutral mutation in the same neutral set. The file produced is then analyzed by program `analyze_neutral.cpp`. The program `sampling.cpp` works similarly as `neutral_walk.cpp`, but saves data only after a given number of steps in the neutral network (specified by `-t` parameter). The data are analyzed by `analyze_sampling.cpp` program. The rest of the source code files in the directory contain functions used by the described programs. Please note that all programs print on the screen an example of usage of they are executed without any parameters. The format of files produced by program for analyzing data from the simulation are described in the `README.txt` file in the `sourcecode/RNA` directory.

A.1.3 Data

The simulation results are located in the `data` directory. The simulations for different RNA molecule lengths are located in `data/AveragedWalks`. The subdirectories's names correspond to the length of the RNA molecule. The files in the directory with file extension `.dat` are produced by the `walk` program, while files with `.txt` extension are processed by program `analyze` into a form displayable in Gnuplot. Simulation results discussed in section 4.8 are in the directory `data/Heterogenity_CHAIN40`. The data that contain properties of neutral networks described in section 5.2 are saved in the directory `data/NeutralNetworkSampling`.

A.2 The simulation code for graphs

The source code for performing simulations for random graphs dynamics was written in C++. It requires only standard C++ library and a compiler compatible with ANSI C++ standard. They are located in `sourcecode/graphs` directory.

A.2.1 Main files

1. `mygraph.cpp` contains a class `graph` that provides interface for creating and managing random graph. Mainly, it randomly generates a random graph with a given fixed degree and number of nodes given by a parameter.

2. `fixedgraph.cpp` Implements absorbing walk and iteration methods as described in sections 5.6.1 and 5.6.2.

A.2.2 Programs

Program `absorption.cpp` is used to launch absorbing walk algorithm from section 5.6.1. It is launched with parameters specifying the number of nodes and degree of a node.
Program `iteration.cpp` is used to launch iteration method algorithm described in section 5.6.2. It runs the algorithm for graphs of sizes that are specified by the parameters when executing the program. Both programs give an example of usage if launched without any parameters.
Finally, `matrice.cpp` generates a random graph and computes numerically its eigenvalues. This program requires *newmat* library for matrix manipulation which can be downloaded from `http://www.robertnz.net/nm_intro.htm`.

A.2.3 Data

The data from random walk simulation with an absorbing node are located in the directory `graphdata`. The degree of a regular graph for which the simulation was run is specified in the name of the file after the letter `Z`. The files were produced by launching program `iteration` for various graph sizes. The data in the first column in the file is the mean first passage time computed for a given graph size and the second column corresponds to the value calculated by formula (5.22).

A.2.4 Return probability calculation

The program `iterHist12.C` used to produce the figure 5.8 is located in the directory `graphdata/MATHvsSIM`. The program requires the libraries from the ROOT program which is available for free at `root.cern.ch`. The histogram for the distribution of return probabilities as shown in figure 5.8 is then produced by calling function `IterHist` which takes as parameters the maximal degree considered ($d_{\max}$ in equation (5.49)), mean degree of a node, number of iterations and the number of bins in the histogram.

Appendix B

Glossary

DNA stands for deoxyribonucleic acid. DNA is a two-stranded molecule containing nucleotides adenine (A), cytosine (C), guanine (G) and thymine (T). The two strands form a double-helix structure, where adenine pairs with thymine and guanine pairs with cytosine. The DNA molecule encodes genetic information in the cell.

Enzyme is a biomolecule that is used to catalyze chemical reactions

Fitness is a number that is assigned to genotypes during evolutionary simulations. How the number is assigned depends on the simulation requirements. Usually only the genotypes with the best fitness are taken into account in the next iteration of the evolution algorithm.

Genotype refers to a genetic constitution of an organism, i.e. its unique genetic information stored in DNA.

Nucleotide Nucleotides are organic molecules that are basic structural units of RNA and DNA.

Phenotype is an observable expression of a genotype. For example, a phenotype might be the color of eyes of an individual. Different genotypes can lead to the same phenotype, for example two individuals might have different genetic code that is responsible for eye color, but still both have blue eyes. The relation between genotype and phenotype is usually quite complicated and often phenotype is a result of interaction of many genes.

Polymer is a large molecule composed of structural units connected together by a chemical bond

Primary structure specifies the composition of a molecule. In case of an RNA strand, the primary structure is a list of *nucleotides* that appear in the chain.

Proteins are polymers of amino acids responsible for many functions in living organisms. Proteins are composed of aminoacids. They are basic constituents of cells, but some of them take part in chemical reactions as enzymes or have signalling function in the organism. Proteins are produced in ribosomes in cells, where the information that serves as a "cookbook" for the protein production is transported by RNA molecules.

Proteome refers to a set of proteins found in a particular cell or organism.

RNA is an abbreviation for *ribonucleic acid.* It is a chain molecule that contains *nucleotide units* adenine (A), cytosine (C), guanine (G) or uracil (U). RNA molecules are usually single stranded. RNA is responsible for transport of information from DNA to ribosomes for protein synthesis. In addition, certain RNA molecules play the role of enzymes. It is believed that RNA molecule served also as a carrier of genetic information in early life forms before DNA appeared.

Secondary structure of a protein is the local ordering in space of the amino acid chain. In the case of an RNA chain, secondary structure simply refers to whether a given nucleotide base is paired with other nucleotide base or not.

Tertiary structure refers to a full three-dimensional description of a molecule. It specifies the coordinates of each atom of the molecule

Neutral mutation refers to a change in genotype that does not change phenotype.

Neutral network In biology, the term neutral network refers to a graph whose nodes are genotypes that correspond to the same phenotype.

Neutral set is an ensemble of neutral networks with some common property (for example all of them have the same fitness).

Nucleotides are biomolecules that are structural units of DNA and RNA.

Bibliography

[1] C. Monthus, J.-P. Bouchaud, *Models of traps and glass phenomenology*, J. Phys. A: Gen., Vol. 29, 3847-3869, 1996

[2] F. Dardel, F. Képès, *Bioinformatique: Génomique et post-génomique*, Éditions de l'École Polytechnique, Palaiseau, 2006

[3] `http://www.tbi.univie.ac.at/∼ivo/RNA/`

[4] M. Zuker, P. Stiegler, *Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information*, Nucleic Acids Res., Jan 10;9(1):133-48, 1981

[5] Reinhard Diestel, *Graph Theory*, Third Edition, Springer-Verlag, Heidelberg, 2005

[6] E. Van Nimwegen, J. P. Crutchfield, M. Huyen, *Neutral evolution of mutational robustness*, Proc. Natl. Acad. Sci. USA, Vol. 96, 9716-9720, 1999

[7] A. Wagner, *Robustness and evolvability: a paradox resolved*, Proc. Biol. Sci., Vol 275, 91-100, 2008

[8] M. Kirschner, J. Gerhart, Proc. Natl. Acad. Sci. USA, *Evolvability*, Vol. 95, 8420-8427, 1998

[9] L. A. Meyers, F. D. Ancel, M. Lachmann, *Evolution of Genetic Potential*, PLoS Computational biology, Vol. 1, 2005

[10] M. C. Cowperthwaite, L. A. Meyers, *How Mutational Networks Shape Evolution: Lessons from RNA Models*, Annu. Rev. Ecol. Evol. Syst. 38, 2007

[11] J. R. Norris, *Markov Chains*, Cambridge University Press, Cambridge 1997

[12] E. Seneta, *Non-negative matrices and Markov chains*, Springer-Verlag, New York, 1981

[13] `http://www.gnuplot.info`

[14] B. Rinn, P. Maass, J.-P. Bouchaud, *Multiple scaling regimes in simple aging models*, arXiv preprint: cond-mat/001161v1

[15] M. Kimura, T. Ohta, *On the stochastic model for estimation of mutational distance between homologous proteins*, J. Mol. Evol., 87-90, 1972

[16] W. Fontana, P. Schuster, *Shaping Space: the Possible and the Attainable in RNA Genotype-phenotype Mapping*, Science, Vol. 280. no. 5368, pp. 1451 -1455 (1998)

[17] A. Barrat: *Habilitation à diriger des recherches: Milieux granulaires, gaz granulaires: des systèmes modèles hors de l'équilibre; Eléments d'étude des réseaux complexes* , Université de Paris XI - U.F.R. des sciences d'Orsay, 2005

[18] N. C. Wormald, *Models of Random Regular Graphs*, London Mathematical Society Lecture Note Series, Vol. 267, 239-298, 1999

[19] L. Lovász, *Random Walks on Graphs: A Survey*, Combinatorics, Paul Erdös is Eighty (Vol. 2), 1993

[20] O. Bohigas, *Random Matrix Theories and Chaotic Dynamics*, Elsevier Science Publishers, 1991

[21] F. Chung, L. Lu, V. Vu, *Spectra of random graphs with given expected degrees*, Proc. Natl. Acad. Sci. USA, Vol. 100, 6313-6318, 2003

[22] O. C. Martin, R. Monasson, R. Zecchina, *Statistical mechanics methods and phase transitions in optimization problems*, Theor. Comp. Sci., Vol. 265, 3-67, 2001

[23] B. Alberts, D. Bray, A. Johnson, J. Lewis, M. Raff, K. Roberts, P. Walter: *Essential Cell Biology*, Garland Publishing, New York, 1997