Combining genetic regulatory network models and a random matrix approach to solve large biological networks

Cas Jiskoot

4564006

Bachelor End Project, TU Delft
BSc Applied Mathematics / BSc Applied Physics
Supervisors: J.L.A. Dubbeldam, J.M. Thijssen
Assessed by N. Parolya and T. Idema, to be defended on 24-04-2020

Abstract

In this thesis we study genetic regulatory networks using a minimal nonlinear model from literature and extend this to multi-compartment gene-to-gene interaction networks using both numerical simulations and random matrix theory. When we add perturbations to a system of two interaction networks, the largest eigenvalue becomes larger, which means the system is more likely to be unstable. It turns out that for a certain region of values of random matrix variables, we can predict the change of their eigenvalues caused by a perturbation with perturbation theory. We quantify the stability of the genetic regulatory network models, where the dynamics is largely governed by random matrices, with maximal Lyapunov exponents (MLE's). We have been able to compute these MLE's when the model is considered with certain constraints. At last, we research the correlation between the connectivity of two networks describing such models and their KL divergence. Here we conclude that the KL divergence between two networks goes up when the connectivity of one of the networks gets larger. We conclude that for possible follow-up research more advanced programs are needed with longer running simulations.

Contents

1 Introduction

A lot of living systems appear to be in a state which is called 'the edge of chaos'. It is called this way because the systems are in a stable state, but only a slight change in the system can put it into chaos. It is important for living systems to be in such a state because they need to be able to adapt to changes in an environment, so complete stability will not suffice. But their internal state should not change too much when small changes occur in their surroundings, so the system should not be in a critically chaotic regime. From these two remarks it follows that living systems should be in a marginally stable regime, meaning they are not asymptotically stable, but also not unstable. In this report genetic regulatory networks, which can be generalized to more general living systems, are considered. A genetic regulatory network is an interacting collection of molecular regulators and together with molecular species in the cell it governs the gene expression levels of mRNA and certain proteins. They will be described mathematically using complex networks, which are large networks that display a substantial amount of non-trivial topological features.

For a long time it was widely thought that the more such complex networks were connected, the more stable the system they describe would be. It seems like a logical assumption that when there are more edges combining the nodes in a large network that it will be easier for the network to adapt to certain changes without going into a state of chaos. However, in the 70's of the twentieth century Robert May used random matrix theory (RMT) to prove that the exact opposite is true [1]. With help of existing theorems regarding matrix eigenvalue distributions in the complex plane, he concluded that when a complex network has more connections between the nodes, the system is more likely to be in a state of chaos and will be less stable then when it has fewer connections. This RMT approach has been used extensively since then for identifying certain factors in large and complex biological ecosystems that strive to be in a stable environment. The models used for this are also used in a lot of other disciplines such as neurosciences, atomic physics and the economy.

In this thesis we take a look at coupled genetic regulatory networks and study how the dynamics for such larger systems adapt to perturbations. Also, we try to compare several of these systems using concepts from information theory. To achieve this we start by introducing some basic properties of network theory in order to define random Erdös-Renyi networks. In particular, we link the eigenvalues of the matrix defined by a network to the stability of a system which is described by this network. Then, we look at the consequence of a perturbation to the system and we approximate the difference of the eigenvalues, as a result of this perturbation, with perturbation theory from quantum mechanics.

Furthermore, the predicted eigenvalue distributions from RMT are applied to the random Erdös-Renyi networks that are considered in the genetic regulatory network. Certain variables will be examined to see for which values the genetic regulatory model will be stable. Using a random network a matrix can be generated from which we can extract important information about the stability of a system. We quantify this stability with so-called maximal Lyapunov exponents (MLE's). When the MLE of a system is negative, the system is asymptotically stable. For symmetric systems this MLE will only be zero at certain fixed points of measure zero. However, this MLE-value of zero is necessary for a system describing a living organism, which is why we need a much larger region than only measure zero points. When the MLE of a system is equal to zero, the system is in a marginally stable state, or the earlier mentioned 'edge of chaos'-state. Which means it is not asymptotically stable, but also not critically unstable. It turns out that with the natural contraint that the concentration of a chemical substance can not be negative, the symmetry of the system is broken and there will be a larger region where the MLE's become close to zero so there arises a larger part in which living systems will be able to exist [2].

At last, another model is being studied where the criticality arising in evolving systems is also a main subject. Instead of looking at different gene-to-gene interactions in a single organism, here the interaction between individual organisms is studied. Each of those individual organisms adjusts in such a way that it can live in a changing environment as efficient as possible. The main parameter to check if an individuals fitness compared to its environment is optimal is called the Kullback-Leibler divergence (KL divergence). By minimizing this KL divergence the internal state of an individual comes closer to the external state of the environment which means it will be easier to adapt to changes in this environment. From simulations with these models it appears that also in this case the fitness of an individual will be the greatest in a state of certain criticality just as with the MLE's [3]. In the last part of this thesis we combine these two models in such a way that the gene expression levels of mRNA and proteins from the first model will be adjusted such that we can calculate their relative KL divergences to each other. Then the hypothesis is that they will converge towards a state in which their MLE is in the same optimal regime as in the genetic regulatory model.

2 Network theory

In the modern world we are surrounded by a lot of structures and systems that can be mathematically described using complex networks. Some examples are the complete society of the world in which billions of people live together, the billions of neurons in our brain that together need to make decisions for one individual, or communication infrastructure consisting of millions of computers and cellphones. The goal of network science is to create models which describe all these real systems as realistically as possible. Most of the real networks do not look well arranged at all and they have a lot of random factors involved in their creation. To cope with this problem random network theory is used, which consists of networks that are completely random-generated. One example of a random network is an Erdös-Renyi network which will be further examined in the next part.

2.1 Erdös-Renyi Random networks

To construct a random network, a few variables are needed. First the number of nodes is specified, this is denoted by the letter N. Also a probability p is defined. Then for every two nodes a random number between 0 and 1 is generated. If the number exceeds p, a link is placed between the two nodes. A random network constructed this way is called an Erdös-Renyi network. Every time such a network is generated, it will have a specific number of links L. The probability that a realization of a network will have L number of links is equal to [4]:

$$p^L = \binom{\frac{N(N-1)}{2}}{L} p^L (1-p)^{\frac{N(N-1)}{2}-L}, \tag{1}$$

and the expectation value of L is

$$\langle L \rangle = \sum_{L=0}^{\frac{N(N-1)}{2}} L p_L = p\frac{N(N-1)}{2}. \tag{2}$$

Because of this it can be seen that the average degree of a random network is

$$\langle k \rangle = \frac{2\langle L \rangle}{N} = p(N-1). \tag{3}$$

Because random networks are discussed, there usually are large differences between the number of links the nodes have. A quantity to describe this is the degree distribution p_k, which gives the probability that an arbitrary node has degree k. With a random network this degree distribution has a binomial form, but when the number of links N is much larger than $\langle k \rangle$ it can be approximated very well by the Poisson distribution

$$p_k = e^{-\langle k \rangle} \frac{\langle k \rangle^k}{k!}, \tag{4}$$

which is preferred over the binomial distribution because it only has one parameter $\langle k \rangle$ instead of two (N and p).

2.2 Random matrix theory

When a random network is constructed, the graph of this network can be represented by an adjacency matrix. For an N-species community, the amount of the i'th species at time t is given by $N_i(t) = N_i^* + x_i(t)$, where N_i^* is an equilibrium value and $x_i(t)$ is a perturbation. The system can then be described using the adjacency matrix in the form of a differential equation which has been linearized around an equilibrium state:

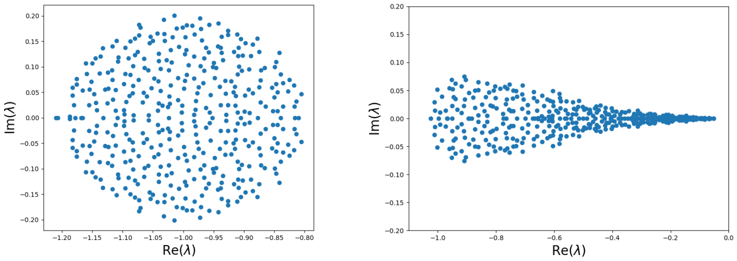
$$\dot{\mathbf{x}} = A\mathbf{x}. \tag{5}$$

Here the $\mathbf{x}$-vector contains all the perturbations from the equilibrium state for the different i's. The dynamics of the system are governed by the adjacency matrix A. The elements of this matrix show how each species effects all the other species, a cooperative effect means that $a_{ij} > 0$, and a negative effect yields $a_{ij} < 0$. First the diagonal elements a_{ii} of this A-matrix, so the self-interactions of the species, are all scaled at -1. The other elements of the matrix are taken from a normal distribution with mean zero and variance σ^2. This matrix can thus be viewed as a sum of the identity matrix and a matrix with random elements: A = -I + B. Now in reference [1] a connectance C is introduced to model the interaction in the network. A fraction (1-C) of the matrix elements a_{ij}, chosen at random, is set to zero. We are looking for local stability in this model, which means the system will return to an equilibrium state after a small perturbation. This equilibrium state in terms of $\mathbf{x}$ is the state where $A\mathbf{x} = 0$, because the system is in equilibrium when there is no perturbation. From [1] it follows that the system is locally stable if

$$\gamma = \sigma\sqrt{NC} < 1. \tag{6}$$

This means that when either the amount of species or individuals, the variance of the normal distribution, or the connectance between the nodes becomes too large, the system will move to an unstable state. With this, May [1] showed that large networks with more connections between nodes are indeed more likely to be unstable than networks with fewer connections.

The eigenvalues of the matrix A are distributed uniformly in a circle in the complex plane which has radius γ. This circle is centered around (-1,0) because A has -1 as diagonal values. Now when γ is increased until it is larger than 1, there will arise eigenvalues with real parts larger than zero, which is a condition for instability [1]. We now consider a new matrix $S = DA$, where $D = diag(N_i^*)$ is a diagonal matrix with positive elements on the diagonal. The matrix S is called the community matrix because this matrix takes all the individual species into account and D can be seen as a population density matrix. So when we want to see if the complete system is stable we need to consider the community matrix S. The eigenvalue distribution of S will lose its circular shape which is shown in figure 1:



(a) Adjacency matrix A

(b) Community matrix S

Figure 1: Eigenvalue distributions of the two matrices A and S. In both cases $\gamma = 0.02$: $N = 400$, $\sigma = 0.01$, $C = 1$.

The eigenvalue distribution of S is obtained in the following way. For large values of N, the eigenvalue density of S will be nonzero for $z \in \mathbb{C}$ satisfying [5]:

$$tr[(D_z D_z^*)^{-1}] \geq \frac{1}{\sigma^2}, \tag{7}$$

with $D_z = D^{-1}(zI - D)$. Applying this inequality to the community matrix S, the following region in the complex plane is found for the eigenvalues of S, [1]:

$$\sum_{i=1}^{N} \frac{(N_i^*)^2}{(z + N_i^*)(\overline{z} + N_i^*)} \geq \frac{1}{C\sigma^2}. \tag{8}$$

So there will only be eigenvalues of S in this region of z-values. The plotted eigenvalues in figure 1 correspond to the values of z in this equation. Also, unlike with the eigenvalue distribution of A, the eigenvalues of S are not uniformly distributed along the region.

2.3 Larger concatenated matrix

In this section we will consider a larger matrix, constructed by two matrices from the size of A. If, for example, we take two different N-species communities, we get two different matrices to use in equation 5. We will call these two matrices matrix A and B respectively and they are put together in the following way:

$$\begin{pmatrix} A & \phi \\ \phi & B \end{pmatrix}, \tag{9}$$

where ϕ stands for a matrix with the same size as A and B consisting of only zeros. In the next step we will add some elements to these empty matrices. We call this new matrix Q and we add this matrix together with its transpose Q^T to the system as follows:

$$\begin{pmatrix} A & Q \\ Q^T & B \end{pmatrix}. \tag{10}$$

Like this we get coupled pairs of elements from Q and Q^T which perturb the system. We expect that this perturbation will cause a change in the dynamics of the system, specifically in the eigenvalue distribution of the new system. In the next chapter we will try to get a quantitative measure for this change by applying perturbation theory from quantum mechanics.

3 Perturbation theory

Perturbation theory is a widely used theory in quantum mechanics when a slightly perturbed system is not solvable analytically. It turns out that we can quite easily adapt this theory from quantum mechanics to use it in random matrix theory. In this chapter the background for the theory in quantum mechanics is shortly treated, after which we will treat the equations we need for the problem with random matrices.

3.1 Quantum mechanics

In quantum mechanics, we search for solutions of the wave function ψ, given a certain Hamiltonian H^0. This is done by solving the Schrödinger equation. Suppose we have such a solution for the time independent Schrödinger equation:

$$H^0\psi_n^0 = E_n^0\psi_n^0. \tag{11}$$

This gives us a complete set of orthonormal eigenfunctions ψ_i^0 with the following condition, [6]:

$$\langle\psi_n^0|\psi_m^0\rangle = \delta_{nm}. \tag{12}$$

When the Hamiltonian H^0 is perturbed slightly by $H = H^0 + \lambda H'$, we need to find new eigenfunctions and eigenvalues according to:

$$H\psi_n = E_n\psi_n \tag{13}$$

Because solving this problem in quantum mechanics does not belong to the scope of this project, we will just give the results for the first and second order corrections from reference [6], without derivations. After this we use these results on our problem from random matrix theory. The first order correction for the energy is

$$E_n^1 = \langle\psi_n^0|H'|\psi_n^0\rangle, \tag{14}$$

which is the expectation value of the perturbation, in the unperturbed state. The second order correction is equal to:

$$E_n^2 = \sum_{m\neq n}\frac{|\langle\psi_m^0|H'|\psi_n^0\rangle|^2}{E_n^0 - E_m^0}. \tag{15}$$

It turns out that for our goal, determining the new eigenvalues for a random matrix after a perturbation, we do not need to go any further than this when we consider the non-degenerate case.

3.2 Perturbed matrix

3.2.1 Analogy between quantum mechanics and matrices

We can easily make the comparison between the quantum mechanics case and the random matrix theory for perturbation theory. Just as with quantum mechanics, we are looking for eigenvalues and eigenvectors, and we can quantify the result of the perturbation in the same way with equations 14 and 15. Because we are only interested in the changing eigenvalues of the system, since they are a measure for the stability of our system, we did not look at the corrections for the eigenfunctions in the last section. To use the equations from perturbation theory we will now define some matrices. First, we call the matrices from equations 9 and 10 matrix C and D respectively, so

$$C = \begin{pmatrix} A & \phi \\ \phi & B \end{pmatrix}, D = \begin{pmatrix} A & Q \\ Q^T & B \end{pmatrix}. \tag{16}$$

Also, we need to define the matrix with only the perturbation and no other terms, we will call this perturbation matrix P:

$$P = \begin{pmatrix} \phi & Q \\ Q^T & \phi \end{pmatrix}. \tag{17}$$

When we multiply A by its eigenvectors, the eigenvalues are denoted as follows:

$$A \left|\psi_n^A\right\rangle = \lambda_n^A \left|\psi_n^A\right\rangle. \tag{18}$$

These eigenvalues and eigenvectors are the same as the first half of the eigenvalues and eigenvectors of matrix C, supplemented by zeroes making them twice as long. We call the Hilbert space of eigenvalues λ_n^A and eigenvectors ψ_n^A the space $\mathscr{H}_A$, and a similar Hilbert space $\mathscr{H}_B$ for B. Now, when we let perturbation matrix P work on the set of eigenvectors ψ_n^A of matrix A, we get a vector in the Hilbert space $\mathscr{H}_B$:

$$P \left|\psi_n^A\right\rangle \in \mathscr{H}_B. \tag{19}$$

This new vector thus consists of N zeros, and then has an number of N elements below these zeros. If we want to use the first order correction, we need to multiply this vector with the bra-eigenvector ψ_n^A of matrix C. But, as we recall, this vector consists of N elements, and then has N elements which are all equal to zero. This means that the inner product of these two vectors will always be equal to zero:

$$(\delta\lambda_n^A)^1 = \left\langle\psi_n^A\middle|P\middle|\psi_n^A\right\rangle = 0. \tag{20}$$

Because of this we need to use the second order correction from perturbation theory to get results different from zero. Analogously to equation 15, this correction for the eigenvalues of matrix A looks as follows:

$$(\delta\lambda_n^A)^2 = \sum_m \frac{\left\langle\psi_n^A\middle|P\middle|\psi_m^B\right\rangle \left\langle\psi_m^B\middle|P\middle|\psi_n^A\right\rangle}{\lambda_n^A - \lambda_m^B}. \tag{21}$$

Note that the 2 above the $\delta\lambda_n^A$ is not a square, but points out that we are using second order perturbation theory. Here we do not exclude $m = n$ from the sum, the reason for that exclusion in quantum mechanics is that eigenfunctions ψ_n^1 of the Hamiltonian can be written as linear combinations of other eigenfunctions ψ_n^0. With our eigenvectors we do not have that problem, because they exist in two completely different Hilbert spaces.

3.2.2 Analytic solution for a 2-by-2 matrix

When we start with a simple 2-by-2 D-matrix we can solve the problem analytically. Then the matrix looks as follows:

$$D = \begin{pmatrix} a & q \\ q & b \end{pmatrix}, \tag{22}$$

where a and b are predetermined elements and q is the perturbation. The eigenvalues without the perturbation are just a and b, and the new eigenvalues of the new perturbed matrix are computed as follows, [7]:

$$(\lambda - a)(\lambda - b) = q^2. \tag{23}$$

The terms on the left are now equal to the differences in the eigenvalues and those are exactly what we are looking for. So we can write $\delta\lambda_1$ = (λ - a) and from this it follows that $\lambda = \delta\lambda_1 + a$. Now we can write $\delta\lambda_2$ as (λ - b) = a - b + $\delta\lambda_1$. If we assume that $\delta\lambda_1$ is small compared to $|a$ - $b|$, we see from the formula that $\delta\lambda_1$ scales with q^2. This is shown in figure 2.

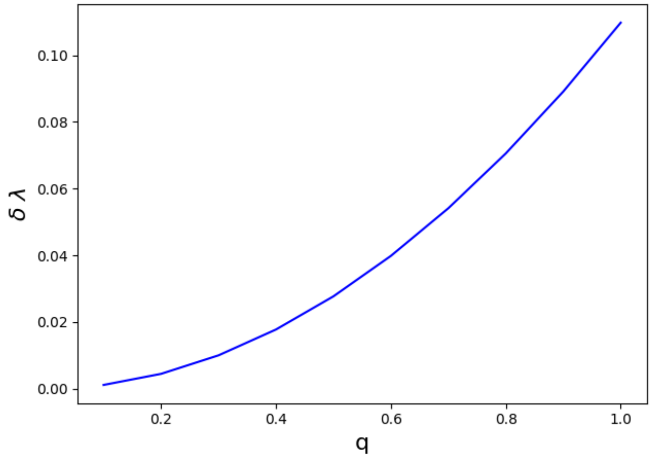


Figure 2: The difference between the first eigenvalue of the perturbed matrix and the unperturbed one, against the perturbation q. a = -10, b = -1, σ = 1.

Clearly this graph shows a quadratic relation between $\delta\lambda_1$ and q. In the results section we check if the calculations from perturbation theory approach this analytic line for some larger networks.

4 The genetic regulatory network model

In this chapter our model is not an N-species community, but we consider the gene-to-gene interactions in a single living organism. In this model the random matrix theory discussed earlier will be of great importance.

4.1 The gene-to-gene interaction model

In this model the assumption is made that gene-to-gene interaction can be modeled as chemical reactions between proteins and mRNA. The concentration of different mRNA molecules will be denoted by $x_i(t)$. Two kinds of noise of noise are important to the problem. First there is intrinsic noise, caused by processes which take place within the cell. this noise is called ξ and it is the multiplicative noise, meaning it is multiplied by x in the differential equation. The second kind of noise taken into account is extensive noise, this noise is caused by processes outside the cell that cause perturbations to the system of the cell. We call this noise η and it is independent of x. Furthermore, a matrix needs to be created which specifies the gene-to-gene interactions. This matrix is created by first constructing an Erdös-Renyi graph, which provides good results when approximating realistic gene-to-gene networks [8]. When this graph is constructed, we take the adjacency matrix from it. Then every nonzero element in the graph, so all the elements with value one, are multiplied with a number taken from a random distribution with mean zero and variance σ_A, similar to the matrix in paragraph 2.2. In the end all the diagonal elements a_{ii} are given the same constant value constant value. This value scales linearly with σ_A in the following way:

$$D \equiv -\frac{a_{ii}}{\sigma_A}. \tag{24}$$

For a large part this quotient governs the characteristic behaviour of the dynamics of the system. Without loss of generality we set $\sigma_A = 1$ for all simulations, because σ_A can be absorbed into a re-definition of the time-scale and the noise amplitudes, [8]. The model is then put in the form of the following linear differential equation [2]:

$$\frac{d}{dt}x_i = \sum_j a_{ij}(x_j - x_j^0) + \xi_i(t)(x_i - x_i^0) + \eta_i(t). \tag{25}$$

We use $x_i^0 = 1000$ for all i, which is a reasonable value when we look at the distribution of the time-averaged gene-expression trajectories of yeast [2]. Also, we set $dt = 0.1$ in all simulations. Now to correctly interpret x as a concentration of a chemical substance the following constraint is needed:

$$x_i(t) \geq 0; \forall i, t. \tag{26}$$

This constraint means that the value of a certain mRNA-type in our system is non-negative for any given time.

4.2 Stability and Eigenvalues of the interaction matrix

When we ignore the positivity condition and the noise terms, the stability of the system will depend on the largest real part of the eigenvalues of matrix A as we saw in chapter 2, so when there are no eigenvalues larger than zero, the system will always be asymptotically stable. According to the earlier discussed random matrix theory the eigenvalue spectrum will lie in a circle in the complex plane [1]. With the connectance being replaced by the probability p, we now have $\gamma = \sigma\sqrt{Np}$. For $\gamma < 1$, the system is locally stable and it will become unstable if γ becomes larger than 1. When the noise terms are also taken into account, the eigenvalue spectrum will no longer have the shape of a circle, the eigenvalues will lie closer to the real axis as we have also seen in chapter 2, with the community matrix S in comparison with the adjacency matrix A. The extra terms in matrix S can then be seen as a sort of noise added to the system which first was described fully by matrix A.

4.3 Discretization of stochastic differential equations with multiplicative noise

The stochastic terms in the differential equation have the consequence that the numerical time integration is a bit different from that of an ordinary differential equation. We now consider a general stochastic process in time which is described by a nonlinear stochastic differential of first order, [9]:

$$\dot{q} = v(q(t)) + g(q(t))\xi(t). \tag{27}$$

Here v and g are nonlinear functions of q in general and ξ is a random force which is modeled as a zero mean Gaussian process. This particular equation is difficult to solve analytically because of the noise term ξ. This means we need a discretized version, to get this we define a small time interval dt. Now when the right hand side of the equation would be integrated formally, we get

$$q(t+dt) - q(t) = \int_t^{t+dt} v(q(t'))dt' + \int_t^{t+dt} g(q(t'))\xi(t')dt'. \tag{28}$$

To further discretize, the right hand side will be expanded in powers of dt, only keeping the first order terms of this expansion gives:

$$q(t+dt) - q(t) = v(q(t))dt + g(q(t))X_1(t), \tag{29}$$

where $X_1(t)$ is equal to the integral over ξ and can be written as

$$X_1(t) = \int_t^{t+dt} \xi(t')dt' = \sqrt{2Edt}\gamma_1(t). \tag{30}$$

$X_1(t)$ is a Gaussian random number with zero mean and variance $\left\langle X_1^2(t)\right\rangle = 2Edt$. It is written in terms of $\gamma_1(t)$ which is a Gaussian random number of zero mean and variance one [9]. We now see that the first term of equation 29 is of

order dt, and the second term of order $\sqrt{dt}$. So the second term dominates for small dt. It should also be observed that in the expansion of the first term of the right hand side of equation 28, the second order expansion term will be of order dt^2. But for the second term the second order expansion term is of order dt, because the first one is of order $\sqrt{dt}$. This means that for this second term we need two terms in the expansion instead of one.
This extra term is obtained by further examination of equation 29 to lowest order in dt, [9]:

$$\frac{dg}{dq}\Big|_{q(t)}\int_t^{t+dt} dt'[q(t')-q(t)]\xi(t) \simeq \frac{dg}{dq}\Big|_{q(t)}\int_t^{t+dt} dt' g(q(t))\int_t^{t'} dt''\xi(t')\xi(t''), \tag{31}$$

which can be simplified to

$$\frac{1}{2}g(q(t))\frac{dg(q(t))}{dq(t)}X_1^2(t). \tag{32}$$

It follows that to first order of dt, the discretized form of the stochastic differential equation becomes, [9]:

$$q(t+dt) = q(t) + v(q(t))dt + g(q(t))X_1(t) + \frac{1}{2}g(q(t))\frac{dg(q(t))}{dq(t)}X_1^2(t) + \mathcal{O}(dt^{(\frac{3}{2})}). \tag{33}$$

In our gene-to-gene interaction model the extra term (the fourth term on the right hand side of this equation) is equal to

$$\frac{1}{2}qdt(N(0,\sigma))^2, \tag{34}$$

where q is equal to $x_i(t')$ and $\sigma = 1$.

4.4 Maximal Lyapunov exponents

When the differential equation is solved, we want to measure the stability of the system. A good way to do this is with Lyapunov exponents. A Lyapunov exponent quantifies the rate of separation of trajectories that start close to each other at a certain starting point. Suppose we start with two trajectories x and x'. Where x' is slightly perturbed compared to x, so they have a small separation of $||x(0)-x'(0)||$ at time $t=0$. Now at a point in time t the Lyapunov exponent is quantified as follows:

$$||x(t)-x'(t)|| = e^{\lambda t}||x(0)-x'(0)||, \tag{35}$$

where λ is the Lyapunov exponent. If we start with a simple first order differential equation $\dot{x} = \alpha x$, it is easy to see that in this case the Lyapunov exponent is just α. The rate of change can change for different values of the initial separation vector, which means there will be a spectrum of Lyapunov exponents, the Lyapunov spectrum. For the stability of the system we will only examine the largest value of this spectrum, because this one is a measure for the predictability in a dynamical system. This largest exponent of the Lyapunov spectrum is called the maximal Lyapunov exponent (MLE). When we rewrite equation 35, we get the following expression for the MLE [8]:

$$\lambda \equiv \lim_{t\to\infty} \lim_{||x(0)-x'(0)||\to 0} \frac{1}{t} \ln\left(\frac{||x(t)-x'(t)||}{||x(0)-x'(0)||}\right). \tag{36}$$

Here the limit of the separation at $t = 0$ is needed to ensure the validity of the linear approximation at any given time. If, for example, we have a larger n-dimensional dynamical system where A is an nxn matrix the system can be described by

$$\dot{\mathbf{x}} = A\mathbf{x} + f(t), \tag{37}$$

where the $f(t)$ values can be noise or other factors depending on the dynamics of the system. If $f(t) = 0$ for all x, the matrix elements completely describe the system. The matrix will have n eigenvalues, the real parts of which will be the same as the MLE's, analogous to the one dimensional case $\dot{x} = \alpha x$ where the MLE is just equal to α, [10]. In our model, this is the case if we take the original system without stochastic terms and without the positivity constraint. When we analyze the model in this way the eigenvalues of A will lie inside a circle in the complex plane, as discussed earlier in this chapter. If a network is fully connected, so with $p = 1$ and with a number of links $L = N^2$, the radius of this circle is equal to $\rho = \sigma_A\sqrt{N}$. For a non-fully connected network with $L < N^2$ the radius is equal to $\rho = \sigma_A\sqrt{L/N} = \sigma_A\sqrt{\langle k\rangle}$. The center of the circle is shifted to (a_{ii},0) because of the elements on the diagonal. Now the MLE λ can be linked to the radius ρ and can be approximated in the following way [8]:

$$\lambda(\langle k\rangle) \sim \rho(\langle k\rangle) + a_{ii} = \sigma_A(\sqrt{\langle k\rangle} - D). \tag{38}$$

This is called the low connectivity approximation. while with the positivity constraint the following approximation is made:

$$\lambda(\langle k\rangle) \sim \sigma_A(\sqrt{\frac{\langle k\rangle}{2}} - D). \tag{39}$$

This is the high connectivity approximation. The one over two factor comes from an approximation of $\langle k\rangle$ for a fully connected network. In figure 3, the eigenvalue with largest real part of the matrix A is plotted for different values of $\langle k\rangle$. Also, the theoretical lines of equations 38 and 39 are plotted.

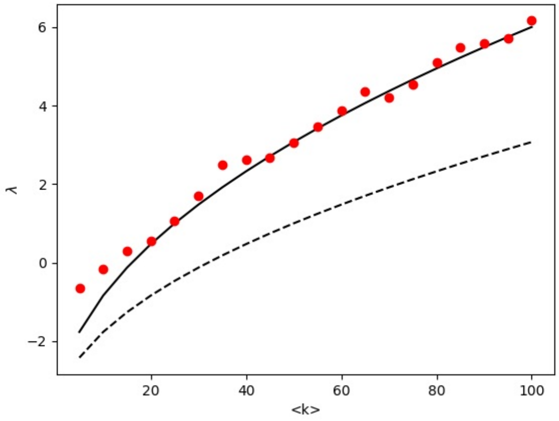


Figure 3: The red dots are values for the Maximal Lyapunov exponents without noise terms and without positivity condition. The flat and dashed black lines are the low connectivity and high connectivity approximations of equations 38 and 39 respectively. $N = 500$.

It is clear from the plot that the eigenvalues lie close to the theoretical line of the low connectivity approximation, just as the theory predicts for a system without noise. From this plot we conclude that we can approximate the MLE of a system without noise very good with the eigenvalue of largest real part of the matrix A, which points out the correlation between random matrix theory and the stability of a system described by a genetic regulatory network.

From the two approximations for λ it can be seen that there are two points where λ will be equal to 0, D^2 for the low connectivity approximation and $2D^2$ for the high connectivity approximation. These λ-values are important because when λ is equal to zero, the system will be in a marginally stable regime, and this is the regime where we want our state to be in. This means that the system is not asymptotically stable, which is when λ is smaller than 0. And that it is not in a critical regime, would be the case when λ is larger than 0. So those critical points will be $\lambda = D^2$ and $\lambda = 2D^2$. To make clear what this looks like, we show a plot from the article of reference [8] from Stokic *et al*. (2008). where they computed MLE's for a similar system.

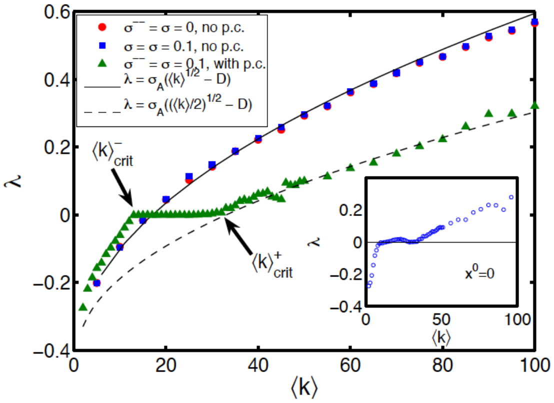


Figure 4: Plot from Stokic *et al*, reference [8].

The red circles and blue squares show their results for the case without noise and with noise respectively, both without positivity condition. The green triangles show the results when the positivity condition is taken into account. We see that with the positivity condition a λ=0 region is formed between the two critical $\langle k \rangle$-values. The small plot inside shows that when x^0 is chosen at 0 the MLE's follow a similar pattern.

5 Emergence towards criticality in different environments

Until now systems with fixed networks were considered. So for certain fixed p-values, and with that also for determined $\langle k \rangle$-values, a random network was constructed. With this information we could determine in what kind of state the system was. In this chapter the network is not fixed, it is put into some environment and the goal is to let it evolve and see in what state it will be after some time, similarly to reference [3]. Our goal is now to compare these systems to the systems described by the genetic regulatory network models. In particular, we compare the p-value of a network to a new variable in the new model, and we search for a correlation between these two.

The states will now be described by probability distribution functions. To adjust well to an environment, living systems need to alter their internal state to a state as close as possible to the state of the environment. The probability distribution function for the state of the environment (the source) is given by $P_{src}(\mathbf{s},\alpha)$ and that of the living system (the internal state) in the environment by $P_{int}(\mathbf{s},\beta)$. Here $\mathbf{s} = (s_1,\ s_2,...,s_N)$ is a string of N binary variables which represents an environmental cue that is processed by the living system [3]. The probability distributions of the state of the source and of the internal state depend on parameter sets $\alpha = (\alpha_1,\ \alpha_2,...)$ and $\beta = (\beta_1,\ \beta_2,...)$ respectively. where the parameter set α represents details of the environment and the parameter set β aims to capture the most essential features of P_{src}. The living systems are still modeled as genetic regulatory networks and now the binary vector $\mathbf{s}$ can be interpreted as the on/off state of the N genes. From reference [3] it follows that the P_{int} of living systems will be as close as possible to the P_{src} of the environment.

5.1 Kullback-Leibler divergence

To get a quantitative measure for the difference between two probability distributions P and Q, we need some background from probability theory and from information theory. First we need to define the entropy of a random variable. When we consider the discrete random variable X with probability mass function $p(x) = P\{X = x\}$, the entropy is defined as

$$H(X) = -\sum_{x} p(x) \log p(x), \tag{40}$$

and it can be interpreted as the measure of uncertainty of a random variable. It is usual to have 2 as a base for the logarithm here, the outcome is then measured in *bits*. However, also e is frequently used as a base, the entropy is then measured in *nats*. For the rest of this report base e is used unless otherwise specified. An important first notion is that the entropy is always nonnegative, this follows directly from the definition and because $0 \leq p(x) \leq 1$ for all x. Now it is our goal to get a quantitative measure for the difference between two probability distributions. In information theory this is called the relative entropy or the Kullback-Leibler divergence (KL divergence), which is defined as follows [11]:

$$D(P|Q) = \sum_x P(x) \log \left(\frac{P(x)}{Q(x)} \right). \tag{41}$$

The KL divergence is a sort of measure of 'distance' between the two distributions. However it is not a properly defined distance since it is not symmetric and because the triangle inequality does not hold. This becomes clear when we consider following example, which is taken from reference [12], page 19. We let $\mathbf{X} = \{0, 1\}$ and define the two probability distribution functions p and q on $\mathbf{X}$, with $p(0) = 1 - r$, $p(1) = r$ and $q(0) = 1 - s$ and $q(1) = s$. The KL divergence is then equal to

$$D(p|q) = (1 - r) \log \left(\frac{1 - r}{1 - s} \right) + r \log \left(\frac{r}{s} \right), \tag{42}$$

and

$$D(q|p) = (1 - s) \log \left(\frac{1 - s}{1 - r} \right) + s \log \left(\frac{s}{r} \right). \tag{43}$$

From this it follows that in general $D(p|q) \neq D(q|p)$. Only when r = s, so when $p = q$, we have $D(p|q) = D(q|p) = 0$. Just as the entropy for a single random variable, the KL divergence is always non-negative, the proof is stated below and comes from reference [12]:

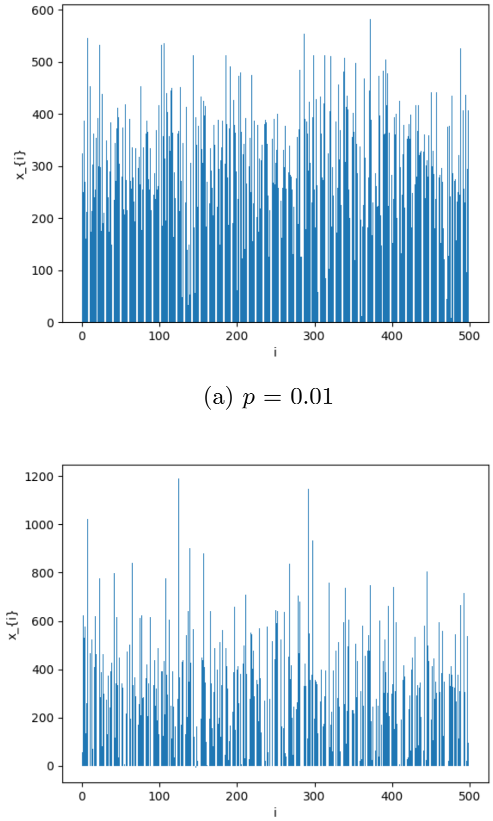
$$\begin{aligned}
D(P|Q) &= \sum_s P(s) \log \left(\frac{P(s)}{Q(s)} \right) \\
&= -\sum_s P(s) \log \left(\frac{Q(s)}{P(s)} \right) \\
&\geq \log \left(\sum_s P(s) \frac{Q(s)}{P(s)} \right) \\
&= \log \left(\sum_s Q(s) \right) \\
&= \log(1) \\
&= 0.
\end{aligned} \tag{44}$$

Here the greater-equal sign comes from Jensen's inequality, because log is a concave function.

In the case with P_{src} and P_{int} the KL divergence will be written as $D(\alpha|\beta)$, where α stands for the source distribution and β for the internal distribution. This KL divergence quantifies the loss of information when P_{int} is used to approximate P_{src}. By minimizing the KL divergence with respect to its internal parameters β, a living system generates an optimal state to cope with cues from its environment [3].

5.2 Combining the network model and the environmental model

Now that we have defined the KL divergence and have discussed some of its applications, we will try to combine it with the MLE-values from the previous chapter. First we create one x_i-trajectory with a fixed p, after this we create a number of agents y_i^1 - y_i^n that are constructed with differential equation 25. Where all these y-agents have different p-values and evolve in time to a final state. We can interpret x_i as a sort of environment for every different y_i's. In this final state the KL divergence is calculated. Now we need to interpret the distributions of the x_i and y_i's as a probability distribution discussed in the last chapter. The plot of figure 5 shows what this looks like for two different values of p.



(a) $p = 0.01$

(b) $p = 0.05$

Figure 5: Two plots of the distribution of x_i-values for different values of p, in both cases $N = 500$, time stopped at $t = 200$.

The figure with the larger p value has a lot more dispersion than the one with the smaller p value. This is what we expected because the system with the smallest p value is in a stable regime so the values of x_i should always be close to the equilibrium solution. To get sensible results for the KL divergence the values of the x-values have to be normalized. So at some point t_{end} were we do the calculations all the values of the x-vector are divided by the total sum. So for all the individuals in the population their 'probability distribution' is equal to

$$X_{norm}(i) = \frac{x_i(t_{end})}{\sum_i x_i(t_{end})}. \tag{45}$$

The formula for the KL divergence from x to y then looks as follows:

$$D(x|y) = \sum_i X_{norm}(i) \log\left(\frac{X_{norm}(i)}{Y_{norm}(i)}\right), \tag{46}$$

and in the same way from y to x:

$$D(y|x) = \sum_i Y_{norm}(i) \log\left(\frac{Y_{norm}(i)}{X_{norm}(i)}\right) \tag{47}$$

Now all the different y-trajectories are compared to the one of x and the KL divergences are computed. By doing this we can compare the KL divergence for different values of $\langle k \rangle$.

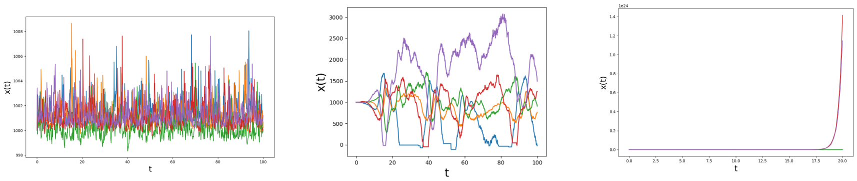
6 Results

In this chapter we show the results from the numerical simulations, the python codes are shown in the Appendix. At first, we plot some trajectories of the solution of the differential equation 25 from the genetic regulatory network model. With those results the MLE's are calculated for the different p-values. Then, we show the eigenvalues of the adjacency matrices for different values of γ and we conclude if the system described my these matrices are stable or not. After this, we look at the consequence when a perturbation is added to the system. Furthermore, we give the results for when perturbation theory is applied to the perturbed matrices. In the final part the model is combined with the KL divergence as described in the previous chapter.

6.1 The genetic regulatory model

6.1.1 The differential equation

In the next three plots we show how the concentrations x_i(t) of equation 25 evolve in time. The integration of the differential equation is done with the Euler forward method [13]. All the three plots show 5 random simulations of x_i(t) for different values of p. One for which $\lambda < 0$, the second one for the $\lambda =$ 0-region, and the last one for $\lambda > 0$.



(a) $\lambda < 0$, p = 0.01 (b) $\lambda = 0$, p = 0.035 (c) $\lambda > 0$, p = 0.19

Figure 6: Five random simulations of x_i(t) for different values of λ.

It is clear to see that $\lambda < 0$ for the first figure, since the x_i-vectors stay close to their equilibrium values. For the $\lambda =$ 0-region the values of x_i show a greater spread but they stay below certain limits, while in the last figure where $\lambda > 0$ they grow exponentially which indicates that the system is in a critically chaotic and unstable region.

6.1.2 Maximal Lyapunov exponents

In figure 3 we showed that the eigenvalues of the matrix A correspond very well with the theoretical prediction of the MLE's for the low connectivity approximation. In this section we show the results of the numerical calculation of the MLE's for a system with noise, but without the positivity condition of equation 26. Those are shown in figure 7:

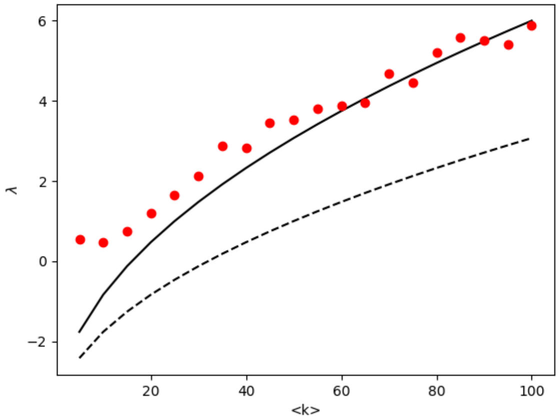


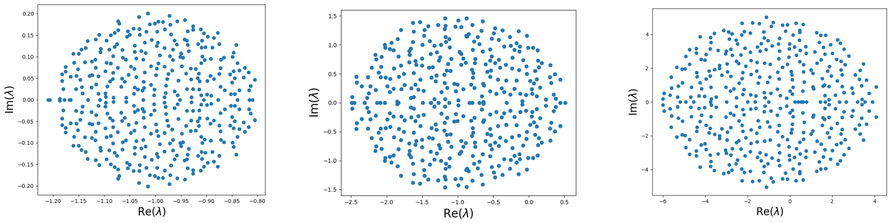
Figure 7: The red dots are values for the Maximal Lyapunov exponents with noise terms and without positivity condition. The flat and dotted black lines are the low connectivity and high connectivity approximations of equations 38 and 39 respectively. $N = 500$.

The calculations lie close to the theoretical line for values of $\langle k \rangle$ of roughly above 40. In the small $\langle k \rangle$-range the values are systematically too high.

6.2 The eigenvalues of the adjacency matrix

6.2.1 Unperturbed matrix A

In the next three plots the eigenvalues of the Adjecancy matrix A are shown for different values of γ from equation 6:



(a) $\gamma = 0.2$

(b) $\gamma = 1.5$

(c) $\gamma = 5$

Figure 8: Eigenvalues of the matrix A for different values of γ. $N = 400$, $p = 1$ and varying σ.

Sub-figure (a) is the same as the first one of figure 1. The three plots are in line with the theory because the radius of circle in the complex plane corresponds with the value of γ. Also, with a γ of 0.2, there is no eigenvalue with real part larger than zero, which implies a stable system. While for the $\gamma = 1.5$ and $\gamma = 5$ plots we see multiple eigenvalues with real part larger than zero, from which we conclude that the systems described by these matrices are more unstable.

6.2.2 Perturbed matrix D

In figure 9 we take a look at the eigenvalue distribution of the concatenated D-matrix.

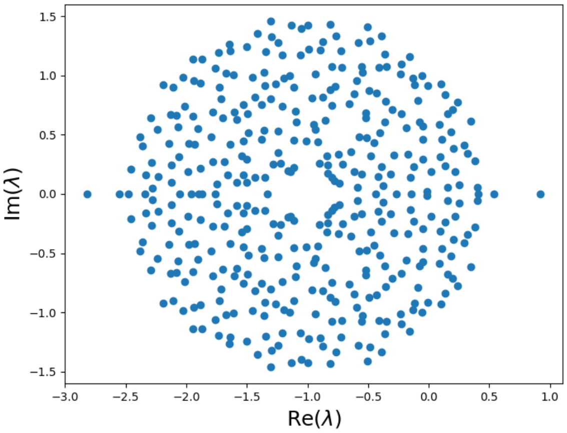


Figure 9: Eigenvalues of the perturbed matrix D, $N_A = N_B = 400$, $p_A = p_B = 1$, $\sigma_A = 0.075$, $\sigma_B = 0.01$, 100.000 perturbations with a value of 0.01.

Because of the perturbation, the absolute values of minimal and maximal value of the real parts of the eigenvalues become larger. They are not part of the circle with radius $\gamma = 1.5$ any more. We know from chapter 2 that only the eigenvalue with largest real part is important for the stability of our system. Because this eigenvalue becomes larger, the system described by the matrix becomes more unstable because of the perturbation. For a value of -0.01 instead of 0.01 for the perturbation the eigenvalue distribution is the same, so the sign of the perturbation does not matter. In the next section we will try to approximate the change of the largest eigenvalue with perturbation theory.

6.3 Perturbation theory for matrices

In figure 10 we compare the outcome of the perturbation theory to the analytic results for a 4-by-4 matrix.

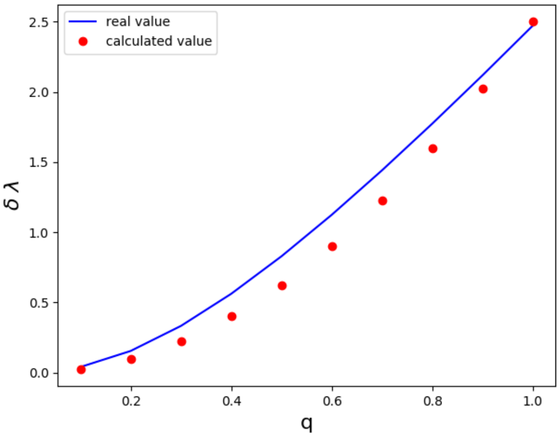
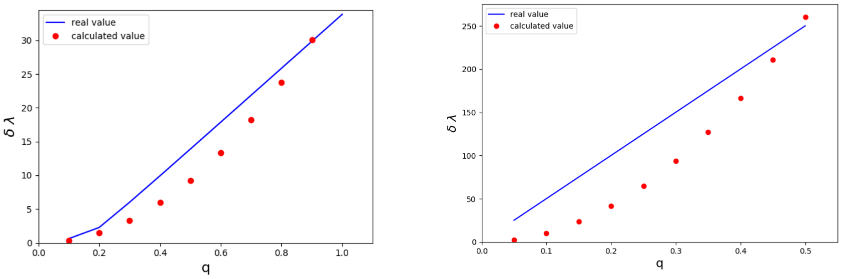


Figure 10: The real value of $\delta\lambda$ and the value calculated by perturbation theory plotted against the perturbation value. Full 4-by-4 Q-matrix.

In figure 11 we show the results for two larger matrices, for N = 40 and N = 500 respectively.

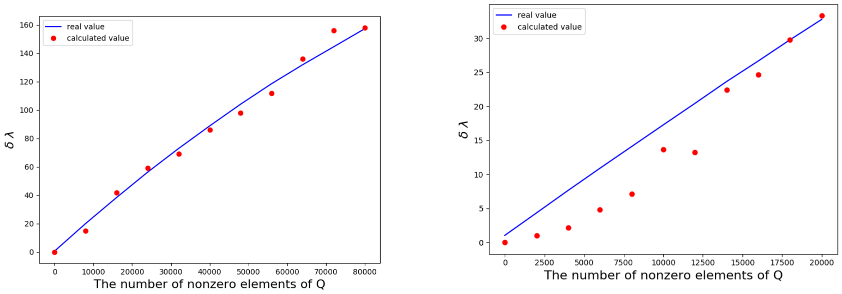


(a) N = 40, Full Q-matrix.

(b) N = 500, Full Q-matrix.

Figure 11: Two plots where the real values of $\delta\lambda$ are compared to the values calculated with perturbation theory.

We see that for the case with N = 40, there is still a quadratic correlation between $\delta\lambda$ and the value q, while for N = 500 is is completely linear. However, the calculations of the perturbation theory still seem to go up quadratically, which is why their approximation gets poor for large matrices. In figure 12 we show two results for a non-fully connected perturbation matrix Q.



(a) $N = 400$, non-zero $q_{ij} = 1$.

(b) $N = 400$, non-zero $q_{ij} = 1$.

Figure 12: Two plots where the real values of $\delta\lambda$ are compared to the values calculated with perturbation theory. For sub-figure (a), $p_A = p_B = 0$. For sub-figure (b), $p_A = p_B = 0.7$.

,

In this case, the number of non-zero elements of Q is varied. We see that the approximation from perturbation theory now scales linearly with this number of elements, just as the real value of $\delta\lambda$. However, when the p-value of the matrices becomes larger, the approximation gets poorer.

6.4 Kullback-Leibler divergence

In figure 13 we show the results for the computation of the KL divergence for different values of $\langle k \rangle$.

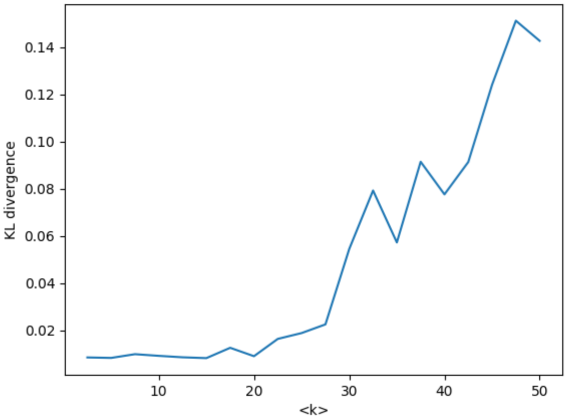


Figure 13: The KL divergence plotted against $\langle k \rangle$. The p-value of x_i to which y_i's are compared to is equal to 0.3, $N = 101$. The KL divergence values are averaged over 50 simulations of every value of p

The value of the KL divergence goes up rapidly when $\langle k \rangle$ goes above 30. This is expected because the system goes to a more critical state for larger values of $\langle k \rangle$ which is an unsustainable state for living systems, meaning that the KL divergence should be higher. We do not see the expected minimum for the optimal $\langle k \rangle$-region, between the two critical points 16 and 32, In this plot. Maybe this is because we did not run over enough simulations. Another possibility is that there is too much randomness in the differential equation to get sensible results when we stop at a certain point in time and consider the mRNA concentrations as a probability distribution. This correlation can be checked in possible follow-up research.

7 Conclusion

The main goal of this project was to research the stability of systems described by random networks when we concatenate multiple networks with each other. Also, we have coupled the parameters for quantifying the stability of systems described by gene-to-gene interaction networks, to the KL divergence from information theory. Lastly we have researched the consequence of a perturbation to such a system consisting of multiple networks. To do this we have combined the theory of genetic regulatory network models with aspects of random matrix theory and information theory. Also, we approximated certain results from random matrix theory with perturbation theory.

At first, we have seen that when we have the eigenvalue distribution of a network, we can see if the system described by this network is stable. When one or more eigenvalues have a real part which is larger than zero, the network is unstable. Also, we have shown a plot of the eigenvalue distribution of a larger matrix, consisting of two concatenated matrices together with a perturbation. In this plot the eigenvalue with largest real part fell out of the circle with radius γ, from this we concluded that a perturbed matrix is more likely to be unstable than an unperturbed one.

In the second part of this report, we have discussed some results of perturbation theory. We saw that for a small matrix, the result of the change in the largest eigenvalue because of the perturbation can be calculated analytically. We have also seen that the results of the calculation of the perturbation theory were close to this analytical curve. It appeared that for larger matrices perturbation theory could approximate the real value only for specific values for the parameters of the random matrices and the perturbation matrix.

When we defined all the needed elements from random matrix theory and perturbation theory, we could define a gene-to-gene interaction model in the same way as in references [2] and [8]. The main goal of this part was to quantify the stability of such a model, using maximal Lyapunov exponents (MLE's). When noise was taken into account, two cases were considered. A realistic model with a positivity condition, stating that there can be no negative values of molecular species in a certain system, and a model without this constraint. For the model without the constraint we could compute the numerical results quite well, while for the model with the constraint, we did not. Because of that we have showed the main graph from reference [8] to make clear what the point was, and to show that a certain MLE=0-regime should be formed in a realistic living system. We needed this result for the last part of the project.

In the final part, we defined the KullbackLeibler divergence (KL divergence) from Information Theory. We coupled the parameters from the gene-to-gene interaction model to this KL divergence to see if there is a correlation. In particular we compared several random networks with different connectances to each other and we compared their KL divergences. Our hypothesis was that the KL divergence would be minimal at a certain set of $\langle k \rangle$-values of the network. It turned out that the minimum was not really in this regime. However, We saw that the value of the KL divergence raised fast for larger values of $\langle k \rangle$ of the network. This is in line with the hypothesis because the KL divergence get larger when a system is less stable.

All in all, we have seen that for some values of random matrices, perturbation theory can predict the change in the eigenvalue spectrum because of the perturbation. In follow-up research, we can search for a more specific quantfication of this. Also, we could try to use degenerate perturbation theory when two systems share certain eigenvalues. We have also seen that we can approximate the MLE's of a system without positivity condition quite well, this study can be expanded by simulating more networks and let time run longer in the simulations. We can also search for ways to compute the MLE's more accurate for systems with positivity condition. At last, we can think of more ways to test the link between the gene-to-gene interaction model and the KL divergence. For example, we can compare a larger variety of networks with each other or we can let time run longer for better simulations.

Acknowledgements

First, I would like to thank my main supervisor Dr. J.L.A. Dubbeldam for his guidence, effort and feedback throughout the entire project and in particular for helping me improve my programming skills. Secondly, I would like to thank Dr. J.M. Thijssen, my second supervisor, for his help and guidence in the last part of the project.

At last, I would like to thank Dr. N. Parolya and Dr. T. Idema for assessing this report and for completing the committee at my final presentation and defense.

References

[1] L. Stone. The feasibility and stability of large complex biological networks: a random matrix approach. 2017.

[2] R. Hanel, M. Pochacker, and S. Thurner. Living on the edge of chaos: minimally nonlinear models of genetic regulatory dynamics. *The Royal Society*, 2010.

[3] J. Hidalgo, J. Grilli, S. Suweis, M.A. Munoz, J. Banavar, and A. Maritan. Information-based fitness and the emergence of criticality in living systems. 2014.

[4] A.-L. Barabasi. *Network Science.* Cambridge University Press, 2016.

[5] Y. Ahmadian, F. Fumarola, and K.D. Miller. Properties of networks with partially structured and partially random connectivity. 2015.

[6] D. J. Griffiths. *Introduction to Quantum Mechanics, Second Edition.* Pearson Education, Inc., 2004.

[7] J. Vermeer. *Lineaire Algebra voor Technische Natuurkunde Deel 1.* TU Delft, 2015.

[8] D. Stokic, R. Hanel, and S. Thurner. Inflation of the edge of chaos in a simple model of gene interaction networks. 2008.

[9] J. M. Sancho, M. San Miguel, S. L. Katz, and J. D. Gunton. Analytical and numerical studies of multiplicative noise. 1982.

[10] M. Braun. *Differential Equations and Their Applications, Fourth Edition.* Springer-Verlag New York Inc., 1993.

[11] S. Kullback. *Information Theory and Statistics.* John Wiley and Sons, Inc., 1959.

[12] T.M. Cover and J.A. Thomas. *Elements of Information Theory.* John Wiley and Sons, Inc., 1991.

[13] C. Vuik, F.J. Vermolen, M.B. van Gijzen, and M.J. Vuik. *Numerical Methods for Ordinary Differential Equations.* Delft Academic Press, 2016.

Appendix

Appendix A: Python codes

(A1) Eigenvalues for the adjacency matrix

```
import networkx as nx
import matplotlib.pyplot as plt
import numpy
import random

N = 400
p = 1
mean = 0
# gamma = 5
sigma = 0.25
Aii = -1
D = -(Aii/sigma)
kavg = numpy.linspace(5,100,20)
pavg = kavg/(N-1)
lowapprox = sigma*(numpy.sqrt(kavg)-D)
highapprox = sigma*(numpy.sqrt(kavg)/numpy.sqrt(2) - D)

A = []
eigvals= []
G = nx.erdos_renyi_graph(N,p)
A = nx.to_numpy_matrix(G)
for j in range(len(A)):
    for k in range(len(A)):
        A[j,k] = A[j,k] * numpy.random.normal(mean,sigma)
for l in range(len(A)):
    A[l,l] = Aii

B = A*0
for i in range(len(A)):
    B[i,i] = numpy.random.uniform(0.05,1)

C = numpy.matmul(B,A)
eig = numpy.linalg.eigvals(A)
plt.plot(numpy.real(eig),numpy.imag(eig),'o')
#plt.ylim(-0.2,0.2)
#plt.xlim(-1.1,0)
plt.xlabel('Re($\lambda$)', fontsize = 20)
plt.ylabel('Im($\lambda$)', fontsize = 20)
plt.show()
```

(A2) Perturbation theory

```
import networkx as nx
import matplotlib.pyplot as plt
import numpy
import scipy.linalg as la
import random

mean = 0
sigma1 = 1
sigma2 = 0.5
diag1 = -4
diag2 = -2
sigmaq = 8
#q = numpy.random.normal(mean,sigmaq)

N = 500
minpert = 1
maxpert = (N**2)/4
pertsteps = 11

p1 = 0.5
p2 = 0.5
G = nx.erdos_renyi_graph(N,p1)
A = nx.to_numpy_matrix(G)
H = nx.erdos_renyi_graph(N,p2)
B = nx.to_numpy_matrix(H)
C = numpy.zeros((len(A),len(A)))
for i in range(len(A)):
    for j in range(len(A)):
        A[i,j] = A[i,j] * numpy.random.normal(mean,sigma1)
for i in range(len(A)):
    A[i,i] = diag1
for i in range(len(B)):
    for j in range(len(B)):
        B[i,j] = B[i,j] * numpy.random.normal(mean,sigma1)
for i in range(len(A)):
    B[i,i] = diag2
# Larger 2Nx2N-matrix, without perturbation:
E = numpy.block([
    [A,C],
    [C,B]
    ])

evalsA, evecsA = la.eig(A)
evalsB, evecsB = la.eig(B)
```

```
evalsE, evecsE = la.eig(E)
realA1 = numpy.real(evalsA)
realA2 = numpy.real(evecsA)
realB1 = numpy.real(evalsB)
realB2 = numpy.real(evecsB)
realE1 = numpy.real(evalsE)
realE2 = numpy.real(evecsE)

for l in range(len(realA1)):
        if realA1[l] == max(realA1):
            nummer = l

diff = []
Sumlist = []
alist = numpy.linspace(minpert,maxpert,pertsteps)
for a in range(len(alist)):
    D = numpy.zeros((len(A),len(A)))
    for k in range(int(alist[a])):
        D[numpy.random.randint(N),numpy.random.randint(N)] = sigmaq
    DT = D.transpose()
    # Larger 2Nx2N perturbation-matrix, without A- and B-matrices:
    F = numpy.block([
    [C,D],
    [DT,C]
    ])
    G = E + F
    evalsG, evecsG = la.eig(G)
    # Larger 2Nx2N-matrix, with perturbation
    realG1 = numpy.real(evalsG)
    # The real d(lambda)’s:
    diff.append(abs(max(realA1)-max(realG1[0:N])))
    #Calculation of the d(lambda)’s with second order perturbation theory:
    Sum = 0
    for j in range(N):
        Sum = Sum + (numpy.inner(realA2[nummer],D.dot(realB2[j]))*\
                     numpy.inner(realB2[j],DT.dot(realA2[nummer])))/(max(realA1)-realB1[j])
    Sumlist.append(abs(Sum))
    print(a)

plt.plot(alist,diff,’b’,label=’real value’)
plt.plot(alist,Sumlist,’ro’,label=’calculated value’)
plt.xlabel(’The number of nonzero elements of Q’,fontsize=16)
plt.ylabel(’$\delta$ $\lambda$’,fontsize=16)
plt.legend()
plt.show()
```

(A3) x_i-values for the gene-to-gene interaction model

```
import networkx as nx
import matplotlib.pyplot as plt
import numpy
import random

# With noise and with positivity condition:
mean = 0
sigma0 = 1
sigma1 = 0.1
sigma2 = 0.1
Aii = -4

tbegin = 0
teind = 100
tsteps = 1001
t = numpy.linspace(tbegin,teind,tsteps)
dt = 0.1

N = 500
p = 0.01
k = p*(N-1)
klist = range(4,100)
G = nx.erdos_renyi_graph(N,p)
A = nx.to_numpy_matrix(G)

for i in range(len(A)):
    for j in range(len(A)):
        A[i,j] = A[i,j] * numpy.random.normal(mean,sigma0)
for i in range(len(A)):
    A[i,i] = Aii

x = numpy.zeros(shape=(N,len(t)))
y = numpy.zeros(shape=(N,len(t)))

# The values for x0 for al x_{i}:
for i in range(N):
    x[i,0] = 1000

# Euler Forward method:
for a in range(len(t)-1):
    for i in range(N):
        Sum = 0
        for j in range(N):
            Sum = Sum + A[i,j]*(x[j,a]-x[j,0])
        if x[i,a] + Sum*dt\
        + numpy.random.normal(mean,sigma1)*(x[i,a]-x[i,0])*numpy.sqrt(dt)\
        + numpy.random.normal(mean,sigma2)*numpy.sqrt(dt) +\
        0.5*x[i,a]*dt*(numpy.random.normal(mean,sigma1))**2 > 0:
            x[i,a+1] = x[i,a] + Sum*dt\
            + numpy.random.normal(mean,sigma1)*(x[i,a]-x[i,0])*numpy.sqrt(dt) \
            + numpy.random.normal(mean,sigma2)*numpy.sqrt(dt) +\
            0.5*x[i,a]*dt*(numpy.random.normal(mean,sigma1))**2
        else:
            x[i,a+1] = x[i,a]

#plot x-values
plt.plot(t,x[numpy.random.randint(0,N),:])
plt.plot(t,x[numpy.random.randint(0,N),:])
plt.plot(t,x[numpy.random.randint(0,N),:])
plt.plot(t,x[numpy.random.randint(0,N),:])
plt.plot(t,x[numpy.random.randint(0,N),:])
plt.xlabel(’t’,fontsize = 20)
plt.ylabel(’x(t)’,fontsize = 20)
plt.show()
```

(A4) maximal Lyapunov exponents

Without noise and without positivity condition:

```
import networkx as nx
import matplotlib.pyplot as plt
import numpy
import random

N = 500
mean = 0
sigma = 1
Aii = -4
D = -(Aii/sigma)
kavg = numpy.linspace(5,100,20)
pavg = kavg/(N-1)
lowapprox = sigma*(numpy.sqrt(kavg)-D)
highapprox = sigma*(numpy.sqrt(kavg)/numpy.sqrt(2) - D)

A = []
realeigmax = []
for i in range(len(pavg)):
    G = nx.erdos_renyi_graph(N,pavg[i])
    B = nx.to_numpy_matrix(G)
    for j in range(len(B)):
        for k in range(len(B)):
            B[j,k] = B[j,k] * numpy.random.normal(mean,sigma)
    for l in range(len(B)):
        B[l,l] = -4/sigma
    eig = numpy.linalg.eigvals(B)
    realeigmax.append(max(numpy.real(eig)))
    A.append(B)

plt.plot(kavg,lowapprox, ’k’)
plt.plot(kavg,highapprox, ’k--’)
plt.plot(kavg,realeigmax, ’ro’,)
plt.xlabel(’<k>’)
plt.ylabel(’$\lambda$’)
plt.show()
```

With noise and without positivity condition:

```
import networkx as nx
import matplotlib.pyplot as plt
import numpy
import random
import scipy

N = 500
mean = 0
sigma = 1
sigma0 = 1
sigma1 = 0.1
sigma2 = 0.1
dt = 0.1

Aii = -4
D = -(Aii/sigma)
kmin = 5
kmax = 100
k = numpy.linspace(kmin,kmax,20)
pmax = kmax/(N-1)
p = k/(N-1)
lowapprox = sigma*(numpy.sqrt(k)-D)
highapprox = sigma*(numpy.sqrt(k)/numpy.sqrt(2) - D)

teind = 100
tsteps = 1001
t = numpy.linspace(0,teind,tsteps)
x = numpy.zeros(shape=(N,len(t)))
y = numpy.zeros(shape=(N,len(t)))
lyap = []
diffmax = []

for n in range(len(p)):
    G = nx.erdos_renyi_graph(N,p[n])
    A = nx.to_numpy_matrix(G)
    for o in range(len(A)):
        for z in range(len(A)):
            A[o,z] = A[o,z] * numpy.random.normal(mean,sigma)
    for l in range(len(A)):
        A[l,l] = Aii
    for m in range(N):
        x[m,0] = 1000
        y[m,0] = 1000.02
    for a in range(len(t)-1):
        for i in range(N):
            Sum = 0
            Sum2 = 0
            for j in range(N):
                Sum = Sum + A[i,j]*x[j,a]
                Sum2 = Sum2 + A[i,j]*y[j,a]
            x[i,a+1] = x[i,a] + Sum*dt + \
            numpy.random.normal(mean,sigma1)*x[i,a]*numpy.sqrt(dt) +\
            numpy.random.normal(mean,sigma2)*numpy.sqrt(dt)\
            + 0.5*x[i,a]*dt*(numpy.random.normal(mean,sigma1))**2
            y[i,a+1] = y[i,a] + Sum2*dt + \
            numpy.random.normal(mean,sigma1)*y[i,a]*numpy.sqrt(dt) +\
            numpy.random.normal(mean,sigma2)*numpy.sqrt(dt)\
            + 0.5*y[i,a]*dt*(numpy.random.normal(mean,sigma1))**2
    diffmatx = x[:,int((tsteps-1)/2):int((tsteps-1))]
    diffmaty = y[:,int((tsteps-1)/2):int((tsteps-1))]
    lyap.append(((1)/(teind/2.5))*(numpy.log(abs(diffmatx-diffmaty).sum()\
    /(sum(abs(x[:,int((tsteps-1)/2-1)]-y[:,int((tsteps-1)/2-1)]))))))
    print(n)

plt.plot(k,lowapprox,’k’)
plt.plot(k,highapprox, ’k--’)
plt.plot(k,lyap,’ro’)
plt.xlabel(’<k>’)
plt.ylabel(’$\lambda$’)
plt.show()
```

(A5) Kullback-Leibler divergence

```
import networkx as nx
import matplotlib.pyplot as plt
import numpy
import random

# With noise and with positivity condition:
mean = 0
sigma0 = 1
sigma1 = 0.1
sigma2 = 0.1
diag = -4
clist = 50

tbegin = 0
teind = 200
tstappen = 2001
t = numpy.linspace(tbegin,teind,tstappen)
dt = 0.1

N = 101
p = 0.3
plist = numpy.linspace(0.05,0.5,10)
klist = plist*(N-1)
G = nx.erdos_renyi_graph(N,p)
A = nx.to_numpy_matrix(G)
for i in range(len(A)):
    for j in range(len(A)):
        A[i,j] = A[i,j] * numpy.random.normal(mean,sigma0)
    A[i,i] = diag

# Creating x- and Xnorm- vector
x = numpy.zeros(shape=(N,len(t)))
for i in range(N):
    x[i,0] = 1000

for a in range(len(t)-1):
    for i in range(N):
        Sum = 0
        for j in range(N):
            Sum = Sum + A[i,j]*(x[j,a]-x[j,0])
        if x[i,a] + Sum*dt + \
        numpy.random.normal(mean,sigma1)*(x[i,a]-x[i,0])*numpy.sqrt(dt) \
        + numpy.random.normal(mean,sigma2)*numpy.sqrt(dt) + \
        0.5*x[i,a]*dt*(numpy.random.normal(mean,sigma1))**2 > 0:
            x[i,a+1] = x[i,a] + Sum*dt + \
            numpy.random.normal(mean,sigma1)*(x[i,a]-x[i,0])*numpy.sqrt(dt) \
            + numpy.random.normal(mean,sigma2)*numpy.sqrt(dt)\
            + 0.5*x[i,a]*dt*(numpy.random.normal(mean,sigma1))**2
        else:
            x[i,a+1] = x[i,a]

Xnorm = x[:,teind]/sum(x[:,teind])

for i in range(len(Xnorm)):
    if Xnorm[i] < 0 or Xnorm[i] == 0:
        Xnorm[i] = 1e-100

Hlist = []
Blist = []
for i in range(len(plist)):
    H = nx.erdos_renyi_graph(N,plist[i])
    B = nx.to_numpy_matrix(H)
    for j in range(len(B)):
        for k in range(len(B)):
            B[j,k] = B[j,k] * numpy.random.normal(mean,sigma0)
        B[j,j] = diag
    Blist.append(B)
    Hlist.append(H)

# Creating y- and Ynorm- vectors
y = numpy.zeros(shape=(N,len(t)))
for i in range(N):
    y[i,0] = 1000

KLdivlist2 = numpy.zeros(len(plist))
for c in range(clist):
    KLdivlist = []
    for b in range(len(Blist)):
        for a in range(len(t)-1):
            for i in range(N):
                Sum2 = 0
                for j in range(N):
                    Sum2 = Sum2 + Blist[b][i,j]*(y[j,a]-y[j,0])
                if y[i,a] + Sum2*dt + \
                numpy.random.normal(mean,sigma1)*(y[i,a]-y[i,0])*numpy.sqrt(dt) \
                + numpy.random.normal(mean,sigma2)*numpy.sqrt(dt) + \
                0.5*x[i,a]*dt*(numpy.random.normal(mean,sigma1))**2 > 0:
                    y[i,a+1] = y[i,a] + Sum2*dt + \
                    numpy.random.normal(mean,sigma1)*(y[i,a]-y[i,0])*numpy.sqrt(dt) \
                    + numpy.random.normal(mean,sigma2)*numpy.sqrt(dt) + \
                    0.5*x[i,a]*dt*(numpy.random.normal(mean,sigma1))**2
                else:
                    y[i,a+1] = y[i,a]
            if a == len(t)-2:
                Ynorm = y[:,a]/sum(y[:,a])
                KLdiv = 0
                for k in range(len(Ynorm)):
                    if Ynorm[k] < 0 or Ynorm[k] == 0:
                        Ynorm[k] = 1e-100
                    KLdiv = KLdiv + Xnorm[k]*numpy.log(Xnorm[k]/Ynorm[k])
                KLdivlist.append(KLdiv)
    KLdivlist2 = KLdivlist2 + KLdivlist

KLnorm = KLdivlist/sum(KLdivlist)
KLnorm2 = KLdivlist2/sum(KLdivlist2)
plt.plot(klist,KLnorm2)
plt.xlabel(’<k>’, fontsize = 18)
plt.ylabel(’KL divergence’, fontsize = 18)
plt.show()
```