Reverse engineering of genetic networks with time delayed recurrent neural networks and clustering techniques

Dissertation

submitted to the
Combined Faculties
for the Natural Sciences and for Mathematics
of the Ruperto-Carola University of Heidelberg, Germany

for the degree of

Doctor of Natural Sciences

presented by

M. Sc. David Camacho Trujillo
born in México City, México

Oral-examination: .....

.....
.....
.....

Referees: Prof. Dr. Ursula Kummer
. P.D. Dr. Ursula Klingmüller

Dedicated to:

Sarah

&

Tere

&

Arturito

INDEX

Summary

In the iterative process of experimentally probing biological networks and computationally inferring models for the networks, fast, accurate and flexible computational frameworks are needed for modeling and reverse engineering biological networks. In this dissertation, I propose a novel model to simulate gene regulatory networks using a specific type of time delayed recurrent neural networks. Also, I introduce a parameter clustering method to select groups of parameter sets from the simulations representing biologically reasonable networks. Additionally, a general purpose adaptive function is used here to decrease and study the connectivity of small gene regulatory networks modules.

In this dissertation, the performance of this novel model is shown to simulate the dynamics and to infer the topology of gene regulatory networks derived from synthetic and experimental time series gene expression data. Here, I assess the quality of the inferred networks by the use of graph edit distance measurements in comparison to the synthetic and experimental benchmarks. Additionally, I compare between edition costs of the inferred networks obtained with the time delay recurrent networks and other previously described reverse engineering methods based on continuous time recurrent neural and dynamic Bayesian networks. Furthermore, I address questions of network connectivity and correlation between data fitting and inference power by simulating common experimental limitations of the reverse engineering process as incomplete and highly noisy data.

The novel specific type of time delay recurrent neural networks model in combination with parameter clustering substantially improves the inference power of reverse engineered networks. Additionally, some suggestions for future improvements are discussed, particularly under the data driven perspective as the solution for modeling complex biological systems.

Zusammenfassung

Für den iterativen Prozess der experimentellen Erforschung biologischer Netzwerke und der computergenerierten Ableitung von Modellen für diese Netzwerke werden schnelle, fehlerfreie und flexible Programmiergerüste benötigt, um biologische Netzwerke zu modellieren und um sie zu rekonstruieren. In dieser Arbeit stelle ich ein neuartiges Modell vor, das genregulierte Netzwerke darstellt, indem zeitverzögerte, rekurrente, neuronale Netzwerke benutzt werden. Zudem führe ich eine Methode des Parameter-Clusterings ein, die Parameter-Set-Gruppen, die biologisch sinnvolle Lösungen darstellen, aus den Simulationen auswählt. Zusätzlich wird hier eine generelle, lernfähige Funktion eingesetzt, um die Konnektivität kleiner genregulierter Netzwerke zu verringern und um diese zu untersuchen.

In dieser Dissertation wird die Leistungsfähigkeit dieses neuartigen Modells, die Dynamik genregulierter Netzwerke aus synthetischen und experimentellen Datensätzen von Zeitreihen der Gen-Expression zu simulieren und deren Topologie abzuleiten, aufgezeigt. Die Qualität der abgeleiteten Netzwerke bestimme ich mit Hilfe von Graph-Edit-Messungen im Vergleich zu den synthetischen und experimentellen Bezugswerten. Außerdem vergleiche ich den Arbeitsaufwand der von den zeitverzögerten rekurrenten Netzwerken abgeleiteten Netzwerke und anderer bereits beschriebener Rekonstruktionsmethoden, die auf zeitkontinuierlichen-rekurrenten und dynamischen-bayesischen Netzwerken basieren. Darüber hinaus befasse ich mich mit Fragen der Netzwerk-Konnektivität und der Korrelation zwischen der Datenanpassung und der statistischen Power der Inferenz, indem ich bekannte experimentelle Einschränkungen des Rekonstruktionsprozesses, wie unvollständige oder höchst rauschbehaftete Datensätze, simuliere.

Dieses neuartige und spezielle, zeitverzögerte, rekurrente, neuronale Netzwerk verbessert zusammen mit dem Parameter-Clustering wesentlich die Ableitungskraft der rekonstruierten Netzwerke. Zudem werden einige Anregungen für zukünftige Verbesserungen erörtert, insbesondere aus der datengestützen Perspektive als der Lösungsstrategie für die Modellierung komplexer biologischer Systeme.

Personal Words

I would like to acknowledge in this dissertation to those persons and institutions that made this work possible.

The support of Professor Ursula Kummer is of special importance due to the circumstances of this work. Therefore, I would like to sincerely acknowledge her for this opportunity to finish all these years of work on a nice way.

I want to acknowledge Professor Randall Beer, who thanks to the internet could provide me with punctual but decisive directions to understand and develop my theoretical model. Analogous are the contributions from Professor Paul Johnson as well as the SWARM community who always helped me to implement my model under the SWARM philosophy. Additionally, I would like to thanks Dr. J.J. Merello for his GA code and directions to use it.

Undoubtedly, this work could not be possible without the support from Sarah. In many ways she has supported me and was at my side in difficult moments. Hence, I want to set this in words. Many thanks for all your support, Saritah.

Special thanks to my family because despite they are not geographically close to me, they always were present and support me with their comprehension. Special thanks to my sisters Rebekita and Lendy, for their tenderness and shift responses when needed.

I acknowledge the opportunity to develop this work at the iBIOS group from the Deutsches Krebsforschungszentrum (DKFZ) and the Viroquant group at Bioquant from the University of Heidelberg.

This work could be developed by the support of the DAAD who I would like to acknowledge, because it is more than a building - they always have been a human Institution.

In this sense, I would like to acknowledge my former University UNAM, because despite of being massive, it has a quality similar to the University of Heidelberg and gave me my formation for free. Once I heard that that massive universe imprint over us some sort of nationality, and I believe there is a little truth in that.

I would like to thank all my friends that have been at my side, in one or another way.

Finally, I would like to thanks to my own “dickopfness” because in many senses, it was not planned for me to be at this point.

List of abbreviations

Activating Protein 1	AP-1
Adaptive time-delay neural network	ATNN
ATP: protein phosphotransferase (cAMP-dependent)	PKA
Boolean network	BN
Carcinoembryonic antigen-related cell adhesion molecule 1	CEACAM1
Catalytic subunit of the main cell cycle -	
- cyclin-dependent kinase CDK (yeast)	cdc28
Cell division control protein 15 (yeast)	cdc15
Cell division cycle 14 protein (yeast)	Cdc14
Cell division cycle 20 protein (yeast)	Cdc20
Cell division cycle 20-like protein 1(yeast)	Cdh1
Central nervous system	CNS
Connectivity	K
Continuous time recurrent neural network	CTRNN
Cyclin-dependent kinase (yeast)	Mcm1
Deoxyribonucleic Acid	DNA
Dynamic Bayesian network	DBN
Early growth response protein 1	EGR1
Epidermal grow factor receptor	EGF-R
Epidermal grow factor	EGF
Escherichia Coli	E.coli
FBJ murine osteosarcoma viral oncogene homolog	FOS
Fetal calf serum	FCS
G1/S-specific cyclin (yeast)	Cln 3
G1/S-specific cyclins (yeast)	Cln 1/2
G2/mitotic-specific cyclin 1/2(yeast)	Clb 1/2
Gene regulatory network	GRN
Gene Set Enrichment Analysis	GSEA
Genetic algorithm	GA
Glyceraldehyde-3-phosphate dehydrogenase (NADP+)	GAPDH
Granulocyte–macrophage colony-stimulating factor	GM-CSF

Graph edit distance	GED
Gray (absorbed radiation units)	Gy
Hepatocyte growth factor	HGF
Human skin dermal fibroblasts	HDF
Immortalized human keratinocytes	HaCaT
Integrin beta 6	ITGB6
Integrin, alpha V	ITGAV
Irradiated human skin dermal fibroblasts	HDFi
Keratinocyte growth factor	FGF-7
Laminin, alpha 3	LAMA3
Laminin, gamma 2	LAMC2
Means square error	MSE
Messenger ribonucleic acid	mRNA
Micro ribonucleic acid	miRNA
Mismatch	MM
Multiple recurrent neural networks	MRNN
Multiprotein bridging factor (yeast)	MBF
Nondeterministic polynomial-time hard	NP-hard
Not pruning	NP
Open reading frame	ORF
Ordinary differential equations	ODE
Perfect match	PM
Plasminogen activator, urokinase	PLAU
Plasminogen activator, urokinase receptor	PLAUR
Prostaglandin-endoperoxide synthase 2	PTGS-2
Pruning	P
Random Boolean networks	RBN
Regulatory protein SWI4 (yeast)	SBF
Regulatory protein SWI5 (yeast)	Swi5
Reverse engineering	RE
Reverse transcriptase-polymerase chain reaction	RT-PCR
Ribonucleic acid	RNA
Ribonucleic acid polymerase subunit II	RNA pol II
Simulating annealing	SA

S-phase entry cyclin 5/6	(yeast)	Clb5/6
Stromal derived factor-1		SDF-1
Substrate and inhibitor of the cyclin-dependent protein kinase CDC28	(yeast)	Sic1
Thiamine-adenine promoter consensus sequences		TATA box
Time delayed neural networks		TDNN
Time delayed recurrent neural network		TDRNN
Transcription factors		TF
Transcription factor Jun B		JUNB
Variance stabilization normalization		VSN
v-ets erythroblastosis virus E26 oncogene homolog 1		ETS1

General Motivation

A group of technologies like microarrays, CGH or mass spectrometry, has become part of the standard laboratory experiments all around the word. These technologies allow us to measure thousands of genes, hundreds of proteins or other cellular components like mRNA´s at the same time. All this information is usually primary stored into databases but generally it analysis is far to be finished. One reason for this situation is that the traditional one to one "cause-effect" correlation, typically used in biology, is not applicable to these large data sets. To handle this information a new kind of approaches has been developed during the last years. Approaches able to store, analyze and develop models for a large number of variables.

One area that is deeply influenced by experimental high throughput data generation technologies is the analysis of gene regulation of the mammalian cells. Because of its complexity and implications in different areas like evolution or drug target generation, gene regulation is widely studied by theoretical works. The holistic integration of gene regulation dynamics has just begun, and it is clear that only the iterative work between lab data driven experiments and theoretical work will be able to generate a new paradigm in the area. In this multidisciplinary context the present work is circumscribed. To understand the goals, achievements and limitations of this work, some basics topics will be described about gene regulation complexity (Chapter 1), and the most relevant related theoretical work developed until now (Chapter 2). In Chapter 3 the methodology used in this thesis is described. A comparative study to with similar approaches is achieved in the results section (Chapter 4) as well as the presentation of results obtained by applying the approach described in the present work, to experimental data. Lastly, the analyses of the results as well as collateral topics are described on the discussion section (Chapter 5) and some final words and outlook is described in the Conclusion section (Chapter 6).

1. Biological context

1.1 Gene regulation

Gene regulation is a complex and not well-understood process. It has several mechanisms to control itself which act at different levels in time scale, cellular space and molecular mechanisms. Here, some of these mechanisms will be described to highlight the implications and restrictions they impose on the theoretical models intended to capture gene regulation behavior.

1.2 Basal transcription apparatus

The basal transcription apparatus is not part of the gene regulation mechanism by itself. However, it is important to remark that its presence is a necessary condition in order to transcribe any gene. Therefore, it is important to know some structural aspects of it, which plays a role for the design of gene regulation kinetic related models (Mjolsness and Sharp, 1991).

While the *enzymatic* behavior of transcription is due to the basal transcription apparatus bound to the tetrameric RNA polymerase II, *substrates* are the relative free diffusible nucleotide bases and the highly conserved thiamine-adenine promoter sequences (TATA boxes, here on) on the DNA. Finally, the mRNA is the obtained *product*. From here on, it should be clear that the TATA boxes do not form part of the same chemical liquid phase as the other substrates. TATA boxes are part of an extreme long polymer associated to thousands other proteins conforming a dynamical semi-solid phase system, the so-called Chromatin. Therefore, traditional kinetic models (as Michaelis Menten and others) should not be applicable here, because they presuppose a freely diffusive Brownian motion of all substrates along an homogeneous liquid phase media, meaning homogeneous concentration.

The basal transcription apparatus plus the tetrameric RNA pol II is a complex of more than 20 proteins that is *in situ* sequentially assembled This multimeric complex requires additional proteins to initiate its formation; the so-called *Mediator* (Lewis and Reinberg, 2003) is part of those required additional proteins. The Mediator is about 20 proteins in size and this large multimeric protein complex requires additional proteins to initiate the transcription, the so-called Transcription Factors (TF here on). These TF are the triggering initial step in gene transcription activation.

The basal transcription apparatus plus Mediator and additional TF is a large multimeric complex, which could have the size of more than 60 proteins. Hence, it is clear that gene transcription initiation requires some structural conditions on the DNA super structure as accessibility to avoid steric impediments. This accessibility is controlled by other regulatory mechanisms that promote chromatin relaxation (Cremer and Cremer, 2001). Additional aspects like DNA malleability (3D curvature of the DNA) or stochastic fluctuation in access generated by the Chromatin "breathing" are a matter of discussion. For sure, these processes play also their roll, but the problem is to know when and how intense every regulatory mechanism contributes to the global behavior. From here on, it should be clear that chromatin accessibility is a tri dimensional level of inhomogeneity and a source of gene activity regulation. Models oriented to represent gene reaction-diffusion kinetics should take this into account.

Concerning the product (mRNA) kinetics; since the mRNA is a polymer, it follows a multiple-step-synthesis. Additionally, as this mRNA product is not a monotonic polymer, its step-by-step formation implies a more complex process (among other processes like *translocation*, *strand separation* etc.) known as *nucleotide selection*. This means that the rate of the mRNA production is not diffusion limited, and has slight variations depending on the template sequence and other context dependent proteins. On the other hand, once the basal transcription machinery is induced it activates the RNA polymerase, and in turn it moves along the gene performing the transcription. The consequence of this RNA polymerase displacement is that another RNA polymerase is able to bind the initiation complex and initiate simultaneously a new transcription process.

Therefore, even though there is only one functional a copy of every gene, it is difficult to quantify the maximum number of transcription complexes at a certain time. Owing to these last two considerations, it is practically impossible to define a transcription rate constant for any gene, because the rate of transcription for every gene is a continuous time-and-context dependent process.

1.3 Transcription factors

Transcription factors belong to a large but limited set of families of proteins that share functionality, in human they are about 300 (Itzkovitz, et al., 2006) TF can recognize (the mechanism is family specific) (Itzkovitz, et al., 2006) pattern sequences into the DNA along the so-called promoter region of every gene. Once the TF are bound to the promoter region of a given gene, they can interact with proteins from the basal transcription apparatus and together recruit the RNA polymerase II initiating the transcription of that gene. The activation *strength* is function of the concentration and physical interactions among the TF that activate a given gene at a certain time.

It has been shown (van Nimwegen, 2003) that the total number of TF (N) of any specie scales with it genome size (G) as a power-law ($N \sim G^{1.9}$ Prokaryotes, $N \sim G^{1.3}$ Eukaryotes). However, all of them are not present at the same time nor with the required concentration to activate their target genes. Instead, TF need to interact with some others proteins in order to activate one gene. Usually, they form dimers (homo and hetero dimers) and in turn form quaternary complexes at the promoter regions (Pilpel and Sudarsanam, 2001), where typically more than one complex regulates its activity. Furthermore, it has been proposed that the total number of TF per family correlates with the number of degrees of freedom (number of base pairs recognized by family, ranging from 4 to 96 in humans) in the binding mechanism. However, an overlap of sequence recognition occurs among different TF, probably to make the system more robust through redundancy.

Regarding the concentration of the TF in the nucleus, the general idea is that the local concentrations of TF are responsible for the activation of their target genes.

The local variation of TF concentrations is influenced by their transport (internalization into the nucleus from the cytoplasm) which in turn is often regulated by their activation (usually by phosphorylation) and negatively regulated by their deactivation and or degradation. However, once the TF are in the nucleus, local variations of TF concentrations occur due to the interaction with other already focalized (attached to) proteins at certain DNA regions, the so-called enhancers.

Interactions between TF and its promoter targets on the DNA are not covalent, the real picture is a dynamical stochastically process where TF are bound and unbound permanently to the DNA. In this sense TF activation state often plays a central role in their activity because often the activated (usually phosphorylated) transcription factors exhibit a higher affinity for the DNA recognition site, but when the concentration of inactive TF is high enough, then the none activated TF could displace the already bound active TF. Therefore, at this TF regulation level, gene activity again is a continuous time combinatorial process, function of TF identities, their transport, local concentrations and often it activation state (phosphorylated or not).

1.4 Enhancers-Insulators

These are short regions on the DNA, which can facilitate the transcription of (cis) genes at a relative long-distance. They are defined like *distant-acting cis-regulatory elements (Blackwood and Kadonaga, 1998)* but their precise mechanism of acting is still not clear. There is evidence that enhancers increase the probability of genes on their surroundings to be transcribed. Enhancers do their task probably by increasing the local concentration (known as nuclear localization) of TF, but there are some other proposed mechanisms such as; chromatin or nucleosomes remodeling, superhelical tension (to facilitate chromatin accessibility) and direct interaction with associated proteins and the transcription basal machinery. Enhancers also have their functional counterpart on the so-called *insulators* (West, et al., 2002), which probably directly inhibit the functioning of enhancers, but probably they promote gene repression by other mechanism like Chromatin condensation.

However, there is not enough information on the final balance between enhancers-insulators until now. Therefore this could be modeled like a none-specific *Bias* of the gene activation process.

1.5 Post-transcriptional regulation of the mRNA

At this point two mechanisms are the most relevant to take into account for modeling gene regulatory networks: alternative splicing and interference RNA. A good description of both could be found elsewhere therefore here the focus is just on the implications for the area of modeling gene regulatory networks.

1.5.1 Alternative splicing

The mRNA is transcribed as a precursor containing intervening sequences (introns). These sequences are subsequently removed such that the flanking regions (exons) are spliced together to form mature mRNA. Alternative splicing pathways generate different mRNAs encoding distinct protein products, those increasing the coding capacity of genes. The resulting proteins may exhibit different and sometimes antagonistic (Cremer and Cremer, 2001) functional and structural properties, as binding affinity, intracellular localization, enzymatic activity, stability and post-translational modifications, and may inhibit the same cell with the resulting phenotype being the balance between their expression levels. Alternative splicing can also act as an on–off gene expression switch by the introduction of premature stop codons (Feyzi, et al., 2007).

The alternative splicing mechanism is achieved by a ribonucleo-protein structure called spliceosome, but most of the splicing regulation that is not part of the basal spliceosome is known to be undertaken by families of splicing regulatory proteins. These splicing factors bind to signals in the vicinity of the exon and promote the exon's inclusion or exclusion by activating or inhibiting the function of the splice site. The number of classes and characteristics of these regulatory proteins and their RNA binding sites are relatively little known and are currently under active investigation (Irimia and Roy, 2008; Pettigrew and Brown, 2008; Solis, et al., 2008).

These properties from alternative splicing should change the vision of gene regulatory networks, defined by a fixed topology and a set of rules of interactions, by a more dynamical concept.

Especially now, that it is accepted that alternative splicing is not the exception but the rule for about 40-60% of the human genes (Downes, 2004; Stamm, 2002). However, until now just a few information concerning alternative splicing regulation is available and therefore it could not to be included into gene regulatory networks models. It should be mentioned that alternative splicing is not a problem for the reverse engineering of gene regulatory networks task by it self, rather for the prediction of gene network behavior (see the difference on chapter 2 between *inference power* and *prediction power*). It is, at the cellular level all signaling mechanisms are context dependent, and what applies for a given cell at certain development stage, is not applicable to another cell line or the same cell line at another cell cycle stage or under different environmental conditions. Hence, one should carefully extrapolate what is encountered in a cell line to another cell line or the same cell but under different conditions. The distinction between these two situations will be further explained on the 2nd Chapter.

1.5.2 RNA interference

Less than 2% of the human genome is translated into proteins (He, 2004), yet more than 40% of the genome is thought to be transcribed into RNA (Ben-Dov, et al., 2008). The vast fraction of untranslated RNA's includes several kinds of functional non coding RNA, like snRNA (Spliceosomal and U7), snoRNA, telomerase RNA, SRP RNA (protein trafficking), tRNA, TSK RNA (transcription elongation), and a group of RNA that interfere with the expression of genes: siRNA, miRNA, piRNA. This last family has different mechanisms to act, but share the characteristic of silencing the expression of genes. The siRNA is oriented to silence exogenous genes like those present in viruses (Juliano, et al., 2008) piRNA is utilized in mammalian germ cells (Paddison, 2008) and finally the miRNA is utilized by plants and animals cells to selectively silence genes during development (Boutros and Ahringer, 2008)

differentiation or proliferation as part of another level of gene regulation.

miRNA regulation is similar to gene expression, they have promoters and enhancers, but also could be transcribed as part of some gene and later spliced during the mRNA maturation. However, they are initially transcribed as pri-mRNA, removed by an enzyme (drosha) in the nucleus, exported to the cytoplasm as pre-mRNA, where are further removed by another enzyme (dicer) generating mature miRNA. It is there, in the cytoplasm where miRNA´s finally meets its target mRNA, basically by Watson-Crick complementarities'. Once miRNAs meet their target mRNA, they decrease its function by different possible ways like direct cleavage (mostly in plants), mRNA deadenylation, affecting the stability of their target mRNA and inhibiting translation (mainly in animals).

Currently there are 328 miRNA's annotated in the human genome (Chen and Rajewsky, 2007), but is thought that there are more than 1000. Interestingly, even though this miRNA is just 22 bases long they exhibit a relative highly conserved sequence, and it has been shown that every class of miRNA could affect several (hundreds) of different genes (He, 2004). It is thought that more than 30% of the human genes are also regulated by miRNA's. Therefore this mechanism should be included explicit or implicitly by any model for gene regulatory networks. Nevertheless, the problem to include this mechanism is that in animals the main mechanism of gene repression through miRNA's is in proteic translation repression. It means, it does not affect the synthesis of mRNA and therefore would not be reflected at the transcriptome level and in case, also would not be reflected in the mRNA microchip technology.

Ideally, there should be information regarding the stability of mRNA, the relationship of mRNA translated to proteins, the proteins half life, the state of activity of proteins involved in gene regulation, localization of this proteins, complex formations etc. to create a dynamical model of gene regulatory networks. But the case of miRNA negative regulation is of particular relevance for reverse engineering of gene regulatory networks, because it acts as a negative regulation of genes. Once again, the distinction between both areas will be further explained in Chapter 2.

1.5.3 Dimensional in-homogeneities

As it has been mentioned before, the accessibility of the genes into the chromatin is a general mechanism of gene expression repression (Cremer and Cremer, 2001). However, chromatin structure is a field under investigation that proved that chromatin has too many levels of control (Lanctôt, et al., 2007). A starting point to exemplify the chromatin complexity is the super-coiling structure with its seven levels of packaging that, according with the previously described, function as a basal repressive steric barrier. Other gene expression regulatory mechanisms at the Chromatin level are the chromosome territories, acethylation and methylation of histones, interaction with the nuclear (actin) matrix, translocations of genes etc. In general all these processes promote or inhibit gene activity by means of favoring or inhibiting diffusion of the reactants (Misteli, 2001).

Additionally to the three dimensional in-homogeneities, there is another level of gene regulation that is far beyond of the scope of the actual models. This additional complexity comes from the fact that some of process previously described as activation of TF, methylation and acethylation of histones are precisely regulated by cytoplastmatic events called signal transduction. Cell signal transduction is a large and very important area of study under intense research that is beyond the scope of this work.

However, the emergent aspect to notice here is that the previously described in-homogenic diffusive processes, and the cell signal transduction processes of communicates with each other through a fiscal barrier: the nuclear envelope (Auboeuf, et al., 2007). This fiscal barrier controls with high precision the fluxes of molecules between focalized processes and signal transduction by the means of nuclear pores that often uses *active transport* (against concentration gradient, an energy dependent process). Hence, a multi compartment modeling should be applicable if simultaneous data of gene activity and signal transduction where available. However, this also could be modeled by two means: reaction diffusion models or Ordinary differential equations (ODE, from here on) with time delays.

Reaction diffusion models are usually based on partial differential equations, (despite some other possible approaches like cellular automata have being used) and have shown to be very useful if the adequate data is available. However, to model the multi-compartment fluxes coupled with diffusion reaction is, in general, difficult to be modeled. On the other hand time delayed models are an alternative to model the final effect of these complex processes as transport and focalizations of metabolites.

This good performance of time-delayed models has the price of not representing a precise mechanics of the original system (kinetics constants) but instead they represent the global behavior and structure through a semi-parametric model (have being called phenomenological ones). However, depending on the goals of a research project this last could be very useful.

2. Reverse engineering and modelling of genetic network modules

2.1 Related work

The task of recovering the *wiring between elements* of a given system and the *rules* governing their interaction, using data of its dynamical behavior is known as *reverse engineering*. In the context of functional genomics, it means finding out which genes regulate which others, how and under which circumstances. In other words, it is the task of finding the *topology* of the *gene regulatory network* (GNR) related to a particular cell line. Several experimental works and theoretical approaches have been performed in this field. Since there are good reviews on the reverse engineering of gene regulatory networks, in the present chapter I will just briefly describe some of these works, starting with some common agreements in reverse engineering of gene regulatory networks field, followed by a description of the most important theoretical approaches on this field.

2.2 General concepts

Into a given cellular system, gene activity influences directly or indirectly the activity of other genes. This system could be uni-cellular or pluri-cellular, where gene activity from one cell influences the gene activity into another cell, as occurs in a tissue. These interactions also could span the cell life cycle, relaying their influence to the offspring cells. Moreover, this influence could be direct (by e.g. through miRNA), or indirect upon the activity of the proteic sub-product of genes as TF or protein kinases. However, more often the activity of one gene influences the activity of another as means of the activity of a third gene. The reason for this behavior is that cellular events occur sequentially, creating orchestrated cascades of gene activation as occurs in differentiation processes. Additionally, activation of genes does not occur only sequentially but also in parallel, meaning that often one gene influences more than another one. Often some genes behave as hubs, where one gene influences several target genes. The last part of this entangling process is the control of it, where the activity of sub sequenced activated genes, in turn feedback influencing the activity of the firstly activated genes. When this feedback process is positive, an amplification occurs and is often used by the cellular system as a bi-stability control switch (Smolen and Baxter, 2000). When the feedback is negative, a dampening occurs and is often used by the cells for oscillatory or periodic behaviors as cell cycle or circadian rhythms (Smolen and Baxter, 2000). In this way a network of precise regulations emerges and is called a gene regulatory network (GRN).

In the systems biology community, the set of interactions (wirings and rules) governing the behavior of a GRN is known as network *topology*. The task of finding this network topology from the data of the dynamics of a given GRN is the goal of reverse engineering. Usually, in order to perform the reverse engineering (RE, from here on) task a formalism to represent the original system dynamics is needed, and is called a *model*. The *modeling* of a given dynamic cellular system is very often not clearly distinguished from the task of reverse engineering from that system. A *model* is a representation of a system that helps us to understand its complexity.

If the model is validated, by e.g. a cross validation technique, then it could be used to make *predictions* about the original system. This last is known as the *predictive power (van Someren, et al., 2002)* of a model. However, to model a system one needs to know the wiring and rules of it topology. As we usually do not have complete information about this topology and if there is information about the dynamic of that system one can use the model to infer the missing topology. If the model is a good representation of the original system, it has more probabilities to make the correct inferences. This capability is known as the *inference power (van Someren, et al., 2002)* of the model.

To perform the reverse engineering task based on dynamical data, series of gene expression data over time (microarrays or RT-PCR) are used to represent a GRN dynamics, as could be the entire genome of a cellular line. In this context every gene from the GRN is known as a *node* (n) from a *network*, which is the GRN. The measurement of how many elements are wired to a particular node is known as the *connectivity (K)* of the node. The level of transcription for a given gene is known as its *activity state*. Here the set of all (*N) activity states* measured at a given time point, represents one transcriptional *state of the system*. In turn, a set of (*system states*) measured time points of microarrays represents a *trajectory* of the cellular system.
Some works started by Kauffman (Iguchi, et al., 2007; Kauffman, 1969; Kauffman, 2004; Kauffman, et al., 2004; Kauffman, 1969; Socolar, 2003) on the late 60´s proposed to see periodic behaviors as the cell cycle from a given cell line, as an *attractor* of the entire organism's genome. A phenomenological description of an attractor could be the *set of systems states where a system tends to exist*. That is, independently from the initial state a system have, it will *tend* to move into to the closest attractor. This implies that, a system could have more than one attractor. The vicinity from these attractors, where they applied their influence (like the size of a funnel), is known as the *basin of attractor*. In this way, a multicelullar organism is the entire system, and it different cell lines represent different attractors[1] from that organism.

[1] There are different kinds of attractors like; the fixed-point attractor (the simplest) limit cycles, toroids and strange attractors. Here, the analogy is between the cell cycle and the limit cycles attractors.

When a system is observed evolving from an initial given state until it reaches an attractor, it is said that the *trajectory* states are *transition states*. In this perspective, the last goal of the RE of GRN works is, to achieve the necessary knowledge to perturb a given cellular system to move it apart from its original attractor to another desired one. For instance, the goal of RE could be to selectively control cells belonging to a cancer attractor to differentiate them into a desired attractor, as could be the apoptosis (programmed cell death).

2.3 Dimensionality reduction by data selection

When using microarray time series the system is highly undetermined. There is a large lack of data in different dimensions, like observational time window, granularity of measurements, diversity of conditions (stimulus response curves for different stimuli and or conditions), repetitions, etc. Therefore, in order to reduce the dimensionality of the system to be analyzed several works circumscribe the system to be analyzed to the set of responding genes to a given stimuli. The criteria to select those responding genes have evolved over the last years, and since it is a crucial step for the reverse engineering of gene regulatory networks, some of them like those who have been tested in this work, will be discussed below.

Threshold data selection from complete data sets

A common practice among biologist to reduce the gene data to be analyzed is to take into account only those genes which expression has changed after a specific cellular stimulus by at least two fold expressions, or by deleting genes with incomplete data or those which standard deviation does not change beyond an arbitrary threshold. Besides the choosing of a certain threshold it always will be arbitrary and this approach faces two main drawbacks: (*i*) after correcting for multiple hypotheses testing, no individual gene may meet the threshold for statistical significance, because the relevant biological differences are modest relative to the noise inherent to the microarray technology.

(*ii*) Alternatively, one may be left with a long list of statistically significant genes without any unifying biological theme. Interpretation can be daunting and ad hoc, being dependent on a biologist's area of expertise.

Functional modules

Due to the previous two problems on the threshold filtering of data, I will discuss here different approaches oriented to reduce the dimensionality of the data. Three of them are based on circumscribing the reverse engineering and modeling process to a smaller functional module or cellular function. In this way the first main task to perform those processes is to isolate the smaller number of state variables, able to describe the given cellular module without missing important information. Therefore, this module should be functionally self-sufficient and the available data reflect the orthogonality (Lipan, 2005) from this module.

GSEA algorithm

The gene set enrichment (GSEA) algorithm is proposed (Subramanian, et al., 2005) to be the solution of the complex task of isolating the important set of genes related to a particular stimulus-response study. The logic of it is that single-gene analysis may miss important effects on pathways, because cellular processes often affect sets of genes acting in concert. They point that an increase of 20% in all genes encoding members of a metabolic pathway may dramatically alter the flux through the pathway and may be more important than a 20-fold increase in a single gene.

Gene Set Enrichment Analysis (GSEA) evaluates microarray data at the level of gene sets. The gene sets are defined based on prior biological knowledge, e.g., published information about biochemical pathways or co-expression in previous experiments. The goal of GSEA is to determine whether members of a gene set tend to occur toward the top (or bottom) of a list, in which case the gene set is correlated with the phenotypic class distinction.

Experimental isolation of modules through periodic stimulus

An experimental approach to the problem of isolation of functional genetic modules has been proposed (Lipan, 2005) by using oscillatory inputs to analyze the response of a cellular system.

Lipan and Wong propose in their work that an oscillatory input has many advantages: (*i*) the measurements can be extended to encompass many periods so the signal-to-noise ratio can be dramatically improved; (*ii*) the measurement can start after transient effects subside, so that the data become easier to incorporate into a coherent physical model; and (*iii*) an oscillatory stimulus has more parameters (period, intensity, slopes of the increasing and decreasing regimes of the stimulus) than a step stimulus. As a consequence, the measured response will contain much more quantitative information.

The genes that interact with the driven gene will be modulated by the input frequency. The rest of the genes will have different expression profiles dictated by the internal parameters of the biological system. This point of view is supported by their findings. Lipan and Wong notice that the measured data can be expressed as a sum of exponentially decaying functions, e-t, if a step stimulus is used while for a periodic input the response contains only exponentials with imaginary argument, ei-t. Mathematically, the main difference between exponentials with real arguments, e-t, and those with imaginary arguments, ei-t, is that with the former one cannot form an orthogonal basis of functions, whereas such a basis can be formed with the latter. Therefore they propose that, in general, the response of the network to a step input will be a sum of components that are not orthogonal on each other. The time dependence of these non-orthogonal components can be more complex than an exponential function; they can contain polynomials in time or decaying oscillations, depending on the position in the complex plane of *eigenvalues* of a transfer matrix H. In contrast, the permanent response obtained from a periodic input is a sum of Fourier components that form an orthogonal set. Orthogonal components are much easier to separate than non-orthogonal ones. This mathematical difference explains the advantage of using oscillatory inputs.

Clustering

Cluster analysis is an exploratory data analysis technique which aims at sorting different objects into groups in a way that the degree of association between two objects is maximal if they belong to the same group and minimal otherwise. Some works like Wahde et al (Wahde, 2000), have follow the logical step to capture the global behavior of the entire cellular system representing it by clusters instead of representing particular genes, reducing in this way the dimensionality of the system to be engineered.

However, in their work Wahde et al assumes that genes belonging to the same cluster share the same function. Therefore, this approach proposes to use the centroid (mean of the Euclidean distances between every gene expression profiles into a cluster) of every cluster to represent the function related to that group of genes. In this way the entire genome is represented by the biological functions that respond to the given stimuli.

However, this approach faces two issues. The first is related to the cluster interpretation. Genes belonging to a particular cluster could or not share the same function. More likely, genes belonging to a given cluster share the same regulation, but they could belong to different biological functions. The second issue of this approach is that at the end, it is not possible to map the centroid to any particular gene. Therefore the practical use of centroids to represent the entire genome is limited.

2.4 Theoretical works

2.4.1 Boolean Networks

Almost 40 years ago Kauffman (1969) developed this model to represent genetic networks. It is based on the assumption that genes exist basically in two possible states; active (ON) or inactive (OFF). The activity state of each gene at a given time is determined by a Boolean function (AND, OR, and NOT) of its inputs (wired nodes) at the previous time step.

It is assumed that each gene is controlled by K other genes in the network. For this models connectivity K is a very important parameter to determine the network dynamics (with large K, the dynamics tends to be more chaotic). In Random Boolean Network models, these K inputs, and a K-input Boolean function, are chosen at random for each gene. The Boolean variables (ON/OFF states of the genes) at time t+1 are determined by the state of the network at time t through the K inputs as well as the logical function assigned to each gene. An excellent tool for calculating and visualizing the dynamics of these networks is the DDLAB software (Wuensche, 1998). Under this approaches the total number of expression patterns is finite, therefore the system will eventually return to an expression pattern that it has visited earlier. Since the system is deterministic, it will keep following the exact same cycle of expression patterns. This periodic state cycle is the previously defined attractor of the network.

Another assumption performed by Kauffman, is that gene regulatory networks could have a structure similar to random Boolean networks (RBN, from here on) and therefore they should share some common features. For instance, the number of distinct attractors of a Random Boolean Network tends to grow as a square root of the number of nodes-genes (Kauffman, et al., 2004). If we equate the attractors of the network with individual cell types, as Kauffman suggests, it is explained why a large genome of a few billion base pairs is capable of a few hundred stable cell types.

This convergent behavior implies immense complexity reduction, convergence and

stabilization in networks of constrained architecture. In this way, with this model it is possible to correlate the size and number of attractors from a RBN to the number of cellular lines of an organism and the size of its genome in a predictive fashion. These correlations are possible for a variety of species, according to some scaling coefficients encountered by this work. Another insight into the behavior of large regulatory networks is that given analogous Boolean functions (Kauffman, 1971) and similar connectivity, RBN as well as biological systems tend to exhibit either a maximum or a minimum of organization.

As recently the microarray technology comes to generate genome size data, this model has been re-utilized to analyze dynamical properties of large GRN´s.
Some of these works utilized Boolean Networks (BN) to infer GRN from real data. Liang and Somogyi (Liang, et al., 1998) have developed the REVEAL algorithm, that utilized the Shannon entropy to correlate state transitions between nodes to infer the mutual information between them, in this way and using a full search approach they could the define the transitions rule table that reconstruct the original network topology. The main drawbacks from this algorithm are the general criticism to BN of representing gene expression by only two states: ON or OFF, and the assumed low connectivity *K*.

Akutsu (Akutsu, et al., 2000) developed a series of different algorithms based on BN to analyze the sample complexity of several variants networks, including noisy Boolean networks. He proved that using a conceptual simpler approach, $O(\log_2 N)$ random measurements are sufficient to identify a network of N genes with bounded connectivity K. This means that for a data set with 1000 genes and a connectivity K=2, in the order of only 10 independent measurements were sufficient for his algorithm to infer the network topology. However, this approach has some strong drawbacks. The first drawback is that its exhaustive search engine would utilize $O(10^{10})$ units of time. The second is that usually the connectivity is larger for real GRN. And the last drawback is a general one for any Boolean network model: genes expression exist in more than two ON/OF states, and very often the information processing of this GRN are related to continuous levels of expression of their genes.

2.4.2 Differential equation systems

To overcome some of the limitations of the BN, some other works have been developed to model the GRN on a continuous gene expression basis, using ordinary differential equations (ODE). A set of ODEs, one for each gene, describes gene regulation as a function of other genes:

$$dx_i/dt = f_i\left(x_1,...,x_N,u,\theta_i\right) \qquad \textbf{2. 1}$$

where $x_i(t)$ is the concentration of transcript i measured at time t, θ_i is a vector of parameters describing interactions among genes (the edges of the graph), i =1 … N, N is the number of genes and u is an external perturbation to the system.
As ODEs are deterministic, the interactions among genes represent causal interactions, and not statistical dependencies as in other methods. To reverse-engineer a network using ODEs means to choose a functional form for f_i and then to estimate the unknown parameters θ_i for each i from the gene expression data D using some optimization technique.

With an ODE-based approach signed directed graphs are obtained and it can be applied to both steady-state and time-series expression data. Another advantage of using ODE approaches is that once the parameters θ_i for all i are known, equation (2.1) can be used to predict the behavior of the network under different conditions (i.e. gene knockout, treatment with an external agent, etc.) as mentioned before as prediction power.

There are many different approaches that can be enclosed into this differential equation approaches, I will just briefly describe the major categories into which particular models could be assigned and mention some examples: generalized additive models, recurrent neural networks, S-systems, pair-wise equations.

Historically, systems of differential equations have long ago proved their validity in modeling simple gene regulation systems. An example is the work of Mjolsness *et al.* (1991), which used a hybrid homogeneous and 2D spatial reaction diffusion approach

to model a small number of genes involved in pattern formation during the blastoderm stage of development in Drosophila (Reinitz, 1995). In this work, the change in expression levels at each time point depended on a weighted sum of inputs from other genes, and diffusion from neighboring "cells". Synchronized cell divisions along a longitudinal axis (under the control of a maternal clock) were alternated by updating the gene expression levels. This model was able to successfully reproduce the pattern of eve stripes in Drosophila, as well as some mutant patterns on which the model was not explicitly trained.

After this model was introduced, several other models that use a similar formalism to the so-called connectionist model (Mjolsness *et al.*, 1991) were developed. Some examples are the linear model (D'Haeseleer, et al., 1999), linear transcription model (Chen, et al., 1999), weight matrix model (Weaver, et al., 1999) etc. Therefore, it has being proposed (D´Haeseleer, 2000)to unify all of them by their common additive nature and classified them into the so-called generalized additive models. Here I will briefly describe some of them.

Linear system

In last term, it is possible to interpret these models as a multiple regression process:

$$dx_i/dt = \sum_j w_{ji} x_j(t) + b_i \qquad \textbf{2. 2}$$

where x_i is the level of expression of gene i at time t, b_i is a bias term indicating the basal expression of gene i when the summation of regulatory inputs is zero, and weight w_{ji} indicates the strength of the influence of gene j on the regulation of gene i. Therefore, given an equidistant time series of expression levels (or an equidistant interpolation of a non-equidistant time series), it is possible to use linear algebra to find the least-squares fit to the data.

Chen *et al.* (1999) presented a number of linear differential equation models, which included both mRNA, and protein levels. They showed how such models can be solved using linear algebra and Fourier transforms. Interestingly, they find that

mRNA concentrations alone are not sufficient to solve their model, without at least the initial protein levels. Conversely, their model can be solved given only a time series of protein concentrations.

D'Haeseleer (D'Haeseleer, et al., 2000) showed that even a simple linear model can be used to infer biologically relevant regulatory relationships from real data sets. He applied a linear model to the central nervous system differentiation rat data. It is one of the few works developed over RT-PCR data, which are considerably more quantitative that microarray data. In his work, he merged two data sets to obtain a new data set with 65 genes and 28 time points. However, the dimensionality problem still appears with several models able to fit equally well the data. To decrease this dimensionality problem he used a spline interpolation scheme to obtain more data points.

Van Someren *et al.(van Someren, et al., 2000)* combined a linear model with clustering techniques to purpose a solution to the dimensionality problem. They applied their model to the Yeast cell cycle data from Spellman.(Spellman, et al., 1998) and showed that by working with the centroids of the clustered data, it was possible to reconstruct a global network of the yeast cell cycle. However, as previously mentioned, this approach has the drawback of not being able to correlate a particular gene with a given cellular function.

Recurrent neural networks

As has been said, one of the pioneer's works on the area using differential equations is the one from Mjolsness et al (1991). However, this work as a series of other developments could be seen as recurrent neural networks. Some advantages of seeing these works under this perspective, is that the artificial intelligence community has developed a series of good optimization methods for this kind of recurrent neural networks, as back propagation through time and global optimization with genetic algorithms (GA).

On the other hand, some works about continuous time recurrent neural networks (CTRNN from here on) have proved that given enough data, there is just one model that better fits the data. This uniqueness *(Albertini and Sontag, 1993)* property is

highly desired and in the RE of GRN has been called *consistency*.

Weaver *et al.* (1999) showed how a non-linear transfer function can be incorporated into a linear model, and demonstrated that some randomly generated networks can be accurately reconstructed using this modeling technique. To handle the dimensionality problem, Weaver proposed the use of the Moore-Penrose pseudo-inverse.

This special matrix inverse produces a solution for undetermined problems that minimizes the sum of the square weights but still perfectly fits the data. To impose a limited connectivity, he proposed a greedy backward search that iteratively sets the smallest weight to zero and then recomputed the pseudo-inverse on the, now slightly less undetermined problem. However, this last technique is extremely sensitive to noisy data.

Whade and Hertz (Wahde, 2000), inspired by the work from Mjolsness, utilized a continuous time recurrent neural network model in the form:

$$dY_i/dt = \left(\sigma \sum_j W_{ij} Y_j - Y_i + \theta_i \right) \tau^{-1}, i = 1,...N, \qquad \textbf{2. 3}$$

to represent artificial as well as clustered GRN, where τ^{-1} are rate constants, Y_i the gene expression levels, dY_i/dt the genes expression levels derivatives in time, θ_i the genes basal expression levels, N the number of genes modeled, giving W_{ij} as an NxN weighting matrix and σ is the logistic sigmoid transfer function: $\sigma(x) = \left(1 + e^{-x}\right)^{-1}$.

Wahde and Hertz utilized a Genetic Algorithm as global optimization technique of parameters and represent the same CNS rat data from Wen, by only four clusters. Additionally, using artificial data they showed that it is better for the RE task, to have multiple shorter time series than one long series.
Again, the major drawback of this work is that just four cellular functions without any specific gene where taken into account.

Recently, Hu (Hu, 2005) introduced a time delay term to the Wahde formulation in order to account for additional process, as translation or diffusion that could delay the response from a gene activity and its influence upon it target genes. They utilized the Back propagation through time method to globally optimize their parameters. In they work they reproduce the kinetics of six genes of the SOS DNA repair system of *E.coli*. taking 50 time points data. In this way their model could recover seven of nine experimentally reported regulations, as well as suggest six additional ones to be tested. A drawback from this approach is the interpretation of the RNA decay, where it is misunderstood and interpreted as the τ^{-1} in a similar formalism as the equation 2.3.

S-system

S–systems (synergistic and saturable system) have long been used (Savageau, 1969) as models of biochemical pathways, genetic networks and immune networks (Akutsu, et al., 2000; Akutsu, et al., 2000). S–Systems are a class of non–linear ordinary differential equations and have the form:

$$dX_i/dt = \alpha_i \prod_{j=1}^{n} X_j^{g_{ij}} - \beta \prod_{j=1}^{n} X_j^{h_{ij}} \qquad \textbf{2. 4}$$

where n is the number of state variables or reactants X_i (X expressed in concentration), and i,j ($1 \leq i,j \leq n$) are suffixes of state variables.

The terms g_{ij} and h_{ij} are the interactive effect of X_j to X_j. The first and second terms represent all influences that increase and decrease X_i, respectively. The constants, gij, and h_{ij} are exponential parameters referred to as kinetic orders. S–Systems have unique mathematical properties allowing large realistic phenomena to be investigated and can be derived from general mass balance equations by aggregating inputs and outputs approximated by the products of power–law functions.

Each dimension of the S–System model represents the dynamics of a single variable represented as the difference of two products of power–law functions, one describing the influxes and the other describing the effluxes. The major disadvantage of the S-system is the large number of parameters to be estimated: 2X(X+1).

Pair-wise equations

Another way to overcome the so-called dimensionality problem is to restrict the complexity of the model by only considering pair-wise relationships. Apparently, Arkin (McAdams and Arkin, 1997) was the first to suggest the use of time-shifted pair-wise correlations to model biochemical pathways. Initially, the position and magnitude at which the maximal time-shifted cross-correlation occurs is computed in a pair-wise fashion. Then a distance measure is constructed to perform hierarchical clustering obtaining a linked tree of associated genes. Finally the model is completed with information about directionality and time lags, and in turn it can provide information about the dynamics of the system.

A further development comes with the work from Chen, who utilize a similar approach but instead of using the correlation he performed the matching of peaks in the expression profiles of genes. His algorithm performed threshold filtering followed by the clustering step, obtaining profiles of sets of expression peaks. Then peaks in the profiles are compared in a pair-wise fashion to determine causal activation scores. From these scores a putative regulation network is constructed by optimizing it with a simulating annealing approach.

Another related approach which combines the logical rules of Boolean network models with some of the advantages of differential equation methods are “Glass networks”. Glass networks have been proposed as a simplified model of genetic networks (Edwards and Glass, 2000) as well as an underlying model for the reverse–engineering of regulatory networks (Perkins, et al., 2006). The main drawback of these models is the low connectivity K they are limited to, basically to single and in some advanced cases to two pair interactions.

As a general criticism could be said that, differential equations presuppose that concentrations of chemical species changes continuously and deterministically, both of which assumptions may be questionable in the case of gene regulation (Gibson and Mjolsness, 2001; Gillespie, 1977; McAdams and Arkin, 1999; Szallasi, 1999). Against the first assumption is the fact that some of the components of GRN acts in small numbers of molecules, as the transcription factors in the cell nucleus and a

single DNA molecule carrying the gene, compromising the continuity assumption. Second, deterministic change presupposed by the use of the differential operator d= dy/dt may be questionable due to fluctuations in the timing of cellular events, such as the delay between start and finish of transcription. As a consequence, two regulatory systems having the same initial conditions may enter into different states.

2.4.3 Stochastic Models

Stochastic models of gene regulatory processes claim to remedy many of the drawbacks of deterministic (mainly differential equation) based approaches. One such shortcoming is the assumption of a continuous rate of mRNA production. Typically, transcription factors exist on very low concentrations in a cellular system, and this is not well represented by continuous models as differential equations. In fact, as exposed at the introduction, mRNA as well as proteins are not produced at a continuous rate, but rather in short bursts (McAdams and Arkin, 1997). In addition, some mechanisms of transcriptional regulation are known to amplify noise, creating heterogeneity within a population. With the addition of noise in gene transcription, individual cells may take different regulatory paths despite having the same regulatory input (Guet, et al., 2002).

It is very likely that evolution has selected networks which can produce deterministic behaviors from stochastic inputs in a noisy environment. In fact, certain topologies in networks can attenuate the effects of noise (such as the mentioned control loops) (Rao, et al., 2007) and also that noise can indeed act as a stabilizer itself in other systems (Hasty, et al., 2000).

There are generally two methods for modeling stochastic gene regulation. The first are stochastic differential equations:

$$dY_i/dt = f_i(Y_i) + \nu_i(t) \qquad \textbf{2. 5}$$

This previous equation gives the form of a stochastic differential equation that explicitly models noise in the system through the term $\nu(t)$. This equation is often referred as the Langevin equation and in general is not analytical tractable. Typically, solutions to the Langevin equations are obtained through the use of Monte–Carlo algorithms. The conditions under which the approximation is valid may not always be possible to satisfy in the case of genetic regulatory systems (de Jong, 2002).

The second approach is to characterize the transitions of a molecule using probability functions. During each individual time step, a molecule is given a certain probability of transitioning to a different discrete state. From this, a probability density function for the behavior of the system can be obtained. Such systems are referred to as the "Master Equation". It has being proposed disregard the so-called master equation altogether and directly simulate the time evolution of the regulatory system. This idea underlies the stochastic simulation approach developed by Gillespie (Gillespie, 1977; Gillespie, 1992).

Although stochastic models are often more realistic than their deterministic counterparts, they are expensive to simulate. In fact, for many realistically sized systems, stochastic approaches are impractical (Swain, et al., 2005). However, stochastic models of gene regulation have been successfully used in Keasling (Keasling, et al., 1995), Arkin (Arkin, et al., 1998) and Kastner (Kastner, et al., 2002) just to mention a few examples. Recently, significant efforts have being performed to reduce the computer simulation cost for the Gillespie algorithm (Cao and Gillespie, 2006; Slepoy, et al., 2008). However, their use for the RE of GRN area has not being assessed.

2.4.4 Bayesian networks

Bayesian networks are probabilistic models. They model the conditional independence structure between genes in the network. Edges in a Bayesian network correspond to probabilistic dependence relations between nodes, described by conditional probability distributions. Distributions used can be discrete or continuous, and Bayesian networks can be used to compute likely successor states for a given

system in a known state.

A formal definition of Bayesian networks is:

A Bayesian Network is a directed, acyclic graph G = (X,A), together with a set of local probability distributions P. The vertices X = {X1, ...,Xn} correspond to variables, and the directed edges A represent probabilistic dependence relations between the variables. If there is an arc from variable Xi to Xj , then Xj depends probabilistically on Xi. In this case, Xi is called a parent of Xj. A node with no parents is unconditional. P contains the local probability distributions of each node Xi conditioned on its parents, p(Xi|parents(Xi) (Radde and Kaderali, 2007)).

In the formalism of *Bayesian networks (Friedman, et al., 2000)*, the structure of a genetic regulatory system is modeled by a directed acyclic graph G = (V; E). The vertices i ∈ V, 1 < i < n, represent genes or other elements and correspond to random variables Xi. If i is a gene, then Xi will describe the expression level of i. For each Xi, a conditional distribution p(Xi | parents(Xi)) is defined, where parents(Xi) denotes the variables corresponding to the direct regulators of i in G.

In this approach the *conditional independency* i(Xi; ***Y*** | ***Z***) express the fact that Xi is independent of ***Y*** given ***Z***, where ***Y*** and ***Z*** denote sets of variables. The graph encodes the *Markov assumption*, stating that for every gene *i* in G, i(Xi; nondescendants (Xi) | parents.Xi)). By means of the Markov assumption, the joint probability distribution can be decomposed into:

$$p(X) = \prod_{i=1}^{n} p\left(X_i \mid parents\left(X_i\right)\right) \qquad \textbf{2. 6}$$

The resulting graphs from this Bayesian networks implies additional conditional independencies. Two graphs, and hence two Bayesian networks, are said to be equivalent, if they imply the same set of independencies. The graphs in an equivalence class cannot be distinguished by observation on ***X***. Equivalent graphs can be formally characterized as having the same underlying undirected graph, but may disagree on the direction of some of the edges see Friedman (Friedman, et al., 2000) for details and references.

Given a set of expression data *D* in the form of a set of independent values for ***X***, learning techniques for Bayesian networks allowed to some works to infer the network, or rather the equivalence class of networks that best matches *D*.

These learning techniques rely on a matching score to evaluate the networks with respect to the data and search for the network with the optimal score. As this optimization problem is known to be NP-hard, heuristic search methods have to be used, which are not guaranteed to lead to a globally optimal solution. However, an additional problem is that currently available expression data underdetermines the network, because just a few dozen of experiments provide information on the transcription level of thousands of genes.

Friedman and colleagues (Friedman *et al*., 2000) proposed an heuristic algorithm for the inference of Bayesian networks from expression data that is able to deal with this so-called dimensionality problem. Instead of looking for a single network, or a single equivalence class of networks, they focus on features that are common to high-scoring networks. In particular, they look at Markov relations and order relations between pairs of variables Xi and Xj. A Markov relation exists, if Xi is part of the minimal set of variables that shields Xj from the rest of the variables, while an order relation exists, if Xi is a parent of Xj in all of the graphs in an equivalence class. An order relation between two variables may point at a causal relationship between the corresponding genes. Statistical criteria to assess the confidence in the features have been developed. A recent extension of the method (Pe'er, et al., 2001) is able to deal with genetic mutations and considers additional features, like activation, inhibition, and mediation relations between variables.

Markov relations and order relations have been studied in an application of the algorithm to the cell cycle data set of Spellman and colleagues (Spellman, et al., 1998) (see Pe'er *et al*. [2001] for another application). This data set contains 76 measurements of the mRNA expression level of 6,177 *S. cerevisiae* ORFs included in time-series obtained under different cell cycle synchronization methods. The Bayesian induction algorithm has been applied to the 800 genes whose expression level varied over the cell cycle. By inspecting the high-confidence order relations in

the data, Friedman and colleagues found that only a few genes dominated the order, which indicates that they are potential regulators of the cell cycle process.

Many of these genes are known to be involved in cell-cycle control and initiation. Of the high-coné dense Markov relations, most pairs are functionally related. Some of these relations were not revealed by the cluster analysis of Spellman and colleagues.
A Bayesian network approach towards modeling regulatory networks is attractive because of its solid basis in statistics, which enables it to deal with the stochastic aspects of gene expression and noisy measurements in a natural way. Moreover, Bayesian networks can be used when only incomplete knowledge about the system is available. Although Bayesian networks and the graph models are intuitive representations of genetic regulatory networks, their disadvantage is to leave the dynamical aspects of gene regulation implicit. To some extent, this can be overcome through generalizations like *dynamical Bayesian networks*, which allow feedback relations between genes to be modeled (Murphy, 1999).

Since this Dynamic Bayesian networks (DBN) are among of the more promising works on the RE of GRN area, in this work I compare the performance of the here introduced TDRNN model with the performance of DBN and the previously explained CTRNN.

3. Methods

3.1 Workflow

In few steps the working scheme could be described as: a) Times series data acquisition. b) Data quality control and normalization. c) Data selection. To reduce the number of state variables here I propose to focus on a small functional module or cellular function. d) Data interpolation. This has been applied already in order to reduce the solution space. e) Data fitting. The parameters of the model are globally optimized by the use of a genetic algorithm (GA) to approximate the dynamics of the selected module. f) Robust parameter identification. Statistical analysis is performed to define the more likely parameters and consequently network connectivity g) Summarization. The last step is the proposal of a network topology to describe the dynamics of the original functional module. h) Error calculation. In case of the test data, this error is calculated between the resultant network and the goal benchmark network.

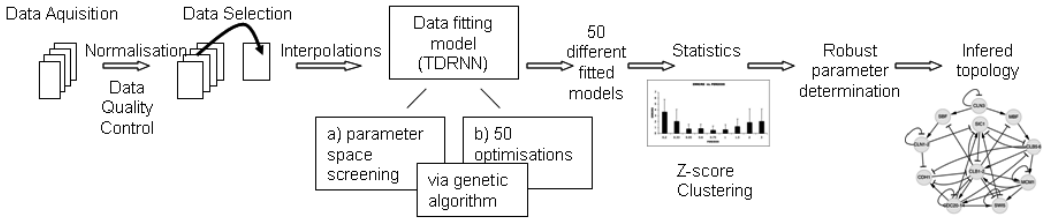


Figure 3.1 Workflow of the reverse engineering of gene regulatory networks. From left to right; times series of data acquisition, normalization and interpolation of data, 50 regression multiples to fit the data using a global parameter optimization approach, parameter significance identification, and finally summarization of results on a network topology graph

3.2 Data pre-processing, Quality control

Given to the many sources of error of the microarray technology, a prerequisite for working with experimental data from microarrays is to check their quality. This includes checking for experimental outliers, discarding them, and checking for statistical uncertainty of the results. In this thesis, quality control of every set of microarray data has been performed with the package: *simpleaffy (Wilson and Miller, 2005)* from Bioconductor (Gentleman, et al., 2004) which is able to detect different sources of errors during the different steps of the pipeline of the microarray data production. For every set of times series, four different sources of errors are checked; a) average background, b) 3-´5´ relationship and c) percentage of positive hybridizing d) scale factor.

a) Average background: should be similar across all chips. There are several reasons for a significant variation on the average background, but generally it is due to some experimental problems, like having different concentrations of mRNA on the hybridization cocktails, or a more efficient hybridization in some of the chips respect to the others.

b) 3'—5' relationship or early degradation of the mRNA: detected by an abnormal signal from some control probe sets present on the chips. The chips contain some specific genes probes, which have a particularly large sequence and well-defined degradation kinetics. A change on this kinetic indicates that an abnormal degradation of the sample has occurs and that measurements on the rest of the probe sets could be lower than should be.

Most cell types ubiquitously express β-actin and GAPDH. These are relatively long genes, and the majority of Affymetrix chips contain separate probesets targeting the 5', mid and 3' regions of their transcripts. By comparing the amount of signal from the 3' probeset to either the mid or 5' probesets, it is possible to obtain a measure of the quality of the RNA hybridized to the chip.
If the ratios are high then this indicates the presence of truncated transcripts. This may occur if the in vitro transcription step has not performed well or if there is general

degradation of the RNA. Hence, the ratio of the 3' and 5' signal gives a measure of RNA quality.

c) Percentage of positive hybridization: an indication of how many probes present on the chip exhibit positive hybridizing. For a given experiment, one should expect that the global number of positive hybridization is similar. However, this is not necessarily the case because usually one is interested in finding the differences between experiments. Therefore, problems with this evaluation should be taken carefully.

These percentages of responding genes (Present/Marginal/Absent calls) are generated by looking at the difference between perfect matches PM and mismatches MM values for each probe pair in a probeset. Probesets are flagged Marginal or absent when the PM values for that probeset are not considered to be significantly above the MM probes. As with scale factors, large differences between the numbers of genes called present on different arrays can occur when varying amounts of labeled RNA have been successfully hybridized to the chips. This can occur for similar reasons (differences in array processing pipelines, variations in the amount of starting material, etc.). The '% Present' call simply represents the percentage of probesets called Present on an array. As with Scale Factors, significant variations in % Present call across the arrays in a study should be treated with caution. Note that usually the absolute value is generally not a good metric because some cells naturally express more genes than others

d) Scale factor: some normalization packages (e.g. MAS 5.0) adjust the mean value of expression between different chips to the same value. If scale factors between arrays are large, it is an indication of issues when trying to compare between chips.

The default normalization used by MAS 5.0 (and many other algorithms) makes the assumption that gene expression does not change significantly for the vast majority of transcripts in an experiment.

(Note that this assumption is also explicit in any analysis that looks for a relatively small number of changing genes within a transcript population containing many

thousands (for example, looking for ~200 differentially expressed probesets from the ~54,000 found on the U133 plus 2 array).

One consequence of this is that the trimmed mean intensity for each array should be constant, and by default, MAS 5.0 scales the intensity for every sample so that each array has the same mean. The amount of scaling applied is represented by the 'scale factor', which, therefore, provides a measure of the overall expression level for an array, and (assuming all else remains constant), a reflection of how much labelled RNA is hybridized to the chip. Large variations in scale factors signal cases where the normalization assumptions are likely to fail due to issues with sample quality or amount of starting material. Alternatively, they might occur if there have been significant issues with RNA extraction, labeling, scanning or array manufacture. In order to successfully compare data produced using different chips, Affymetrix recommend that their scale factors should be within 3-fold of one another.

3.3 Data normalization

The goal of normalization is to be able to compare between different microarrays chips. In general this is achieved by adjusting an average value of every experimental array equal to that of the baseline array, in the case of time series data, the reference array is that array (or those, in case of repetitions) without the external stimuli.

VSN data normalization

In the case of the microarrays chips (Affymetrix) used, their preprocessing involves the following steps: a) combining the perfect match (PM) and mismatch (MM) intensities in one number per probe, b) calibrating, c) transforming, and d) summarizing the data. The algorithm *vsn* (Huber, et al., 2002), from the Bioconductor platform, addresses the calibration and transformation steps.

This algorithm is considered to be the first choice for these tasks. The goal of this algorithm is to provide robustness and avoid overfitting by first calibrating and then performing a transformation of the data.

Calibration is performed as follows:

Let y_{ki} be the matrix of uncalibrated data, with k indexing the rows and i the columns, then the calibrated data y'_{ki} is obtained by scaling with a factor λ_{si} and shifting the (Draper and Smith, 1998) data by the factor $O_{si,}$:

$$y'_{ki} = \frac{y_{ki} - o_{si}}{\lambda_{si}} \qquad \textbf{3. 1}$$

s is the so-called stratum (a classification of regions of the chips according to their background signal) to probe k.

The transformation to a scale where the variance of the data is approximately independent of the mean is performed by the use of the function:

$$h_{ki} = \arcsin h(a_0 + b_0 y'_{ki}) = \log\left(a_0 + b_0 y'_{ki} + \sqrt{(a_0 + b_0 y'_{ki})^2 + 1}\right) \qquad \textbf{3. 2}$$

Where *a* and *b* are constants of proportionality calculated at the beginning of the algorithm (here not shown). Both are applied simultaneously to the data in order to obtain an almost constant transformed variance for every spot, independently from the transformed mean for that spot. In this way, it is possible to work simultaneously with high and low expression values, while at the same time avoiding any bias.

3.4 Dimensionality problem. The use of interpolation approaches

As it has been explained in the section 2.3 the system is highly undetermined due to the lack of data in different dimensions, like time window, granularity, diversity of conditions (stimulus response curves for different stimuli and or conditions), repetitions, etc. Therefore, in order to increase the amount of data, here, it is assumed that changes in gene expression between one measurement and the next one follow a smooth function. Hence, interpolations in time are performed for every gene to obtain a continuous set of data and to impose some constrains on the system.

Linear interpolation

Interpolation is a process for estimating values that lie between known data points. The simplest way to perform interpolation consists in the use of a linear function to produce continuous data points along the gap between two points using the shortest trajectory. This kind of interpolation is named linear interpolation and supposes a normal distribution of the measured data in relationship to the unknown real data.

If one have repetitions of experimental data measurements then it is possible to estimate the dispersion of the data in the forms of variance or standard deviation of gene expression. However, this variability would be the sum of two different processes, the biological variability and the error involved in the measurement process. The former fluctuations are usually hard to estimate, but the available information points to a normal distribution of this kind of variability. In relationship to the error associated to the experimental measurements, the supposition is that there is not a systematic error associated, but in case, the Quality Control procedure previously described should help to detect this problem.

Cubic spline interpolation

Another possibility used in this work to increase the data is the cubic spline interpolation. This algorithm also guarantees continuity of the interpolated data. Therefore, the generated data could be differentiable everywhere and this characteristic is often used by optimization methods like the gradient descendent in other contexts. However, in any case interpolation with cubic splines implies the smoothing of the original data.

There are two assumptions that underlie smoothing; a) the relationship between the original data and the predicted data is smooth. b) The smoothing process results in a smoothed value which is a better estimate of the original value because the noise has been reduced. However, one should not fit data with a parametric model after smoothing, because the act of smoothing invalidates the assumption that errors are normally distributed (Draper and Smith, 1998). Nevertheless, I also tested and compare the performance of this interpolation technique.

The Ziv Bar-Joseph et al. algorithm for gene expression representation

It occurs very often that experimental data is incomplete, very noisy and not uniformly sampled. To overcome the missing data point problem, this algorithm (Bar-Joseph, 2004) splits the entire data set into clusters of genes showing similar behavior and calculates the intra cluster noise. With this information, the algorithm calculates the most likely values for the missing data. In this way, they provide with entire vectors of data points of gene expression to a B-spline smoothing algorithm that also takes into account the noise on the data obtaining a better representation of gene expression behavior over time. This algorithm was tested while using experimental data.

3.5 Data fitting

As explained in the workflow section, the next step in reverse engineering basically consists in performing multiple regressions to find the set of parameters of a given model that best fits the available data. This last presupposes the existence of an already chosen model to perform the regressions. Since the system is an *ill posed* problem (Radde and Kaderali, 2007), due to the so-called dimensionality problem, there are several models that could fit the data and one have to distinguish between them (Often named system identification or inverse problem).

Here, two different groups of related models were tested: the CTRNN and the new specific TDRNN here introduced. The data to be fitted is the data obtained from interpolation of the time series gene expression as explained in the previous section. The fitting is achieved mapping the behavior of every node from these models to a specific gene from the original data set. Finally, since for every model there are several sets of parameters that equally well fit the data, the next step is to discriminate between those sets of parameters by performing statistical analysis to conclude with a consistent solution to the reverse engineering problem; this will be covered in sections 3.6.3 and 4.14.

Error measuring, mean square error.

The fitting of the data by the models is measured by the use of the mean square error between the output from every node and the data over time:

$$MSE = \sum_{0}^{L}\sum_{i=0}^{N}\left(d_i - o_i\right)^2 \Big/ L \bullet N \qquad \textbf{3. 3}$$

Where $(d_i - o_i)$ is the error between the desired output and the obtained output from a node at a particular time, N represents the total number of nodes studied and L the simulation measured duration.

Parameter optimization

This work uses a canonical genetic algorithm (Whitley, 1993) to globally optimize the parameters of the models using a forward Euler[2] integration scheme to simulate the original system.

Genetic algorithms

A genetic algorithm (GA) is basically a search technique inspired in the nature of evolution. It is often used to find an exact or approximate solution to optimization or searching problems. Genetic algorithms are categorized as global optimization heuristic searching technique.

A typical genetic algorithm requires two things to be defined:

1. a genetic representation of the solution domain (Blanco and Delgado, 2001),
2. a fitness function to evaluate the solution domain.

The fitness function is defined over the genetic representation and measures the quality of the represented solution. The fitness function is always problem dependent. Once I have defined the genetic representation and the fitness function, GA proceeds to initialize a population of solutions randomly, then I applied repetitive adjustments to the mutation, crossover, inversion and selection operators as described in table 3.1 until achieving a general increasing of the fitting.

Initially many individual solutions are randomly generated to form an *initial population*. The population size depends on the nature of the problem, but typically contains several hundreds or thousands of possible solutions. Traditionally, the population is generated randomly, covering the entire range of possible solutions, the so called search space. Occasionally, the solutions may be "seeded" in areas where optimal solutions are likely to be found.

[2] In general is a bad idea to perform serious calculations with Euler integration, here the caution taken is the general recommendation in the computer science area to choose an integration step at least ten times smaller than the smallest time constant.

The next step consists of the evaluation of every individual from the initial population according to the selected fitness function. Analogous to an evolution process, this initial population is the first generation that will evolve through generations towards a better solution to the problem in question. During each successive generation, a proportion of the existing population is selected to breed a new generation. This one is called the intermediary population. To generate the intermediary population, individual solutions from the actual generation are selected through a fitness-based process, where fitter solutions are preferentially selected. Usually the selection methods (threshold ranked selection, elitist strategy) rank the fitness of each solution and preferentially select the best solutions.

Other selection methods rate only a random sample of the population, this process may be very time-consuming and also uses a fitness-based function. Most fitness-based functions are stochastic and designed in a way so that a small proportion of less fit solutions are selected. This helps keeping the diversity of the population large, preventing premature convergence and therefore a poor final solution. Popular and well-studied selection methods include roulette wheel selection and tournament selection.

The next step is to generate a new-generation or population of solutions from those selected by the use of the genetic operators: crossing-over (also called recombination), and/or mutation.

Many crossing-over techniques exist for population's individuals which use different data structures to store themselves. In a single crossing over scheme, a crossover point on both parents' organism strings is selected. All data beyond that point in either organism string is swapped between the two parent organisms. The resulting organisms are the children.

The classic example of a mutation operator involves a probability that an arbitrary bit in a genetic sequence will be changed from its original state.
A common method of implementing the mutation operator involves generating a random variable for each bit in a sequence. This random variable tells whether or not a particular bit will be modified.

For each new solution to be produced, a pair of "parents" solutions is selected from the intermediary population to breed a new individual. By producing a "child" solution using the above mentioned methods of crossover and mutation, a new solution is created which typically shares many of the characteristics of its "parents". New parents are selected for each child, and the process continues until a new population of solutions of appropriate size is generated.
These processes ultimately result in the next generation population of chromosomes which is different from the initial generation. Generally, the average fitness of every new generation will increase since only the best organisms from the previous generation are selected for breeding, along with a small proportion of less fit solutions, for reasons already mentioned above.

This generational process is repeated until a termination condition has been reached. Common terminating conditions are

- A solution is found that satisfies minimum criteria
- Fixed number of generations reached
- The highest ranking solution's fitness is reaching or has reached a plateau such that successive iterations no longer produce better results
- Manual inspection
- Combinations of the above.

Here binary encoding was used with four Bytes per parameter. The population size was fixed at 1000 in all cases. Fitness rank-based selection was performed and the stopping criterion was in all cases the number of generations (N=1000). Data was pre-screened to conveniently adjust the mutation and crossing-over as shown in table 1 for the different data sets.

Table 3.1 Optimization parameters adjustment

	Repressilator			Yeast		
	Initial	Adjusted at gen:	Factor	Initial	Adjusted at gen:	Factor
Mut:	0.33	50, 150, 300, 750, 850	x 0.33	0.5	100, 250, 500, 750	x 0.2
C.O.	0.4		+ 0.8	0.5		+ 0.9

Mut = mutation rate C.O. = crossing over gen = generation

Fitness rank-based selection was performed and the stopping criterion was in all cases the number of generations.

Two optimization functions derived between the model outputs and the interpolated data along the optimized period of time were used by the GA as fitness functions. The first

$$Fitness = (MSE + 1)^{-1} \quad \textbf{3. 4}$$

optimization function was used to obtain a bounded fitness space, while the second function:

$$Fitness = MSE^{-1} + \lambda e^{-\beta MSE} \quad \textbf{3. 5}$$

additionally incorporates a fitness-adaptive weight pruning function (Bebis, 1996) with β controlling the onset of the pruning starting point. λ is a function of the gene interaction weights W_{ij} according to the following parabolic function:

$$\lambda = \sum_{i,j=1}^{N^2} \left(aW_{ij}^2 + bW_{ij} + c\right) \Big/ \left(dW_{ij} + e\right), \quad \textbf{3. 6}$$

Where N is the number of nodes and a, b, c, d and e (50, -170, 100, 5 and 1 respectively) are parameters controlling the shape of the parabolic function.

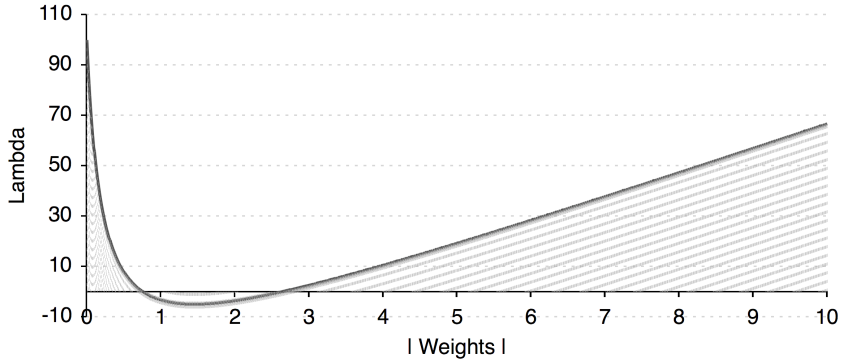


Figure 3. 2 Shape of the Lambda function that evaluates every weight to be pruned. Weights between 1 and 3 are penalized while zero and values higher than 3 are promoted

This function penalizes weights in the range [1,3] and promotes values close to 0 and bigger than 3 to favor a sparse interaction matrix without affecting the MSE-derived fitness. Changing the shape of the parabolic function or changing the range of permissible interaction weights W_{ij} did not change the results qualitatively.

Fifty optimization runs were performed for each of the synthetic and experimental time series, randomly initializing the model parameters and the GA. The parameters to be optimized were $[\tau, \vartheta, \theta, W]$ and for the TDRNN additionally the time delay $[\delta]$.

3.6 Models

3.6.1 The CTRNN model

In the continuous time recurrent neural network (CTRNN from here on) approach, it is assumed that gene activity is reflected at the level of mRNA expression while monitoring a cell stimulus-response or any other normal dynamical cellular processes. The observables changes on mRNA expression are the balances of all those processes described in the introduction, but projected at the mRNA expression level. The GRN power inference of this model relays on the analogy between the continuous activity level from it nodes and the continuously regulated gene expression of a given GRN.
For the internal structure of this model, - its nodes are predictors, from a multiple regression point of view - it could be classified into the generalized linear models (Bay, et al., 2002). This model could be derived from the gene activity analogy as follows:

For a given gene (Yi), changes in mRNA expression over time are its synthesis (S) and degradation (ϑ) balance:

$$dY/dt = S(Y_i) - \vartheta(Y_i) \qquad \textbf{3. 7}$$

Since very sparse information is available with respect to the mRNA degradation, this model assumes a first order degradation kinetics for the gene expressions:

$$\vartheta_i(Y_i) = K_i(Y_i) \qquad \textbf{3. 8}$$

In turn, it synthesis rate is the balance between direct or indirect interactions with other genes (Y_j) that activates or repress the gene in question:

$$S_i(Y_i) = \sum_{j=1}^{j=N} Y_j \qquad \textbf{3. 9}$$

An additional term, a constant bias (θ_j) is added to take into account a basal level of activity for any gene:

$$dY_i/dt = \sum_{j=1}^{j=N} (Y_j - \theta_j) - \vartheta_i(Y_i) \qquad \textbf{3. 10}$$

However, every interaction between two genes is a complex process following a non-linear behavior. In gene regulatory networks this means a sensitive switch like behavior as described by Hill's kinetics (Hofmeyr, 1997; Setty, et al., 2003; Zaslaver, et al., 2004). Therefore, on this family of generalized additive models, every node-to-node interaction is passed through a logistic sigmoid (σ) function:

$$\sigma(x) = \frac{1}{1+e^{-x}} \qquad \textbf{3. 11}$$

giving us:

$$dY_i/dt = \sum_{j=1}^{j=N} \sigma(Y_j - \theta_j) - \vartheta_i(Y_i) \qquad \textbf{3. 12}$$

The model here described represents the relative changes of gene expression rather than being a model of mRNA concentration kinetics. The reason is that as explained on the normalization section, the experimental data from the microarrays technology is far from being quantitative mRNA´s concentrations. Microarray time series data are rather strong indications of fold changes in relationship to a predefined state of activity, usually respect to time cero defined as the time before any stimuli is supplied to a cellular system.

The application of the sigmoid function to every interaction and not to the entire summation of them increases the precision of the calculation with respect to the Whade and D´Haesseleer CTRNN models. At the same time, other consequence from using the original Beer (Beer, 1995) CTRNN approach is, that a gene induction or repression change scales proportional to the number of genes:

Since:

$$\lim_{(Y_j-\theta_j)\to\infty} \sigma\left(Y_j-\theta_j\right)=1 \qquad \textbf{3. 13}$$

and assuming that all interactions on the summation of a given node are activations (positives) at their respective maximum value:

$$Y_i(\max)=\lim_{N\to\infty}\left\{\sum_{j=1}^{N}\sigma\left(Y_j-\theta_j\right)\right\}=N \qquad \textbf{3. 14}$$

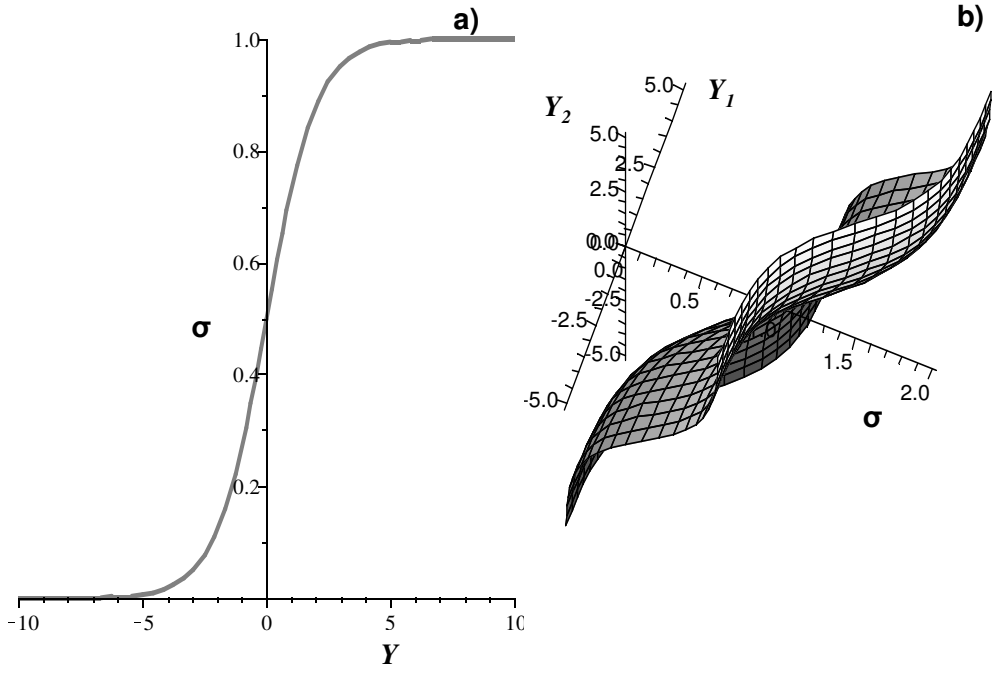


Figure 3.3 Sigmoid function comparison. Comparison among the output space after applying the sigmoid function to the summation of all interacting terms in a) and the output space after separated application of the sigmoid function to every interaction b). Here, for simplicity in b) is depicted only to the second term of N.

with N = number of total genes modeled by the CTRNN. Thus, the balance between genes activating and inactivating each other acquires more relevance. This kind of recurrent neural networks is full connected and explores every possible interaction between nodes, in order to fit the data there are two possible tendencies for the final topology: a) A mutual cancellation of interactions. b) the elimination of unnecessary interactions giving a sparsely connected network. To the actual knowledge, this last is the kind of network topologies present on the real genetic networks and has to be favored by any model. Therefore, the second optimization function previously described was used.

Additionally, a weighting (W) factor is applied to every interactions pair (ij), in order to accentuate the importance of that interaction. Additionally, this weight assigns a final sign to the interaction:

$$\frac{dY_i}{dt} = \sum_{j=1}^{N} W_{ij}\sigma\left(Y_j - \theta_j\right) - \vartheta_i Y_i \qquad \textbf{3.15}$$

In order to integrate possible external information (for classification by e.g.) or stimuli, an independent weighted term could be added:

$$\frac{dY_i}{dt} = \sum_{j=1}^{N} W_{ij}\sigma\left(Y_j - \theta_j\right) - \vartheta_i Y_i + wI \qquad \textbf{3.16}$$

Finally, a general transcription rate parameter *tau* is added to describe the differences on the response time from different classes of genes.

$$\tau\frac{dY_i}{dt} = \sum_{j=1}^{N} W_{ij}\sigma\left(Y_j - \theta_j\right) - \vartheta_i Y_i + wI \qquad \textbf{3.17}$$

This model is exactly the CTRNN formalism known as continuous time recurrent neural network (Beer). At the same time this is a series of coupled non-linear differential equations. An advantage of seeing this model as CTRNN is that the work from Funahashi et al (Funahashi, 1989) showed that they are universal aproximators

of smooth functions. Additionally, Albertini and Sontag (Albertini and Sontag, 1993) showed that, if there is enough data, there is just one network topology that best fit the data.

3.6.2 The TDRNN model

However, by the amount and quality of mRNA expression data required by the CTRNN model there would never be enough. Another drawback of this model is that it does not take into account that in Eukaryotes exist an important time delay from gene activation until their product can interact with other genes. Here a modification of eq. 3.17 was introduced by adding a delay (δ_i) to the interaction between genes term:

$$\tau \frac{dY_i}{dt} = \sum_{j=1}^{N} W_{ij} \sigma\left(\left(Y_j\left(t-\delta_i\right)\right)-\theta_j\right)-\vartheta_i Y_i + wI \qquad \textbf{3.18}$$

This time delayed recurrent neural network model (TDRNN) increases the non-linearity with respect to the CTRNN, which non-linear parameter space behavior has been analyzed by other works (Beer, 2006; Mathayomchan, 2002). More important is that the differential time delay for every node moves this model apart from a Markov chain process. Therefore no analytical solution is feasible. Hence this work will follow a statistical approach to validate it in section 4.1.1. Moreover, despite this abstraction is still far from integrating all gene regulation complexity, in this thesis, I will demonstrate the utility of this novel model to represent gene regulatory networks and in the RE task of them.

This TDRNN model is different from previously developed ones (Hu, et al., 2005; Kim, 1998; Liao and Wang, 2003; Ma, 2004) in some senses. The model from Kim demonstrates that TDRNN are superior than previous time delayed neural networks (TDNN), adaptive time-delay neural network (ATNN) and multiple recurrent neural networks (MRNN) for the tasks of temporal signal recognition, prediction and identification. However, even though the general principle is similar, the delayed information processing of his TDRNN is achieved trough the architecture of the

network, having input delay and output layers. Even though this architecture is suitable for these purposes, it makes it impossible to do the mapping of one node to one gene in a multiple regression model as the one here introduced. A similar situation is present on the works from Ma and Liao and Wang, which are artificial neural models developed for more general purposes. However, Ma demonstrates that TDRNN are also capable to develop certain memory for spatio-temporal sequences.

On the other hand, this model is different from the model of Hu et al. in three different aspects; a) This model explicitly model the decay rate associated to the mRNA produced. b) The considered delays are constant for every gene, instead of being particular for every interaction as proposed by Hu et al. One advantage of this is having less parameters to estimate. However, more important is – as I consider - that this is a more realistic situation, because the associated delay is due to translation and diffusion of genes and proteins and it is constant among genes.
c) The non-linearity in my model is considered for every interaction instead as for the entire summation. This marks the same difference as with previous CTRNN models.

In this work, I compared the performance of the TDRNN to infer GRN, respect to the modified CTRNN version here exposed, the original utilized by Wahde and Hertz and an available Dynamic Bayesian Network (DBN) implementation from Wu (Wu, 2004) in the GeneNetwork package.

3.6.3 Robust parameter determination

After performing the optimization runs, the weight matrix parameters were evaluated from their z-score (D´Haeseleer et al., 2000),

$$z_{ij} = \frac{\left|\overline{W}_{ij}\right|}{\sigma_{ij}} \qquad \textbf{3.19}$$

where $\overline{W}_{ij}$ and σ_{ij} denote the mean and the standard deviation of every matrix element W_{ij} is calculated from the 50 independent optimization runs. For the synthetic and

experimental GRN data robust parameters were defined as those having z-scores greater than 2 and 1.5, respectively, corresponding to statistical significance values of ≈ 0.05 and ≈ 0.13375.

3.6.4 Graph generation and error distance measurements

A graph from the robust parameters determined on the previous step was generated, by discretizing every robust parameter to a ternary representation according to values between [1,-1] = 0; [5, 1] = 1; [-1,-5] = -1 generating in this way the, from here on, so-called adjacency matrix. Then we used the Cytoscape (Shannon, et al., 2003) facilities to generate a directed graph from every adjacency matrix.

With the adjacency matrix, the dissimilarity was calculated between it and the desired benchmark network, the repressilator or the Yeast cell cycle, by the use of a directed-weighted version of the graph edit distance algorithm (GED) developed by Robles-Kelly et al. (Hancock, 2005; Robles-Kelly and Hancock, 2005). To calculate the transformation cost between every resultant graph and the target network graph this algorithm takes into account the existence of shortcuts (deletions-insertions) from the semantic of directed weighted graphs. This is especially suitable for middle and large size networks, and is the more objective way to compare between them.

3.6.5 Clustering of results

Cluster analysis simply discovers structures in data without explaining why they exist. Therefore, cluster analysis methods are mostly used when we do not have any a priori hypotheses, but they are still in the exploratory phase of research. Therefore, statistical significance testing is really not appropriate here, even in cases when p-levels are reported, as in k-means clustering. In a sense, cluster analysis finds the "most significant solution possible."

In the RE area case, several dissimilar network topologies could fit the data equally well. To distinguish between these solutions, clustering over the vector representation of the matrix of weights from each experiment was performed. Using the Genesis platform (Sturn, et al., 2002), a hierarchical and self organized maps clustering was used to define the best partition number for a standardized k-means splitting procedure. Then the splitting k-means algorithm was applied with the same parameters for all the experiments to be compared. Robust parameters were identified using the z-score method. To distinguish between these networks, the ratio between the size and the mean fitness of every cluster was calculated.

An important prerequisite for clustering is the need of measuring the similarity or dissimilarity between the elements of the sample to be grouped. There are several different measurements of dissimilarity known as "distance" as Euclidean, Manhattan, squared Euclidean, Chevichev, power distance etc. In this work the Euclidean distance was utilized in all the cases where clustering is referred. It simply is the geometric distance in the multidimensional space and it is computed as:

$$dissimilarity(x, y) = \left(\sum_i x_i - y_i \right)^{1/2} \quad \textbf{3.20}$$

Being x and y the two different elements of the sample to be clustered where I want to measure the dissimilarity and I represents the number of different dimensions measured. Euclidean (and squared Euclidean) distances are usually computed from raw data, and not from standardized data. This method has certain advantages (e.g., the distance between any two objects is not affected by the addition of new objects to the analysis, which may be outliers). However, the distances can be greatly affected by differences in scale among the dimensions from which the distances are computed.

Once the distances between elements of a given data sample were calculated, the clustering algorithm groups or splits the sample into subgroups by mainly two different strategies:

a) agglomerative techniques; starting from being every element on an isolated group, those with the lowest distance are grouped together until ending with one unifying cluster. This technique is typical for hierarchical clustering.

b) Splitting techniques; where given the initial set of data it is spliced according to it elements similarity until ending with a desired number of clusters or a certain rule is accomplished. This last technique is typical for k-means clustering.

Hierarchical Clustering

Hierarchical methods return a hierarchy of nested clusters, where each cluster typically consists of the union of two or more smaller clusters. The hierarchical methods can be further distinguished into agglomerative and divisive methods, depending on whether they start with single object clusters and recursively merge them into larger clusters, or start with the cluster containing all objects and recursively divide it into smaller clusters.

K-means Partitioning

The *k-means* algorithm (MacQueen, 1967) can be used to partition N genes into K clusters, where K is pre-determined by the user. Where K initial number of clusters is chose by the user, and each distance among genes is calculated. Then starting from the lowest K distances, every gene is assigned to the cluster with the nearest mean named centroid. Next, the centroid for each cluster is recalculated as the average expression pattern of all genes belonging to the cluster, and genes are reassigned to the closest centroid. Membership in the clusters and cluster centroids are updated iteratively until no more changes occur, or the amount of change falls below a pre-defined threshold. K-means clustering minimizes the sum of the squared distance to the centroids, which tends to result in round clusters. Different random initial seeds can be tried to assess the robustness of the clustering results.

Self –Organizing Maps clustering

The *Self-Organized Map* (SOM) method is closely related to k-means However, the method is more structured than k-means in that way that the cluster centers are located on a grid. In each iteration, a randomly selected gene expression pattern attracts the nearest cluster center, plus some of its neighbors in the grid. Over time, fewer cluster centers are updated at each iteration, until finally only the nearest cluster

is drawn towards each gene, placing the cluster centers in the center of gravity of the surrounding expression patterns.

Drawbacks of this method are that the user has to specify *a priori* the number of clusters (as for k-means), as well as the grid topology, including the dimensions of the grid and the number of clusters in each dimension (e.g. 8 clusters could be mapped to a 2x4 2D grid or a 2x2x2 3D cube). The artificial grid structure makes it very easy to visualize the results, but may have residual effects on the final clustering.

3.6.6 Dynamic Bayesian Network

To compare the performance of the TDRNN with an established modeling framework, the dynamic Bayesian network (DBN) approach implemented in the GeneNetwork package (Wu et al., 2004) was used. The performance of the TDRNN and DBN networks were compared on the same experimental data set (see section 4.2.1) under the same conditions: the networks were inferred from 100 linearly interpolated data points. The GA implementation of the GeneNetwork package was set to use a population size of 1000 individuals, running for 1000 generations with a mutation rate and crossing over rate of 0.05 and 0.5, respectively. For the TDRNN the optimization scheme as described in table 1 was used.

4. Results

This section is divided into three main parts. The first part contains the results obtained with synthetic data of a system analogous to the so-called repressilator synthetic system. Additionally, in this section, aspects are exposed to analyze the introduced Time Delayed Recurrent Neural Network (TDRNN) model, through a parallel study with a Continuous Time Recurrent Neural Network. This analysis and the address uncovers some open questions in the reverse engineering area of gene regulatory networks area as network sparsity, over-fitting and information required by the here introduced model. In the second part the results obtained with the TDRNN model of real data of the yeast (*Saccharomyces cerevisiae*) cell cycle are compared with other approaches as the previously used CTRNN model and a Bayesian dynamic network. In the third part, results related to the keratynocytes-fibroblast communication system will be presented.

4.1 Synthetic benchmark: The Repressilator

To assess the inference power of the TDRNN and CTRNN models, I tested both models on generated synthetic data of the so-called repressilator system. The repressilator system was chosen because it is among the simplest experimental synthetic systems showing realistic characteristics of GRN as cyclic behavior. Cycles occur often in biochemical networks and some of the more promising models for the reverse engineering of GRN, the Bayesian networks (de Jong, 2002), cannot infer cyclic networks. Furthermore, the repressilator work has become a good bench work for different models on different areas (Elowitz, 2000).

The original repressilator is a synthetic GRN engineered in the E.coli bacteria, and constitutes a network of three mutually repressing genes capable of undergoing limit cycle oscillations. In this work, the repressilator dynamics is represented by three sine waves derived from eq. 4.1 with a phase shift of $2/3\,\pi$ between each of them.

$$y = \sin\left(x \cdot 2\pi/150 \pm 2/3\,\pi\right) \cdot 5 \qquad \textbf{4.1}$$

The sine curves have amplitude of 5 units and a period of 150 units to be in the period time scale of the original work and on the expression amplitude scale of microarray data as depicted in figure 4.1a.

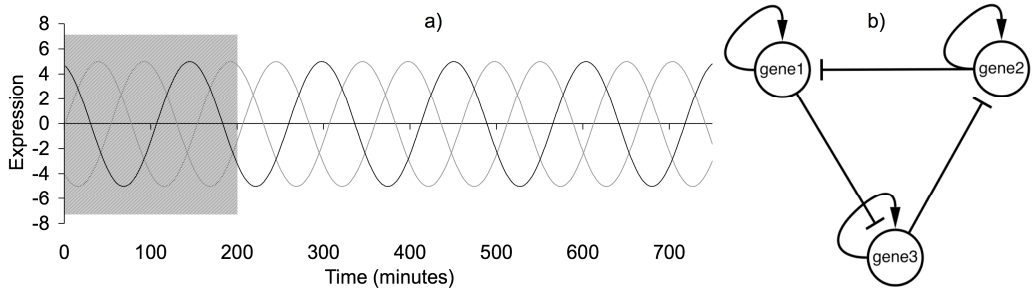


Figure 4.1 Repressilator scheme. On a) is shown the oscillatory dynamics of the three mutual repressing genes named repressilator b). This is a synthetically engineering system in E.coli. The shadowed time elapsed in a) is assumed to be a relaxation lapse, therefore is not included into the optimization process.

With this synthetic data, the GA optimizations runs of the next sections were performed. But in all cases, the first 200-simulation time units were discarded (see figure 4.1a) to allow the model to reach an oscillatory steady state.

To represent the expression induction of every gene in the original experiment, the recursive connection from every node is the only possibility in our working scheme. Hence the goal network will consist of three nodes and six edges: auto-regulation of each node and mutual repression in a cyclic way as depicted in figure 4.1 b. This has the advantage of keeping the model from being driven by any arbitrary input function.

4.1.1 Parameter space selection

As has been stated, the biological systems to be engineered are highly undetermined, and therefore exist an infinite number of parameter combinations able to equally well fit the data. Additionally, the models tested here are semi parametric and in principle have no bounds in the ranges their parameters could take. This models are focused to perform the RE task at the network organization level rather than inferring kinetic constants. Therefore, to choose the right parameter space is not a trivial task. In fact, the chosen parameter is a compromise between different restrictions and objectives.

On the one hand, a large parameter space is desired in order to assure the convergence between the model and the data. Additionally, some aspects of the parameter space could be of particular relevance through a broader parameter range. In this case, it is the range of the taus (τ) parameter, because the systems to be modeled have different response time scales as the fast (in the order of minutes and probably seconds) signal transduction process and the slower (in the order of hours) GRN. The range of the τ parameter was chosen to be as broad as $3 \leq \tau \leq 66$ on the rest of this section. The reason for this choice is that this result should also be valid in the next sections with experimental data sets involving both kinds of processes.

Additionally, it is highly desirable to work under narrowed parameter space to speed the searching task. But it is more important that some parameters should not correlate with the RE task. Ideally, only the weights parameters should be correlated to a given

network topology. Therefore, it is highly desirable that the rest of the parameters exert their influence mainly to fit the data, here, it is the bias parameter. In the absence of any external stimuli, the individual biases are the only sources of dynamic behavior. Actually, the effects of the biases and the external inputs on the locations of the regions known as nullclines in the synaptic input space are identical (Beer 1995). Therefore, considering these aspects, the chosen biases parameter space has the $-1 \le \theta \le 1$ short range.

Another parameter that should have a small effect on the RE task but plays an important role to fit the data in a biologically inspired way, is the decay (υ) parameter. However, since the present generalized additive models do not impose upper and lower limits to the activity space of every node as it do the generalized linear models, the maximum activity of every node scales with the network size according to equation (3.18):

$$Y_i(\max) = \lim_{N \to \infty} \left(\sum_{j=1}^{N} \sigma\left(Y_j - \theta_j\right) \right) = N \qquad \textbf{4. 2}$$

Therefore, the υ decay parameter that balances the global activity of every node on the TDRNN should also vary proportionally. Hence, here were used the $Log_{10}(Y_i(\max))$, giving a $0 \le \vartheta \le 0.5$ range for the repressilator data set and $0 \le \vartheta \le 3.5$ for the experimental data set presented on the next sections.

Parameter screening

Finally, for the delay δ parameter, I performed a fast screening of the ranges this parameter together with the taus τ parameter and beta pruning controlling factor could take and I chose the range where the TDRNN model was performing better in respect to the MSE and to the RE task. To evaluate this, the resultant weights of every optimization run of the screening was discretized to a ternary representation: [1,-1] = 0; [5,1] = 1; [-1,-5] = -1 and compared with the goal network adjacent matrix of figure 3.1b. The screening was performed with a population size of 100 individuals. The results of this screening are scatter-plotted on figure 4.2.

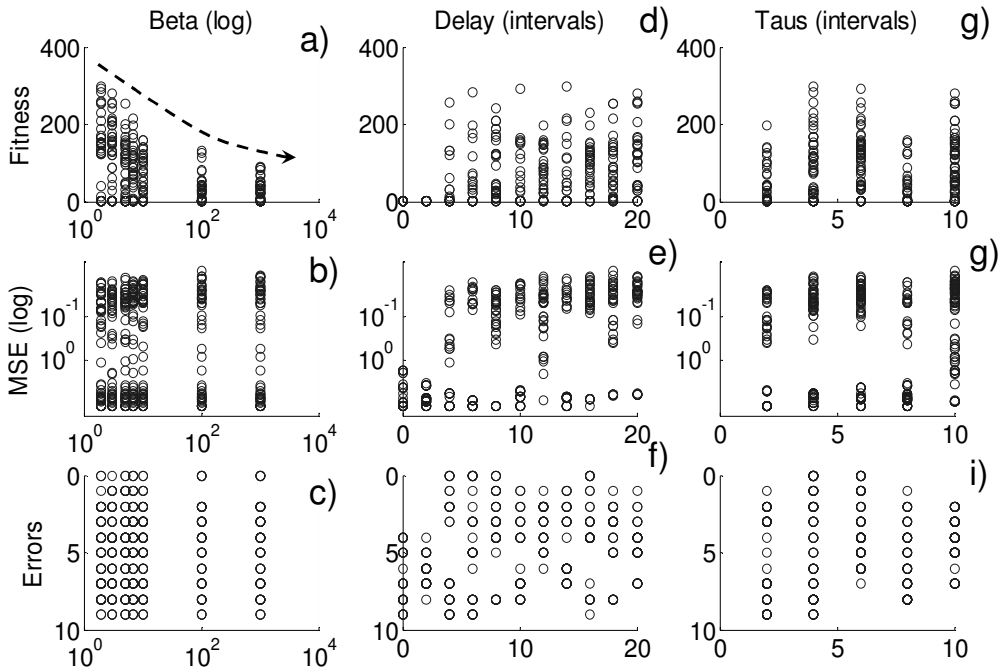


Figure 4.2 Scatter-plot of the parameter space screening. On a, b and c panels are the scatterplots of the influence of pruning over the fitness and inference power of the model. In the middle vertical panels d, e and f, is represented the influence of the delay parameter of the TDRNN model over the same performance indicators fitness and errors. Panels g, h and i shown analogous information for the influence of the taus τ parameter. In dashed lines are represented the asymptotic or oscillatory tendency that some parameters to follows.

On figure 4.2 a, b, and c is possible to see the functioning of the pruning function. On 4.2.a values of the fitness function clearly decrease for larger values of the *Beta* β pruning controlling factor of equation 3.5. Notice that for $\beta = 10^2$, β plays almost no role and for $\beta = 10^3$ does not play any role at all. On the panels 4.2 b and c is shown that the MSE and the inference power (expressed in number of errors) are not affected by the pruning process ($\beta \leq 10^2$) despite the fitness varies with the beta factor. On 4.2.b, it is possible to see two distributions; on the upper part are runs with a good fit (MSE on logarithmic scale) and on the lower part are runs with a poor fit of the data. However, this bimodal distribution is present for all Beta values. On 4.2.c, the scatter-plot suggests that those runs with no pruning ($\beta \geq 10^2$) are only slightly more sensitive concerning the inference power, showing no runs with one error.

On the scatterplots of the delay (δ) parameter ranges (figures 4.2. e and f), it is possible to observe the same bimodal distribution previously observed. Here, it is

possible to see that the inference power of the model for a delay range of six units is in a maximum performance zone. This apparently oscillatory behavior of the parameters ranges is clearly related to the high symmetry of the repressilator system. As demonstrated by the works of Beer (1995, 2006), the parameter space of the CTRNN for such a dynamical system is divided into regions of topologically-equivalent dynamics by bifurcation manifolds. Hence, for the reasons previously exposed, it was chosen to work with the shortest delay interval of six units showing the best performance.

An analogous situation to the delay parameter could be observed on the figures 4.2 g, h and i for the taus (τ) parameter. However, for the reasons previously exposed, here it was chosen to work with a broader parameter space to observe the implications of this cyclic behavior in just one of the parameters. Ideally this parameter should work on a broader space to cover diverse biological systems. However, since the others parameters are constrained to a smaller space, the manifolds should not appear. Under these circumstances, problems related to the solution of the systems of stiff differential equations could rise, see next sections.

Parameter correlations and inference power

To corroborate that the inference power of the TDRNN and CTRNN models is mostly insensitive with respect to the chosen parameters ranges ($3 \le \tau \le 66, -1 \le \theta \le 1,$ $-1 \le \upsilon \le 1,\ -5 \le W \le 5$), I calculated all correlations in between parameters, to the MSE and the inference power (as means of errors). For this purpose, 150 optimization runs under the previous standardized conditions were performed with the TDRNN model using a $0 \le \delta \le 6$ range. I used a population of 100 individual's size. To discard spurious correlations due to none fitting problems (as observed by the bimodal MSE distributions) a MSE= 0.4 was used as stopping criteria (see figure 4.3a). This is a normal optimization procedure when most of the runs share a similar MSE.

To avoid interference, the pruning function was not used. On figure 4.3 the global distributions of the parameters are depicted by their histograms (figure 4.3 c, d, e, f and g).

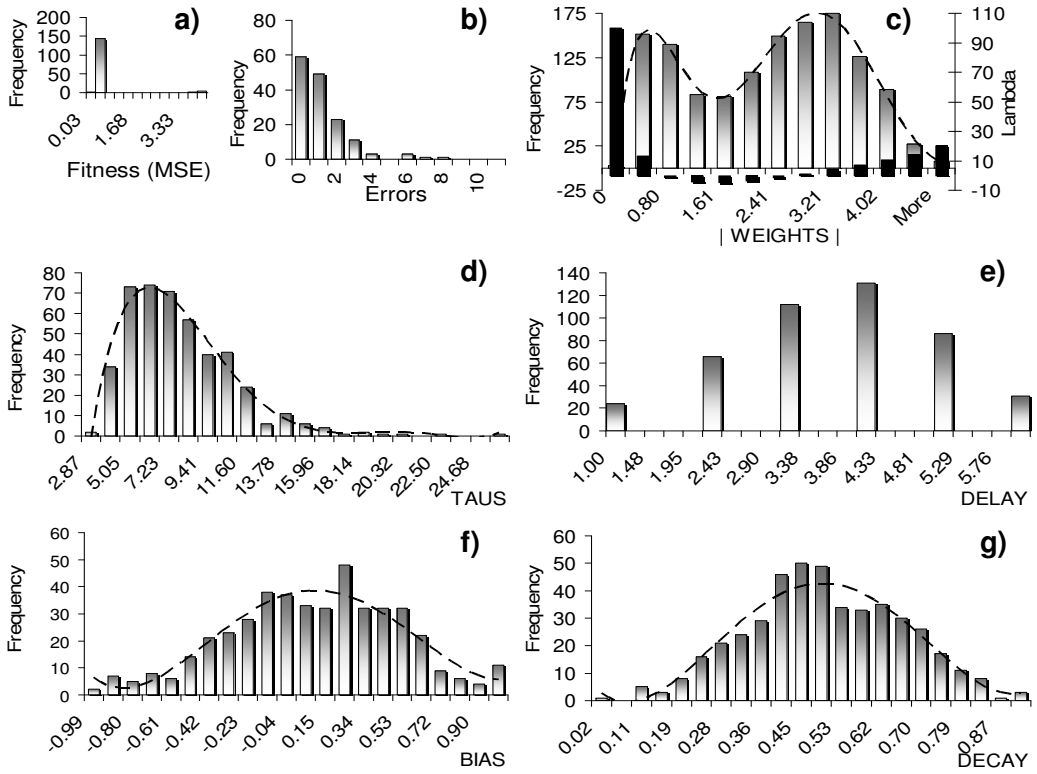


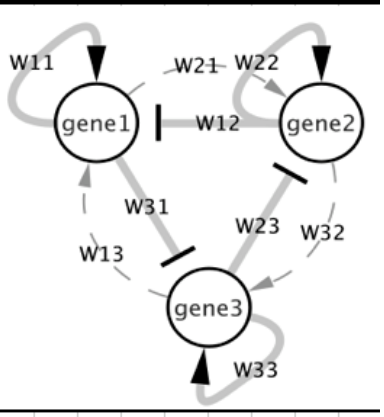
Figure 4. 3 Parameters histograms

The distributions depicted on figures 4.3 e, f and g, suggest a normal distribution for the global decay, delay and bias parameters; this was corroborated by a Kolmogorov-Smirnov test of normality ($p \leq 0.05$). Interestingly, figure 4.3.c shows a bimodal distribution of the absolute value of weights. One of the processes appears to be close to zero while the other is centered on interactions around 3 units. These are the expected distributions for a network using 2/3 of it full connectivity as it occurs in the repressilator topology. Superposed, in figure 4.3.c are the values of the Lambda function (black bars using the second y axis scale at the right side) for the same weight values. Notice that the shape of the λ function, exactly penalize the weights between the two distributions, at the same time this λ function promote the moving of the two distributions towards zero and larger than 3 units values respectively.

This pruning distribution is desired because it is easy to probe correlations with large interactions, and obviously no interaction is the most desired feature of the pruning function.

Table 4.1 Pearson correlation among parameters and fitness (MSE) and inference power (ERS).

	ERS	MSE	T1	B1	D1	C1	T2	B2	D2	C2	T3	B3	D3	C3	W11	W12	W13	W21	W22	W23	W31	W32	W33
ERS	1																						
MSE	0.1	1																					
T1	0.3	-0	1																				
B1	0.1	***0.5***	0.2	1																			
D1	-0.1	-0.2	-0.1	-0.2	1																		
C1	-0.3	-0.2	-0.2	-0.3	0.2	1																	
T2	0.4	0.1	-0.1	0	-0.1	-0.1	1																
B2	0.4	0.2	-0.1	0.1	-0.1	-0.2	0.2	1															
D2	-0.1	-0.3	0	-0.1	0.2	0.2	-0.1	-0.3	1														
C2	-0.2	0	0.1	0	0.1	-0.1	-0.1	-0	-0.2	1													
T3	0.2	-0.1	0	-0.2	0	0.2	-0	0.1	0.2	0	1												
B3	0.1	***0.6***	-0.2	0.3	-0	-0.2	0	0.2	-0.2	0.1	-0.1	1											
D3	-0.1	-0.2	-0	-0.1	0.1	0.1	-0.1	-0.2	0.1	0	-0.1	-0	1										
C3	-0.3	0.1	-0.1	0.2	-0	-0.1	0.1	-0	-0.2	0	-0.3	0.1	-0	1									
W11	-0.2	-0	-0.3	-0.2	0.1	***0.9***	-0	-0.1	0.1	-0.1	0.1	-0.1	0	-0	1								
W12	0	-0.1	0.5	-0	0.2	0.6	-0.2	-0.2	0.2	0	0.1	-0.2	0	-0.1	0.6	1							
W13	***0.5***	0.2	***0.8***	0.3	-0.1	***-0.5***	0	0.1	-0.1	0	-0.1	0.1	-0.1	-0	***-0.5***	0.2	1						
W21	***0.5***	0.1	0	0.1	-0.1	-0.2	***0.9***	0.2	-0.1	***-0.4***	-0.2	0	0	0	-0.1	-0.2	0.2	1					
W22	-0.4	-0.1	0	-0.1	0.1	0.1	-0.3	-0.1	-0.1	***0.9***	0.2	0	-0	-0	-0	0.1	-0.1	***-0.6***	1				
W23	0.1	0.1	-0	0	-0.1	-0.1	0.7	0.1	-0	0.5	0.1	0.1	-0.1	-0	-0.1	-0.1	0	0.3	0.4	1			
W31	0.1	0.2	0	0.2	-0.1	-0.1	0.1	0.2	-0.3	0.1	0.2	0.2	-0.1	0.6	-0.1	-0	0.2	0.1	0	0.1	1		
W32	***0.5***	-0	0.1	-0.1	0.1	0.1	0.1	0.2	0.1	0	***0.6***	0.1	-0	***-0.5***	0	0.1	0.1	0.1	0	0.1	0.2	1	
W33	-0.4	0	-0.1	0.2	-0.1	-0	0	-0.1	-0.1	0	-0.2	-0	-0	***0.9***	-0	-0.1	-0.1	-0.1	0.1	-0	0.4	-0.5	1



ERS = number of errors **T=taus** **B = bias** **C= decay** **D=delay**

On the 4.3.d figure, it is clearly shown that the τ parameter have a Log normal like[3] distribution skewed to the right side, with a median toward short values (the peak is around 7 units). This shows that the model does not have any problems integrating stiff equations systems; instead, it automatically chose those values that promoted its stability according to the period size on the approximated sine functions.

Since the size of the sample was bigger than 100 elements, deviations from normality on the previous distributions are less important (Hill and Lewicki, 2006), therefore Pearson correlation (p-value ≤ 0.05) was calculated between every parameter pair as well as to the MSE and the inference power (expressed in errors). The resultant correlations are in table 4.1

[3] The lognormal distribution could be used to model the time required to perform some task when "large" values sometimes occur. It is always skewed to the right and it has a longer right tail than the gamma or Weibull distributions. The lognormal distribution is closely related to the classical normal distribution

On the 2nd and 3rd columns of the table 4.1, are shadowed the two more important correlation series. The first is the correlation between the nodes individual parameters and the fitness (MSE 3rd column) and the second between the same nodes individual parameters and the inference power represented by the individual-errors (ERS, 2nd column header). Here, it is demonstrated that no other parameter than the weights could exert a stronger influence to the inference power of the model. Particularly, at the ERS column, it is quite obvious that the more important weights concerning the inference power are those which should be eliminated (W_{13},W_{21} and W_{32}) to obtain the repressilator topology.

By contrast, the bias (B1 and B3 rows on the table 4.1) is clearly the parameter that correlates more strongly to the fitness (MSE column) of the model. These results corroborate the assumption that only the weights correlate to the inference power while the rest of the parameters mostly correlate to the fitting of the data. This result is of high importance because of results from sections 4.1, 4.2 and 4.3; the analyses will be focused on the square matrices of the weights (W_{ij}).

One can argue that the weights should correlate strongly to the inference power [4] since it is calculated exactly on the weights. However, this is not true since the errors are calculated in relationship to the ternary discretization of the weights. This decreases the variance of the errors respect to the weights; therefore, this is not an auto correlation measurement. Moreover, it is important to notice that even some weights almost do not correlate to the errors, as W_{13}, W_{21} and W_{32} - which are exactly those of the mutual inhibitions - despite they are strongly related to the desired topology.

Notice that in general there are mostly low correlations between the parameters and the fitness and the inference power as one can see at the first two columns of the correlations in the 4.1 table. Again, one can argue that it is that way because the distribution of the fitness is monotonic with a low variance, as the MSE was prefixed as stopping criteria. Nevertheless, this is not the case of the inference power. On figure 4.3.b, the histogram of the errors distribution is shown and it clearly shows that

[4] the inference power here refers to the inverse of errors

errors have a geometric distribution[5] as opposed to the MSE. This means that actually, one should expect stronger correlations for the inference power to any other column than the expected for the fitness, but only the weights have this characteristic.

Instead of strong correlations between parameters and fitness or inference power, on table 4.1 there are strong correlations (shown with grey scale background) between the parameters and the weights (depicted on the 3rd quadrant of table 4.1). Particularly, the strongest interactions are between the decay parameter and all the edges of a given node (depicted in respective node's boxes in table 4.1). In decreasing order of correlation strength, those with the stronger correlations, like $r \geq 0.9$, (cells with black background and white numbers) are the direct correlations between the positive auto-feedback loop of every node (W_{11},W_{22} and W_{33}, see the weights scheme into the table 4.1) and their respective decay parameter (see Figure 4.4 a, b and c). These correlations are expected for this model in order to stabilize its outputs. So, for a bigger positive auto-feedback loop a bigger stabilization decay parameter is needed.

[5] The geometric distribution (discrete) with **probability** = *p* can be thought of as the distribution of the number of failures before the first success in a sequence of independent Bernoulli trials, where success occurs on each trial with a probability of *p* and failure occurs on each trial with a probability of 1 - *p*.

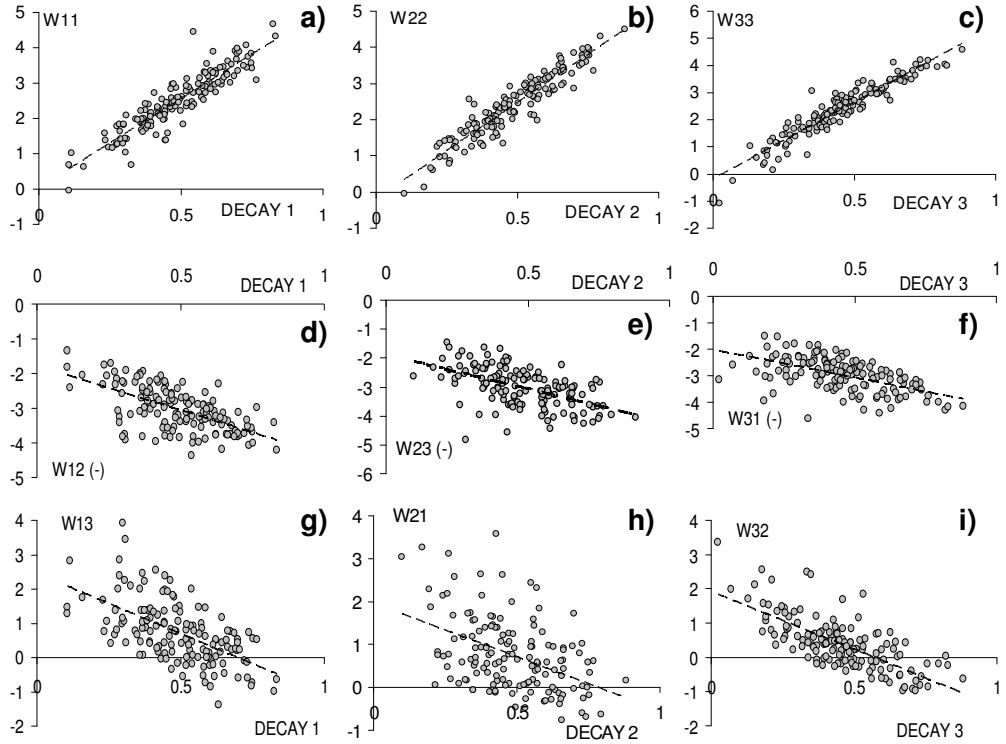


Figure 4. 4 Principal interactions scatterplots

The next strength correlation level ($0.5 \leq r \leq 0.6$) for interactions between the decay parameter and weights corresponds to those weights of the mutual repression (W_{12}, W_{23} and W_{31}) edges. Particularly, the correlations are between the decay parameter of a given node and the incoming repression edge from another node. Notice that since those weights are negative, the correlation is a direct and not an anti-correlation as one should expect for two processes that act together decreasing the node activity. The last mentioned process is clarified on the scatterplots of figures 4.4 d, e and f, where the angle of the correlation trend is negative, shown in the figure with dashed lines.

Since no important correlation conclusions should be done based just on Pearson correlations, a scatter-plot visual examination has been performed for all the interaction pairs between parameters to check for any possible omission or spurious correlation, because a nonlinear correlation could occur, or spurious correlations due to bimodal distributions also could takes place and both cases could been mishandled by a linear (Pearson) correlation.

Additionally to the mentioned upper panels scatterplots of figure 4.4, the rest of the more important scatter plots are shown in figures 4.4 middle panels (d, e and f) and 4.4 lower panels (g, h and i). In figure 4.4 middle panels, just linear correlations (dashed lines) or anti-correlations were found for all cases, showing that even the time delay increases the nonlinear behavior of every node's activity, the global behavior is far from being unpredictable.

Finally, in respect to the decay parameter and with an anti-correlation ($-0.5 \leq r \leq -0.4$), lie those correlations between the decay parameter of a given node and the weight of the incoming edge from another node. Notice that there are just two incoming edges per node: the repressive one and the other one, this last is mostly working as an activator. In this case, the reference is to the W_{13}, W_{21} and W_{32} weights that should not exist or being slightly positive.

Proceeding with an analogous study for another parameter, I found on table 4.1 that the highest level of correlation strength ($0.6 \leq r \leq 0.9$) corresponds to the correlation between the taus (τ) parameters of a given node and the incoming activation from another node to the first one (see the figure on first quadrant of table 4.1.) Notice that this is the edge that should not exist. According to this, the stronger the incoming activator weight the larger the τ parameter, meaning a smaller expression ratio of the node in question. The same situation occurs for the incoming negative repression edges (W_{12}, W_{23} and W_{31}) and the τ parameter of every node, but at lower correlation strength ($0.2 \leq r \leq 0.7$). These two correlations are logically needed to stabilize the so-called node's reactivity or reaction rate. The larger the weights strength is, the larger is the τ parameter which needs to be on the equation 3.18 to stabilize the global activity of the node:

$$\frac{dY_i}{dt} = \frac{1}{\tau}\left(\sum_{j=1}^{N} W_{ij}\sigma\left(\left(Y_j\left(t-\delta_i\right)\right)-\theta_j\right) - \vartheta_i Y_i + wI \right) \qquad \textbf{4.3}$$

However, for the interactions between the τ and the auto regulatory edge weights (W_{11}, W_{22} and W_{33}), the weak range ($-0.5 \leq r \leq -0.3$) of anti-correlation is partially explained by the fact that, in order to produce an oscillatory behavior, this kind of

dynamical systems requires amplification. In this sense, the shorter the taus parameter is, it needs a larger auto regulatory edge weight to keep the system oscillating.

Notice that the last two parameters analyzed are those which are not passed trough the sigmoid function. This it is very logical because their range of influence over the entire node activity is in the same order than the weights. Therefore a similar strong influence is expected from the external I input term from the previous equation 4.3 (or 3.18). From here on, one has to be very cautious while analyzing this model with an external input because then it could be not possible to separate its influence on the Matrix of weights (Wij) as it is proved in this study.

Without pretending to create an unnecessary statistical model about the TDRNN model topology, the next analysis was looking at the scatter plots of every pair of interaction among weights in order to corroborate the absence of non-linear interactions among them. The relevant correlation results are plotted in figure 4.5, where the encountered strongest correlations on the 4th quadrant of table 4.1 were those without a normal distribution of the weights.

Notice that all but two correlations (W_{32}-W_{33} and W_{31}-W_{35}) show a linear correlation (dashed trend lines) among them. However, the two correlations that are better represented by a second order trend line are those with sign transitions in the weights, while those showing all the data distributed in one quadrant show a linear correlation.

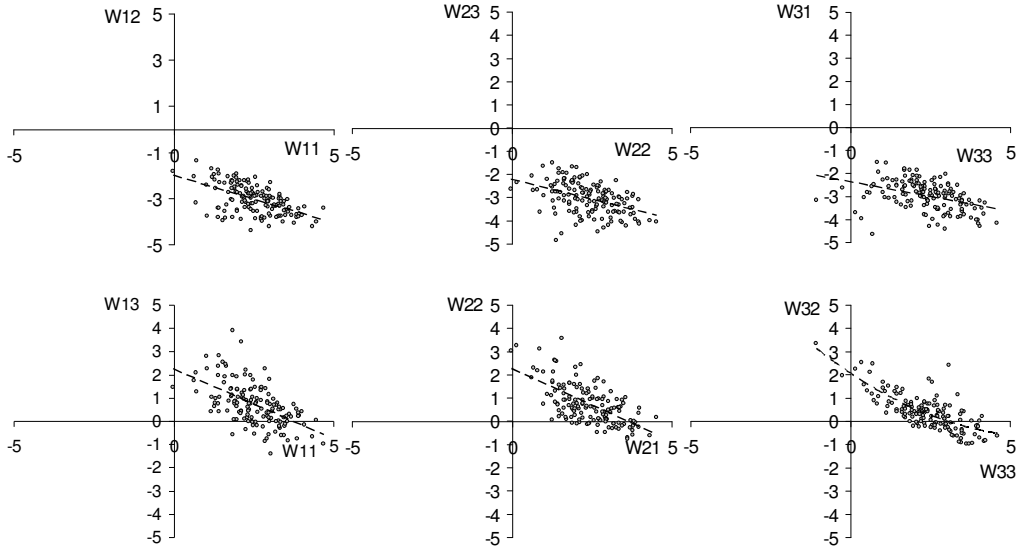


Figure 4. 5 Scatter-plot of the principal weights interactions

All this results are consistent with the expected behavior according to the works from Beer (1995, 2006). More important is the fact that this results corroborate that the inference power of the TDRNN and CTRNN models is mostly insensitive concerning the chosen parameters ranges ($3 \le \tau \le 66$, $-1 \le \theta \le 1$, $-1 \le \upsilon \le 1$, $-5 \le W \le 5$) of the former results, restricting in this way the further analysis of the Wij matrices of weights as described in the methods section.

4.1.2 Required data length.

To determine the data required to reverse engineer the cyclic repressilator system, the model was simulated for 0.2, 0.33, 0.5, 0.6, 0.75, 1.0, 1.5, 2, and 3 oscillation periods after an initial transient of T = 200 units. The 50 optimization runs were filtered from outliers (MSE-mean +/- 2 standard deviations) and the robust parameters and adjacency matrices were identified for each of the 9 simulation intervals as described in methods. To compare the performance of the networks, the errors were defined as the numbers of mismatches between the adjacency matrices of the inferred and the goal network showed in figure 3.1b.

The results are plotted on Fig. 4.6 as a comparison between the errors of the TDRNN and the CTRNN models for different simulation runs using weight pruning (P) or unbiased parameter selection (not pruning - NP). Summing over all errors from all simulations of nine time intervals the TDRNN-NP model performed best, having only three errors. In comparison, the reverse engineering using the CTRNN-P, CTRNN-NP and TDRNN-P gave a total of 11, 7 and 10 errors, respectively.

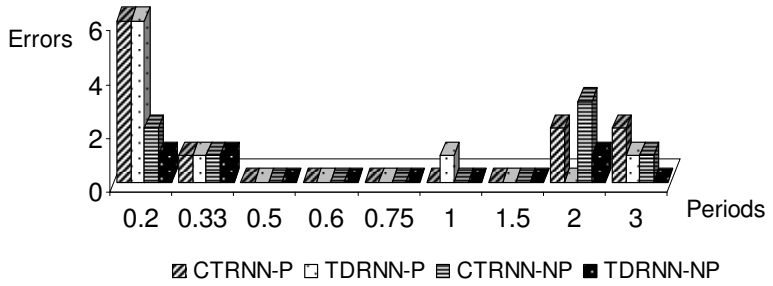


Figure 4. 6 Inference power comparison along nine different optimized period intervals between the TDRNN and CTRNN models with and without parameter pruning (P, NP). The comparison is expressed in terms of errors, defined as mismatches between the robust parameters calculated by any model and the topology of the goal network. For 0.5 periods, all models can infer the topology without errors, but below and higher than 1.5 periods, NP and TDRNN perform best.

In order to interpret the previous result in the light of the data fitting by every model, a comparison of performance to fit the data by the four different model-reverse engineering combinations (in the following called models for simplicity) is shown in the boxplots of figure 4.7 This boxplots produces a box and a whisker plot for each of the simulated periods of every model. The box has lines at the lower quartile, median and upper quartile values. The whiskers extend from each end of the box to the adjacent values in the data of the fittings, the most extreme values within 1.5 times, the interquartile range from the ends of the box. In this boxplots, outliers are those fittings with values beyond the ends of the whiskers. Outliers are displayed with a + sign. Notches display the variability of the median between samples. The width of a notch is computed so that boxplots whose notches do not overlap have different medians at the 5% significance level. The significance level is based on a normal distribution assumption, but comparisons of medians are reasonably robust for other distributions. Comparing boxplot medians is like a visual hypothesis test, analogous to the t test used for means.

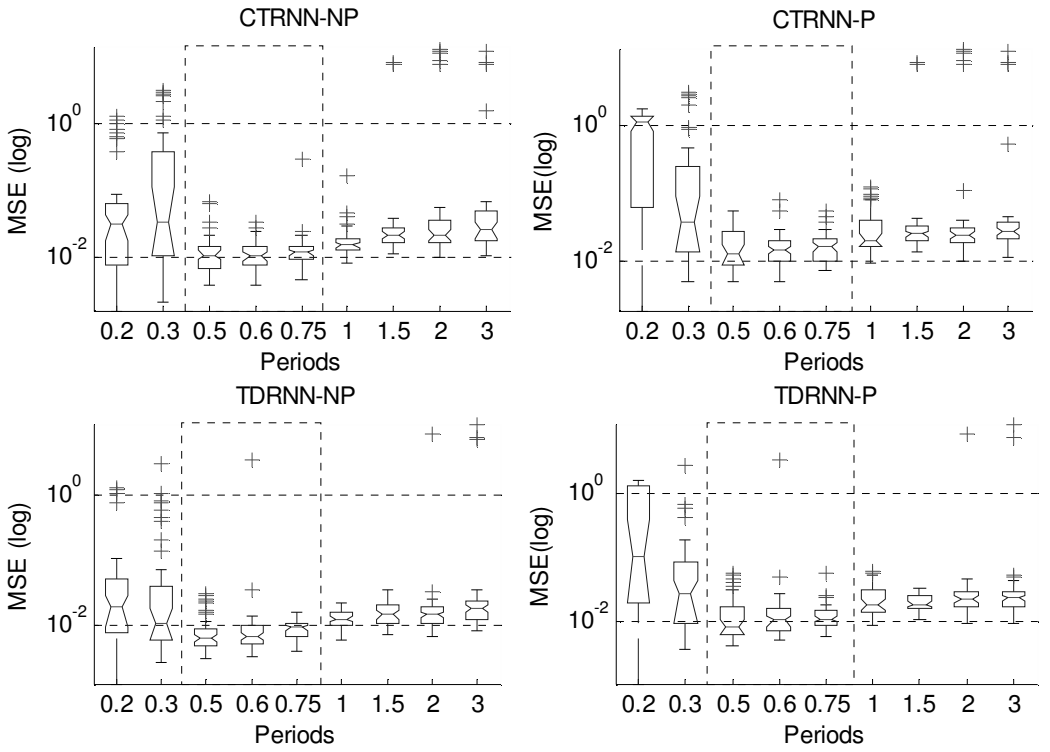


Figure 4.7 Data fits comparison among the TDRNN and CTRNN along different optimized period intervals using (P) and not (NP) the adaptive pruning function. For every group an easy-to-fit zone is identified into the dashed line box. Outliers of every group distribution are represented by "+" sing.

The results showed in figure 4.7 clearly show 3 zones (separated by vertical dashed lines) of fitness values for all the models. In all cases, the central region comprising the 0.5, 0.6 and 0.75 period intervals, are the regions with better median fitting of the data. These regions are called henceforth easy to fit regions. However, notice that in all cases the inferior whiskers (extreme data) follow a clear decrement with the quantity (expressed in period fractions) of fitted data. In this way, for all models the extreme lower fitting is close to 0. This last result is logical and was expected, by contrast, the easy to fit regions were an unexpected result.

Additionally to the boxplots a normality distribution test was performed (Lillie test, $p < 0.05$), and since the distribution of the MSE deviated from normality between all models, a parameter free test (Kruskal-Wallis test, p-value < 0.05) was performed to compare between MSEs. Here I found that the TDRNN-NP model fitted the data significantly better in 7 out of the 9 intervals. In figure 4.8 are shown the respective boxplots showing the same result favourable for the TDRNN model.

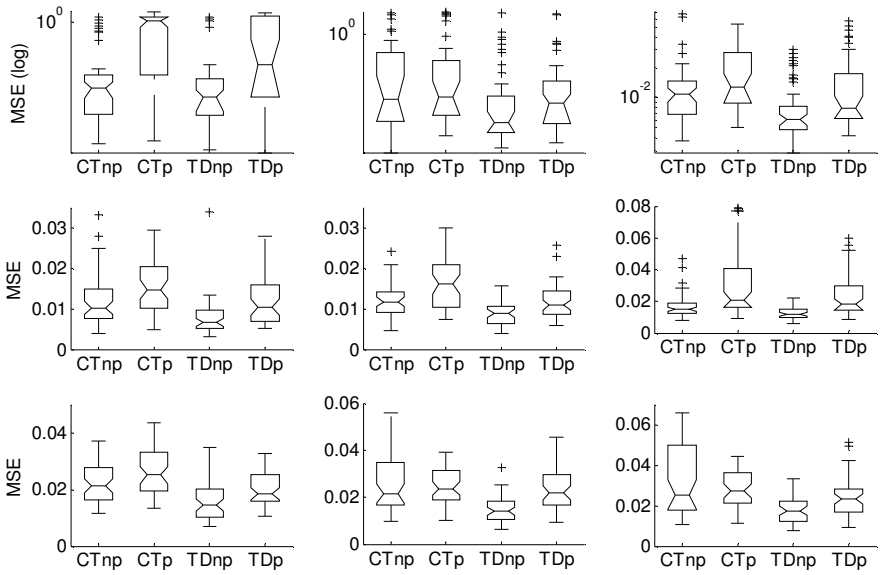


Figure 4.8 Comparison of the fitting by the two models using (p) and not (np) the pruning function at every optimized time window. CT = Continuous time recurrent neural network, TD = Time delayed recurrent neural network.

This result suggests that fitness would be a well-suited criterion to determine the topology inference power by the models. However, this is a very strong statement that has to be confirmed or rejected. Therefore, to corroborate this result in the light of possible parameter-overfitting, the individual-errors of every optimization run was calculated by discretizing every resultant weight matrix of the optimization runs to a ternary distribution according to the following mapping: [-5,-1] -> -1 , [-1,1] -> 0, [1,5] -> 1. It is important to notice that even for this discretization, the goal network is just one in $3^{N\wedge 2}$ = 19 683 possible solutions. In this way, information was obtained to correlate individual fitness (MSE) with individual quantification of errors (individual-errors, from here on).

Since MSE and individual-errors distributions were not normal, a parameter free test (Spearman) was performed to correlate these two variables along every model group. In figure 4.9, the bars depict the Spearman correlation between the fitness, the individual-errors and their respective p-values (shown by the lines with second *y* axis) for each simulation interval and model respectively, after filtering for the 2 standard deviations.

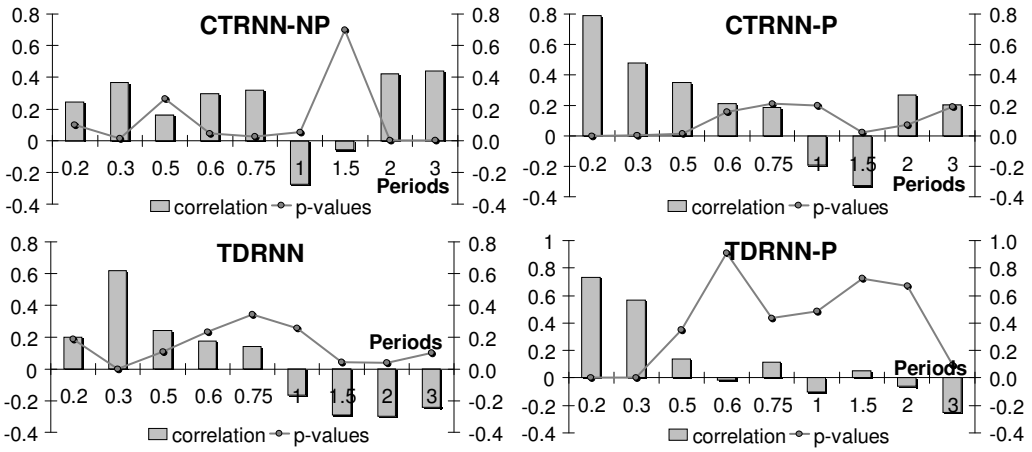


Figure 4. 9 Spearman correlation between individual-run-errors and their respective fitness (MSE) for the 4 models along the 9 optimized time intervals. Corresponding p-values are show as thin red lines**.**

The correlation between individual errors and fitness is weak and varies with the data structure and the model equation used to represent the GRN. For the simulation time interval 0.5<T< 0.75, where T denotes the time in multiples of the oscillation period, the MSE median was significantly lower (in the so called easy to fit region) for all models with a weak, yet significant correlation between fitness and errors. Such a correlation was strongest (Spearman ≈ 0.5, $p < 0.05$) in the region 0.2<T<0.3 for all models. Optimizing for one oscillation period (T=1) it is found an anti-correlation between fitness and errors in all models. Notice in figure 3.11 the strong change in correlation between 0.75<T<1 for the CTRNN-NP model. In contrast to the TDRNN model, the CTRNN model exhibits again a strong correlation for the remainder of the MSE intervals (2 <T<3). This result rejects the suggestion that fitness could be a good indicator of inference power.

To explain the existence of these 3 MSE zones, the dynamics of every optimization run was analyzed for a longer period of time (T=7.2). In the upper part of figure 4.10, this is exemplified by plotting the dynamics of three different period fractions from the TDRNN-NP model representing these 3 regions. To measure, in some extent, the stability of every solution, the instantaneous-fitness was calculated along this time interval for every run. At the lower part of figure 4.10 is depicted the so called instantaneous-fitness along the dynamics of every run of the respective dynamics

from the upper part. Here, the mean of the instantaneous-fitness of every model is represented by a thicker black line and the standard deviation by a dashed red line. The optimized interval is depicted by a vertical black dashed line.

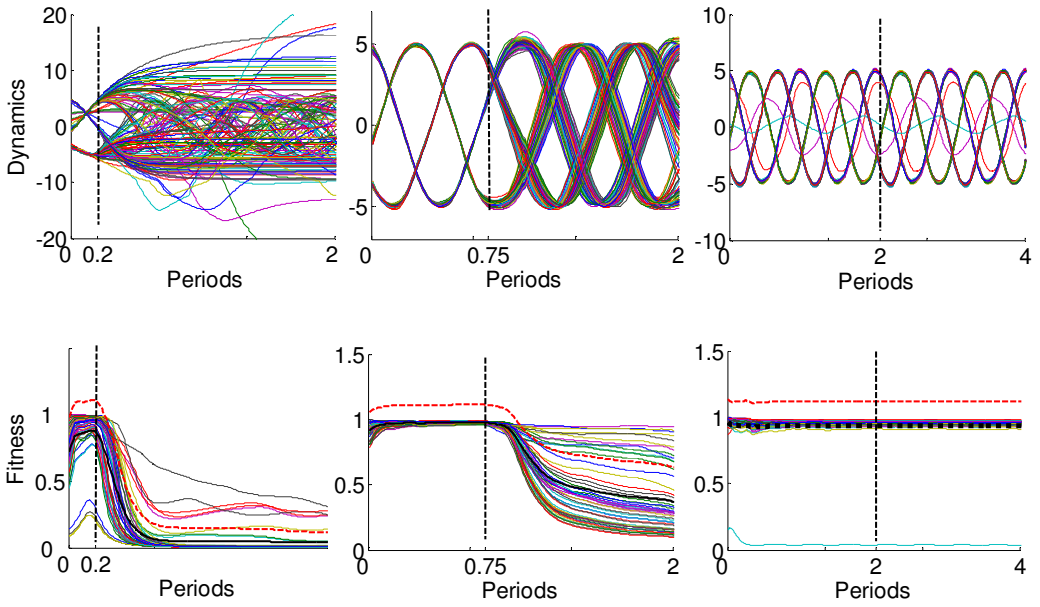


Figure 4.10 Long term simulation dynamics of the TDRNN-NP model inferred from three different time intervals (from 0 to the vertical dashed line).On the left side panels over-fitting is present. On the middle panels de-synchronization occurs for the majority of the models after the optimized interval. In the right side panels a stable long term fitting is observed for the majority of the models. However, some failures to fit the amount of data exist.

While all solutions fit the data for 0.2<T<0.33 (figure 3.12 left panels), only few solutions show the desired oscillatory behavior for long times. This could be corroborated with the instantaneous fitness in the left lower panel of figure 4.10. Simulating and fitting the system for 0.5<T<0.75 (figure 3.12 middle panels) all solutions show stable oscillatory behavior, yet with different oscillation periods, as they start to desynchronize for long times. Again, here it is easy to corroborate with the respective (lower panel) graph of the instantaneous-fitness, where after the optimized period a broad spectrum of similar frequencies is depicted. Finally, for T>1 (figure 4.10 right panels), two solution groups were found: one group of solutions showed almost perfect agreement between model simulation and synthetic data, with a stable long term instantaneous-fitness while the second group showed significant discrepancies in the inferred amplitude and frequency of the oscillations also showing a very poor instantaneous fitness.

Taken together, all these results demonstrate that the TDRNN model could infer the goal topology from as little information as just one third of the cyclic repressilator model with just one error, approx. 90% of accuracy of correctly predicted weighted and directed edges, outperforming the CTRNN model in terms of congruence with the goal network.

4.1.3 Robustness against noise

To determine the influence of measurement errors on the reverse engineering process, I added Gaussian distributed noise to the time series data. I equidistantly sampled the repressilator sine functions at ten time points over a time interval of two cycles and added noise with a standard deviation $\sigma = sI$, where I and s denote the amplitude of the sine wave and a proportionality factor, respectively. The latter is used to define the noise strength in subsequent experiments. To assess the performance of different interpolation approaches under the influence of noise, I interpolated the sampled data using a linear and cubic spline interpolation for s set as 20%, 30%, 40%, and 50%, respectively. The new datasets were then reverse engineered to investigate the relationships between measurement noise, data interpolation, the model functions and the use of the parameter pruning resulting in 8 models under four noise conditions in total. On figure 4.11 are the boxplots from the distributions of the fitness of these 8 groups.

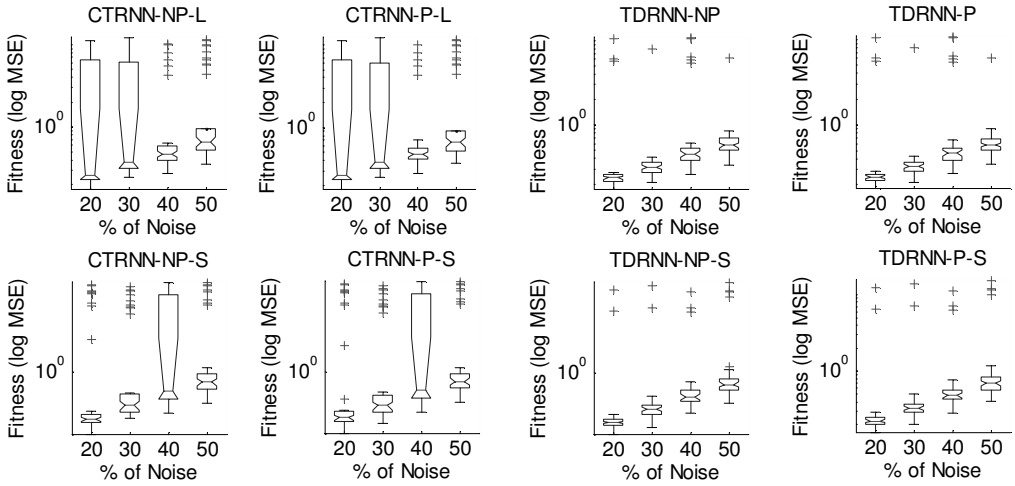


Figure 4.11 Data fits comparison among the two models (TDRNN and CTRNN), pruning (P) and not pruning and using linear (L) or spline (S) interpolation giving a total of eight groups**.** Distribution outliers are represented by a "+" sign.

The boxplots of the fitness from the optimization runs on figure 4.11 show that the TDRNN models have less outliers (+ signs) than the CTRNN models, meaning that the TDRNN models fail less often to fit the data. Comparing between the intergroup medians, I found in all 8 groups, that the more the noise the worst (higher MSE) is the fitting of the data. On the other hand, comparing corresponding intra group medians among models, the TDRNN has always a better fitness than the CTRNN model. Moreover, without the outliers, the distributions of the fitness (MSE) in all the TDRNN model groups follow a normal distribution while the respective fitness distributions from the CTRNN do not. Therefore, since the CTRNN shows too many problems to fit the data, in order to compare among the media of both models and interpolation approaches, a double filtering was applied.

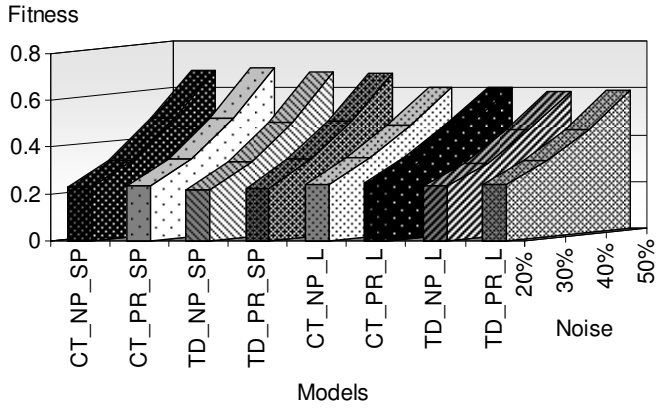


Figure 4. 12 Inference power comparison among the eight groups. Linear spline interpolation performs better that spline interpolation for higher noise quantities

The first filter was to split those clear failure fittings with MSE >1, and the second was the previously used +/- 2 standard deviations. The result of the means of this double filtering is on figure 4.12.

On figure 4.12, it is clearly shown that besides the model utilized, in all cases the models can better fit the data when using linear interpolation instead of the cubic spline interpolation. This is particularly true for noise quantities ≥ 30%. This result is of significance since on the reverse engineering area this is an open issue (Bar-Joseph, 2004; Bar-Joseph, et al., 2004). On the normal reverse engineering workflow introduced in this thesis, the more important result is the calculus of the robust parameters that determine the network topology. In this case, the comparison of this robust parameters and its comparison to the goal network is plotted in figure 4.13 by the means of errors as mismatches between inferred topologies and goal network.

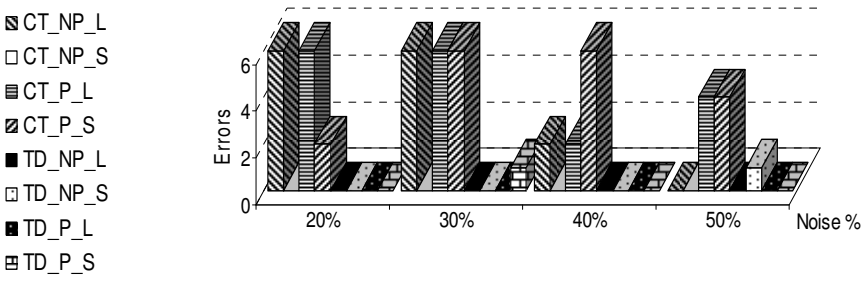


Figure 4. 13 Inference power comparison among the two models (TDRNN and CTRNN), pruning (P) and not pruning and using linear (L) or spline (S) interpolation giving a total of eight groups**,** under different strength of Gaussian noise.

The direct result from the robust parameters calculation depicted in figure 4.13 shows that the TDRNN is by far more robust against noise than the CTRNN model. Actually the TDRNN models using linear (L) interpolation present no errors regardless of the quantity of noise *s* in the data and the use or not of the pruning function (NP and P). The same TDRNN models (NP and P) using spline interpolation showed one error each for $s \geq 20$ %. Interestingly the CTRNN model using spline interpolation and not using the pruning function, showed no errors regardless of the presence of noise. By contrast, the rest of CTRNN models present several errors for all quantities of noise.

Finally, analogous to the analysis performed in the length of data section, individual-errors per run were calculated to correlate the fitness (MSE) of the runs with their respective individual-errors. Again, this was achieved through the ternary discretization of individual run's matrix of weights and comparison to the goal network.

After filtering for 2 standard deviation ≥ intra group mean of MSE, the CTRNN model still has no normal distributions of its MSE runs. Therefore, Spearman's (none parametric) correlation was performed over the filtered (to avoid for spurious correlations) MSEs of the models and the individual-errors of every run. The results are plotted on figure 4.14.

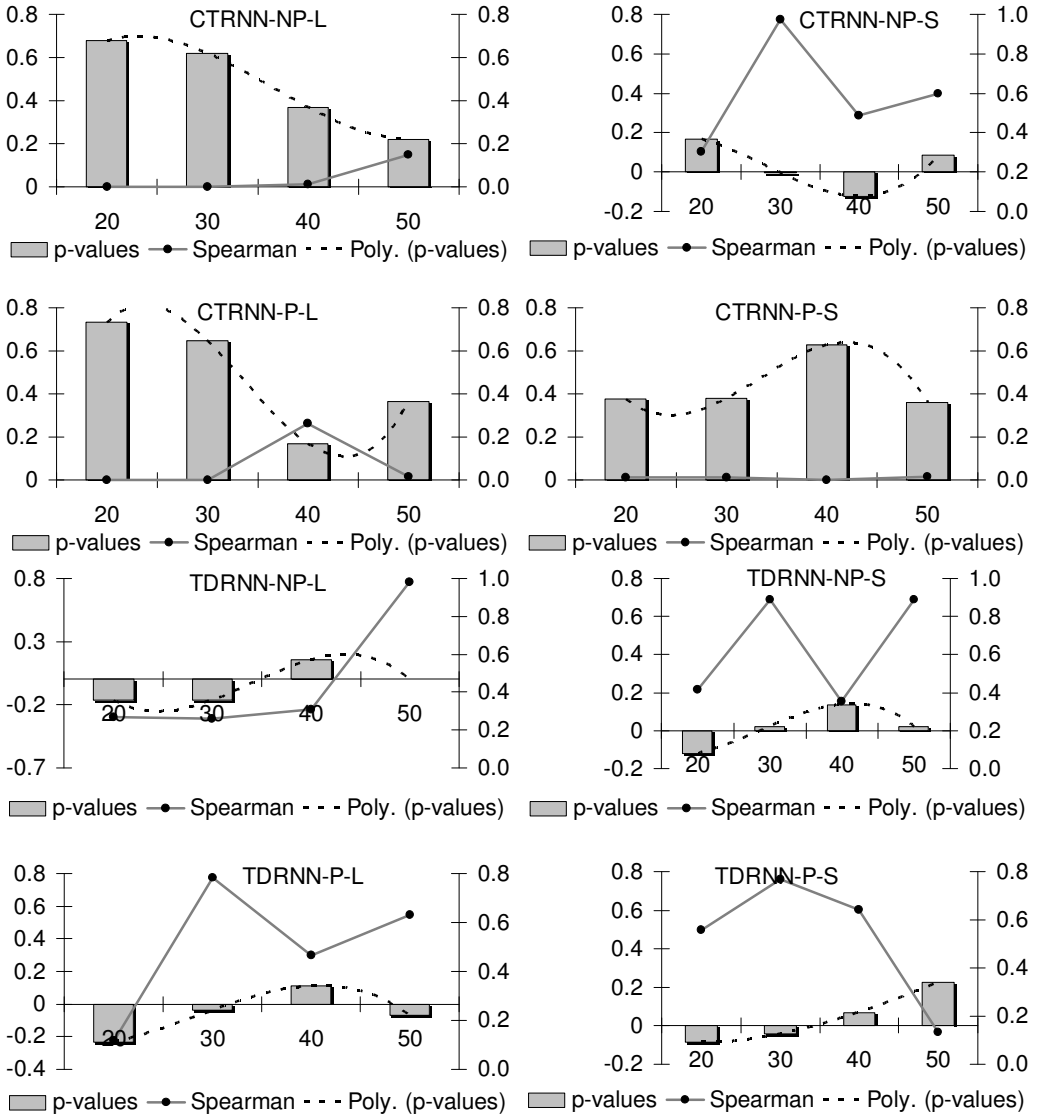


Figure 4.14 Spearman correlation between the individual errors and individual fitness. Comparison among the eight models groups under different strength of noise, correlation trends are in dashed lines.

After filtering for outliers, figure 3.16 clearly shows that those models showing no or a weak correlation between their MSE distribution and individual errors, are exactly those with no errors encountered by the robust parameter calculation technique (TDRNN NP-L, TDRNN P-L and CTRNN NP-S) depicted in figure 3.15. This result could be seen counterintuitive at first glance, but bear in mind that since the errors are of discrete nature and the fitness (MSE) of continuous nature, therefore small variations on the fitness (MSE) do not correlate to the errors because they could have no variation at all (zero error zones). Therefore, only strong correlations with low p-

values are significant and positive ones are associated to fitting problems while negative ones to over fitting problems. However, usually one could not know the goal topology in advance and these results are useful just to identify that fitness alone is a bad indicator of the inference power of a model.

4.1.4 Robustness against incomplete information: Clustering improves the standard reverse engineering task, quantitatively and qualitatively

On this section, the model is evaluated for the common situation when the network under consideration is actually larger than the number of genes selected for reverse engineering (see Methods, data selection). In order to elucidate the effect of incomplete information on the modelling procedure optimizations using three-node CTRNN and TDRNN models were performed, while only the fitness in one or two of the three nodes was measured.

The distributions of the fitness (MSE) and the individual-errors for this experimental set-up are plotted in figure 4.15 trough boxplots. As expected, (figure 4.15 lower panels) an increase in the number of individual-errors was found for both models with a reduction of information. For the one-node-case all four models (TDRNN, CTRNN both using and not pruning) were unable to properly infer the goal network, showing more than four out of nine possible errors. Consistent with this observation, a reduction in the number of nodes to be optimized increased the computed fitness as shown in upper panels of figure 4.15 while they decreased in the sense of the MSE.

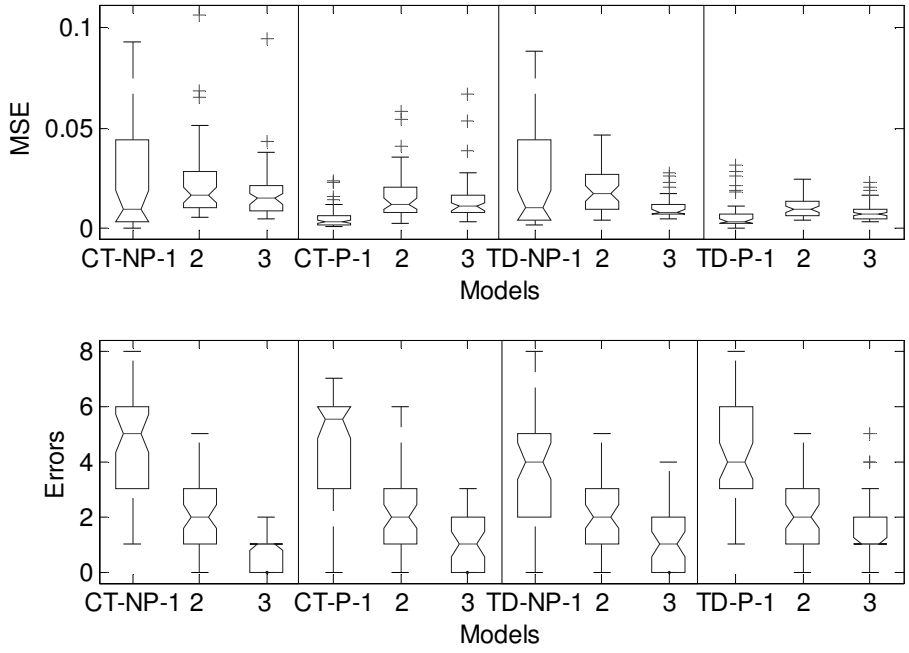


Figure 4.15 Models fits comparison between the TDRNN and CTRNN using (P) and not (NP) the pruning function, along different proportions of incomplete information represented by the number of nodes optimized: 1 = two nodes not optimized, 2 = one node not optimized and 3 all nodes were optimized.

Concerning the calculus of the number of errors through the identification of the robust parameters approach, a similar tendency was found, here depicted in figure 4.16 Even though the TDRNN models have significant less number of errors for the case of two optimized nodes, in general all models failed when just one out of three nodes were optimized.

As mentioned, this case of incomplete information occurs very often in the RE of GRN area. Therefore, it would be of high interest for the community to increase the inference power of any model for such a situation. Hence, here I performed an additional analysis with the objective of improving the inference power of my model.

In the two previous sections, it was shown that different network topologies are inferred along the reverse engineering process (Figures 4.10). To systematically analyze these distributions, the total sets of different parameter solutions were clustered. For this purpose, the matrix of weights of every optimization run was represented as a vector.

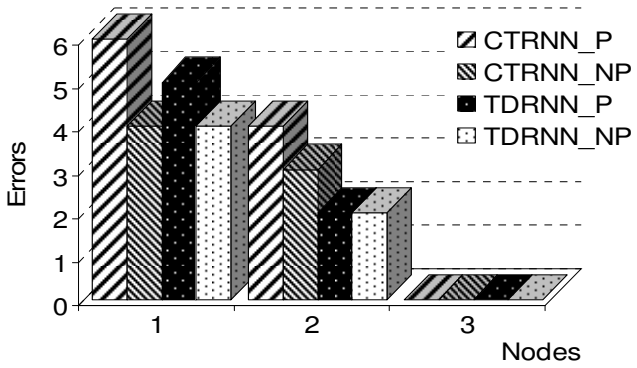


Figure 4. 16 Inference power along different quantities of missing information. 3 = no missing information

The next step was to identify the ideal number of clusters to split the entire set of solutions. Therefore, hierarchical and SOM clustering (see chapter 3 Methods) were applied over the sets of 50 vectors per model and where determined five as the number of naturally formed clusters. Then I used the k-means algorithm to split the 50 optimization runs on each of the four models. Finally, I recalculated the robust parameters to obtain the adjacency matrix for each cluster.

The first and more important result of this section appears for the analysis of the clusters of the different models when the optimization of just one node was performed. The following results are related only to that case. Additionally, since five different possible solutions per model were obtained, I needed a way to differentiate among them. Therefore, the inference power of every cluster was calculated as the ratio between its size and its MSE mean fitness. The upper panels (a-e) of figure 4.17 depict the five clusters of the TDRNN-NP model, the mean (over cluster row in colour scale) and the standard deviation (over the mean row in grey scale) of every cluster and their inference index. The respective weighted directed graph is obtained from the robust parameters calculations from the last two parameters (see chapter 3, Methods) and it is placed below their respective cluster. On figure 4.17, on the lower panels (f-j) are depicted the analogous results for the CTRNN-NP model. The colour scale of every cluster and their mean is depicted in figure 4.17 k.

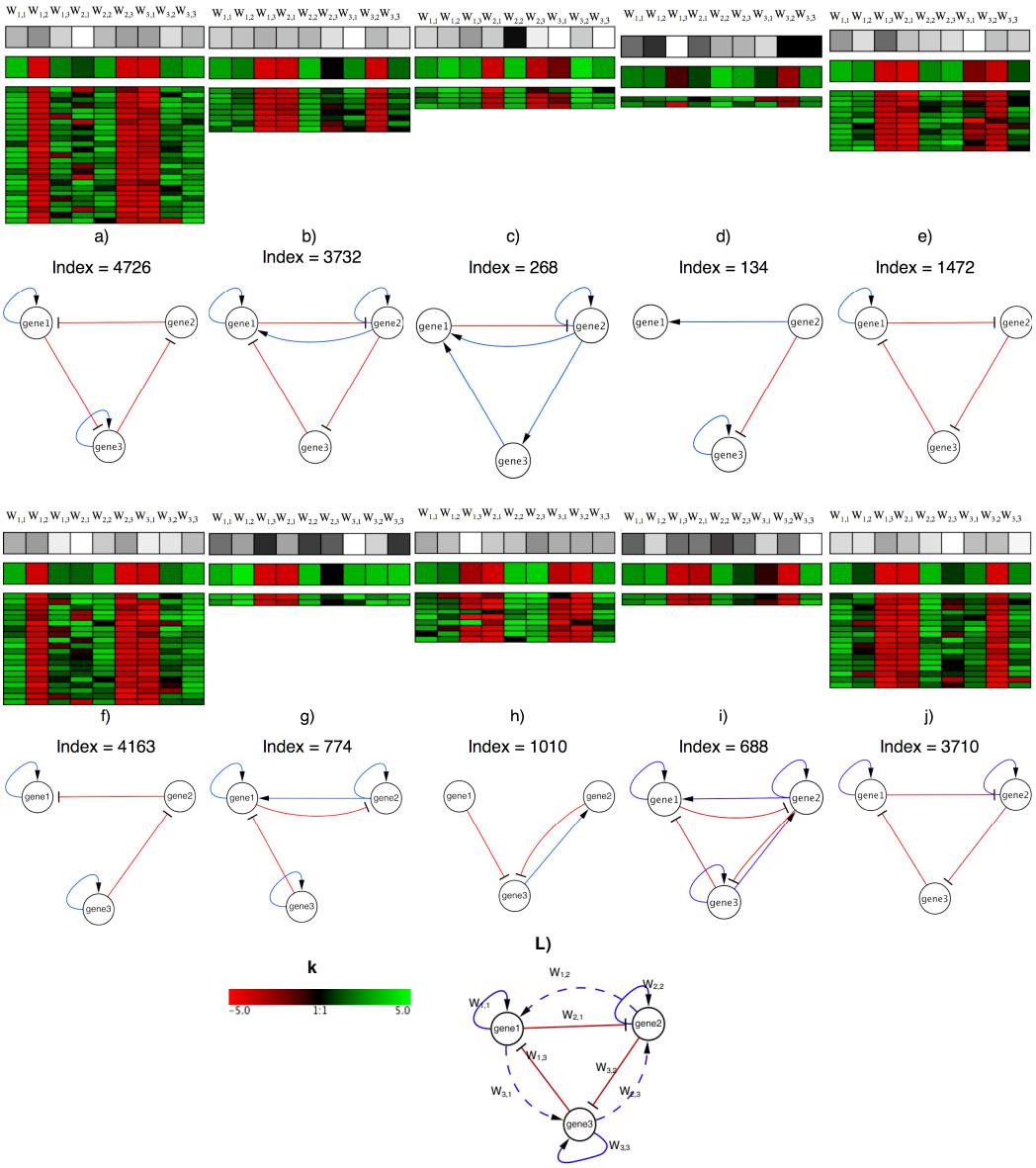


Figure 4.17 On top of a, b ,c ,d and e are clusters representing groups of weight parameter solutions between the three nodes from the TDRNN-NP model. Every row is a vector representation of the matrix of weights ($W_{i,j}$) from each of the 50-optimization runs from this model. At the top of ever cluster are the columns heads for mapping positions (W_{ij}) of these vectors into the matrices of weights, below these column heads are the standard deviation (grey scale) and the mean value (color scale) of every column. On k) is placed the weights color scale and below in l) is a scheme with the map of weights positions representing the repressilator (blue dashed lines does should not exist, showed just for mapping purposes). With the mean and standard deviation, the z-score is calculated and those robust parameters (z-score > 2) determine the edges on the lower graphs. Networks b) and e) are closely related. Those networks and network a) are topologically equivalent solutions to the repressilator, with the rotation direction being the only difference between them. To differentiate between the four architectures encountered (a, b + e, c and d) we calculated the ratio between number of elements and mean fitness per cluster and named it the cluster index. Those with the highest index are the two repressilator-like architectures. On f, g, h, i and j, are the analogous clusters from the CTRNN model and below them, their respective inferred networks. Here i and j are related solutions. The network on a, is the counter wise repressilator solution, despite the cyclic behavior is incomplete

While performing a visual inspection of the formed clusters, an unexpected feature emerged. For the entire repressilator study, a counter clockwise architecture as the goal network was considered (see figure 4.1a). However, while optimizing just one of three nodes this restriction banishes because only the sinusoidal dynamics of one node is important. In this way, the equivalent cyclic repressilator architecture functioning clockwise is also a valid topology which is depicted in figure 4.17-L. Taking this second architecture into account, it is easier to explain the fact that the three clusters with the largest index (> 1000 units) from the TDRNN-NP model (clusters *a b* and *e* from figure 4.17 upper panel), have an equivalent repressilator graph among them. Cluster-graph *a* is the counter clockwise repressilator architecture while clusters-graph *b* and *e* represent the second valid clockwise repressilator architecture depicted in 4.17-L.

Performing an analogous analysis for the CTRNN-NP model leads to the graphs f-j from figure 4.17. However, notice that here are only two clusters with an index > 1000 units and that the model failed to find the cyclic topology of the counter clock wise repressilator (figure 4.17f). For the other two models, as well as for the previous ones, the results are summarized on table 4.2. However, in table 4.2, I additionally compared some other properties of the directed graphs derived from the clusters as the number of false positives, false negatives and if a cyclic topology of any repressilator were found. As it can be seen, in all four models the cluster-graphs with the higher index (bold fonts) are those related to any of the two repressilator architectures. Taking only these solutions into account, all the models decrease their number of errors dramatically.

Table 4. 2 Comparison among the different clusters and networks from the two different models TDRNN and CTRNN using (P) and not (NP) the pruning function

MODELS		*Clust. 1*	*Clust. 2*	*Clust. 3*	*Clust. 4*	*Clust. 5*
CTRNN_NP	Errors	**2**	2	5	2	**1**
	F. positives	0	1	1	2	0
	F. negatives	2	1	4	0	1
	Cluster size	**20**	2	9	2	**17**
	MSE mean	0.00	0.00	0.01	0.00	0.00
	Cluster index	**4163.59**	774.29	1010.11	687.99	**3710.31**
	CYCLE	**NO**	NO	NO	YES	**YES**
CTRNN_P	Errors	**3**	4	4	8	**3**
	F. positives	0	1	1	3	0
	F. negatives	3	3	3	5	3
	Cluster size	**14**	10	7	3	**16**
	MSE mean	0.01	0.05	0.05	0.04	0.01
	Cluster index	**1828.78**	199.85	129.00	78.17	**2373.65**
	CYCLE	**YES**	NO	NO	YES	**YES**
TDRNN_NP	Errors	**1**	**2**	7	5	**2**
	F. positives	0	1	3	1	0
	F. negatives	1	1	4	4	2
	Cluster size	**25**	**8**	4	2	**11**
	MSE mean	0.01	0.00	0.01	0.01	0.01
	Cluster index	**4726.47**	**3732.65**	268.00	134.39	**1472.27**
	CYCLE	**YES**	**YES**	YES	NO	**YES**
TDRNN_P	Errors	4	**2**	5	5	**2**
	F. positives	1	0	1	1	0
	F. negatives	3	2	4	4	2
	Cluster size	5	**15**	12	9	**9**
	MSE mean	0.01	0.01	0.05	0.05	0.01
	Cluster index	639.78	**2451.19**	246.57	181.61	**1137.29**
	CYCLE	YES	**YES**	NO	NO	**YES**[6]

F positives = false positives F negatives = false negatives
CYCLE = cyclic inferred network

Summarizing, applying the cluster analysis, a decrease in the error incidences from 4 to 1.5 was found for both CTRNN-NP and TDRNN-NP models. However, only for the TDRNN-NP model was found three different network topology solutions with oscillatory behavior. These clusters with the highest inference power constitute the topologically equivalent repressilator architectures with clockwise or counter-clockwise cyclic repression (see figure 4.17 b, e and a respectively).

Clustering solutions of the CTRNN-P and TDRNN-P models decreased the error incidences from 6 and 5 (see figure 4.16) to 3, respectively (see table 4.2). Both models were able to find the two repressilator architectures. Repeating the same cluster analysis for reverse engineering with two nodes all models succeeded to infer

the desired repressilator topology without errors. This demonstrates clearly that the parameter clustering approach improves the model's inference power.

4.2 The yeast cell cycle

To assess the predictive power of the TDRNN as a gene regulatory network model and the inference power of the clustering extended reverse engineering workflow for biological systems, in this section, I compared the performance of the TDRNN with that of a CTRNN and a Dynamic Bayesian Network to infer the well studied (Chu, et al., 1998; Futcher, 2002; Gavin, et al., 2006; Guelzim, et al., 2002; Harrison, et al., 2007; Ihmels, et al., 2002; Krogan, et al., 2006; Murray and Beckmann, 2007; Tsai and Lu, 2005) transcription-signal transduction cell cycle network of *Saccharomyces cerevisiae* based on experimental data.

The goal network

On their work, Li et al. (Li, et al., 2004) proposed a cell cycle network model of *Saccharomyces cerevisiae* having 11 nodes that comprise 18 genes, proteins and black boxes. The functional network elements are divided into 4 groups: cyclines (Cln 1,2 and 3), inhibitors degraders and competitors of the cycline cdc28 (Sic1, Cdh1, Cdc20/14), transcription factors (SBF, MBF, Mcm1/SFF, Swi5), check points and self repressions (cell size, DNA replication), see figure 4.18a. There are 34 interactions, out of which 19 are positive and 15 negative. Note that five negative interactions are of unknown nature and were added in order to make the network functional in logical terms.

The data source, selection and quality control

In this section, the gene expression kinetics of the yeast cell cycle from the alpha-factor data set was used (Spellman et al., 1998). Utilizing four different techniques to synchronize the yeast colonies they generate four data sets named; cdc28, cdc15, alpha factor, and elutriation. Unfortunately, other works (Fellenberg, et al., 2001) has shown that desynchronization occurs on the cdc15 and elutriation data sets; therefore

we do not consider them. After analyzing the left two data sets, we discarded the cdc28 data set because the high frequency of missed data points on our selected genes. Therefore, I end up with just the alfa factor data set. After performing the quality control described on methods, no particular issue was detected for the selected data set.

However, inferring the cell cycle network proposed by Li et al from this transcriptomic data is challenging, as only four of the eleven network nodes are transcription factors and it includes 2 dimers (SBF, MBF) and 5 nodes that are represented by 2 proteins: Clb 1/2, Clb5/6, Mcm1/SFF, Cdc14/20 and Cln1/2. Fortunately, the genes encoding the dimers and the redundant proteins exhibit similar expression kinetics so that I represented these nodes by the mean expression of their respective genes.

Data interpolation

I tested two different interpolations approaches for this data set: linear interpolation and a B-spline interpolation method suggested by Bar-Joseph (see Methods) for the continuous representation of gene expression.

Models

The data was fitted using the TDRNN and the CTRNN models choosing the parameter ranges for $[\tau,\theta,\vartheta]$ to $[10-55,0-1,0-3.5]$. For the TDRNN, an optimization time delay range equivalent to 10 minutes was chosen as a biological compromise between the fast signal-transduction and the slow transcriptional responses.

4.2.1 TDRNN shows superior inference and predictive power than previous models on experimental data

Here, I compared the TDRNN performance on the cell cycle data with CTRNN and Dynamic Bayesian Networks (DBN) as described in chapter 3. The quantitative comparison of the different solutions was performed by computing the adjacency

matrix for each group of model solutions and calculating the cost of transforming the resulting directed graph of every model group into the goal network.

As an objective transformation cost function, a directed-weighted version of the graph edit distance algorithm (GED) was used. This algorithm calculates the cost to convert one graph into another by changing edge weights and/or directions. Additionally, it accounts for shortcuts (the deletions-insertions of nodes) from the semantic of directed, weighted graphs. The results for the GED together with other graph measures are depicted in table 4.3

Table 4. 3 Mean square errors (MSE) and graph edit distance from inferred networks using the CTRNN, Dynamic Bayesian Network and TDRNN models. The MSE is averaged over 50 optimization runs. The columns two and three denote the number of nodes and edges for which robust interactions were inferred. Columns four and five show the number of correctly and falsely predicted interactions, which column 6 shows the GED costs. The rows with label Cluster 1-4 show the MSE and the GED for the individually clustered solutions from 50 independent optimization runs for the TDRNN-NP model. The Bootstrapping results for model are given in the bottom row

Model	MSE	Nodes	Edges	Positives	FP	GED	Index
CTRNN	0.98	3	2	0	2	134	
Dynamic Bayesian network	NA	11	34	10	24	119	
TDRNN B-Spline	0.71	9	9	2	7	107	
TDRNN-P	0.36	10	18	8	10	80	
TDRNN-NP	0.37	11	19	8	11	68	
Cluster 1	0.36	11	21	9	12	62	50
Cluster 2	***0.36***	***11***	***26***	***15***	***11***	***41***	**28**
Cluster 3	0.37	11	24	9	15	61	22
Cluster 4	0.39	11	25	11	14	70	20
Bootstrap	GED Mean =99, GED Standard deviation = 14[7]						

Nodes = number of nodes with at least one edge Positives = number of correct inferred correlations among two nodes F P = false positive edges GED = graph edit distance cost

According to this analysis, the TDRNN-NP model using linear interpolation has the best inference power of all models having the lowest GED costs. The results depend crucially on data interpolation where this model shows also the best predictive power exemplified in figure 4.18 and to a lesser extent on parameter pruning, changing the GED cost by 64% and 23%, respectively.

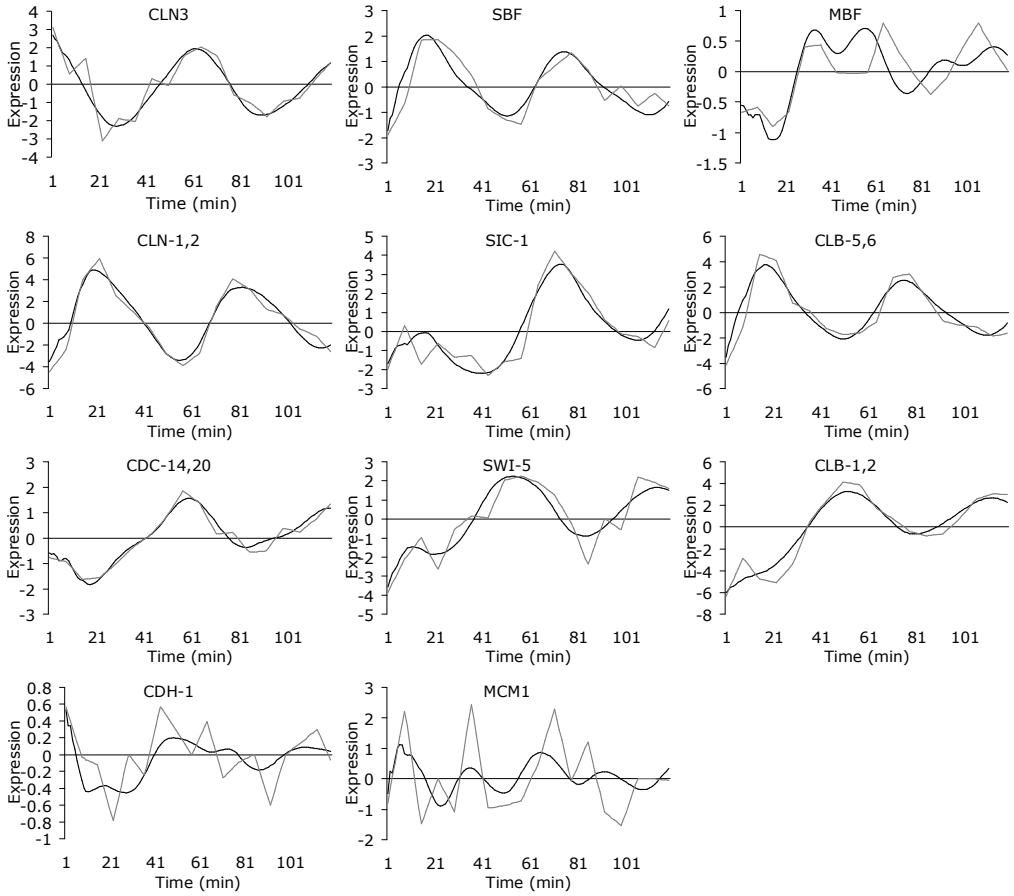


Figure 4.18 Data fitting example from one of the TDRNN runs. In straight lines are the original linearly interpolated data; in smooth lines are the approximation of every node of one model for their respective gene expression. The global fitness of this average run was MSE= 0.36

Notice that on figure 4.18, the approximation of most of the gene dynamics is acceptable. Only for the MCM1 gene, the approximation presents some discrepancy to the data (straight line). The reason for this behavior is that the data for this gene presents some missing data point. Therefore its pattern appears almost randomly changing from active to inactive. This is a limitation from the data set and therefore, it was not considered necessary to improve its fitting.

4.2.2 Bootstrapping validation

To validate the TDRNN-NP results for the yeast data, a bootstrapping test was performed by randomly shuffling the order of the microarray time series data 50 times and repeating the entire workflow. Then, the GED cost of each of the 50 adjacency

matrices obtained was calculated. The resulting GED costs were normally distributed as expected (see figure 4.19). Additionally, the Lillie test of normality was performed. The mean and standard deviations of the bootstrapped optimization runs are shown on the bottom row in table 4.3. Notice that the mean of the GED cost from this bootstrapping deviates more than 2 standard deviations in comparison with the TDRNN-NP model. This confirms that the results from our TDRNN-NP model were not obtained by chance which further validates the correlation between the inferred network topology and the proposed cell cycle network.

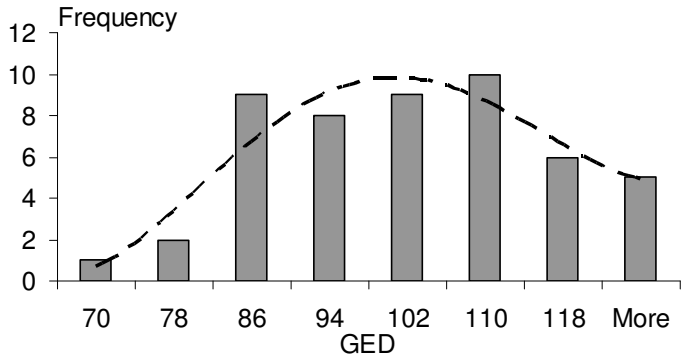


Figure 4.19 Bootstrap data histogram. The distribution of the cost to transform networks, inferred from randomizing the original data, into the goal network is normal with mean = 99 edition units.

4.2.3 Clustering improves the RE process with real data

After performing the clustering approach described in the section 3.1.4, I found that the two clusters with the highest index (clusters 1 and 2) decreased the GED cost from 68 to 62 and 41, respectively (see table 4.3 and fig. 4.20b). The GED cost from cluster 2 deviates by more than four standard deviations from the mean of the bootstrapping test, i.e. the result is unlikely to be obtained by chance.

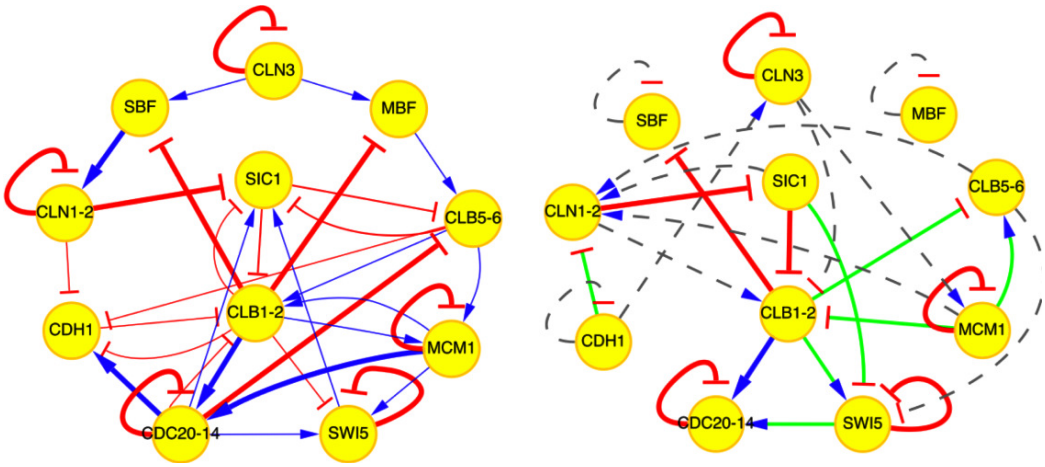


Figure 4.20 On the left side, the original Yeast cell cycle network from (Li et al., 2004), where blue and red arrows mean activation and inhibition, respectively. The indispensable edges according to the work from Stoll et al. are on thicker lines. On the right side, is the network topology inferred by our TDRNN model, (cluster2, of the yeast results on table 2). Red and blue thicker lines (8) are correctly inferring directed weighted edges, green lines (7) denote matches with reported correlations between two nodes on the goal network (direction and/or weight are not properly inferred on this edges) and grey dashed lines are false positives (11). Seven of the 8 properly inferred weighted directed edges, belongs to the 13 indispensable ones reported by Stoll while none of the misdirected/weighted falls into this category.

Comparing the inferred network from cluster two (figure 4.17b) with the cell cycle goal network (figure 4.20a), I found that 7 out of 8 weighted and directed edges were predicted correctly according to the 13 edges defined by Stoll (Stoll and Rougemont, 2006) as being the only necessary interactions for this yeast cell cycle circuit. The low graph editing costs of the best inferred network, i.e. cluster 2, demonstrates that the TDRNN model together with the clustering of reverse engineered solutions is able to extract biologically meaningful information from a combined protein signalling and gene regulation network by just using one experimental time series (44% [15/34] of the total correlations, or 54% (7/13), if considering the indispensable weighted directed edges).

4.3 Reverse engineering of keratinocyte-fibroblast communication

In this section, I present a case study of an unknown network. This scenario is a real situation where there is little information about the structure of the function one is interested in to understand its dynamics through reverse engineering. Along this case, the assumptions, limitations and improvement opportunities that offer such experimental scenario will be explained. Probably, the more important gain from this case scenario lies in making the experience of how to direct a data-driven experimental setup as well as how to avoid certain problems in the iterative experiment-modeling process.

The unknown goal network

The idea behind this experimental setup was to study the communication between two cell lines, keratinocytes and fibroblast. This interaction is important for processes like skin wound healing and some authors (Birchmeier, et al., 2003) have even suggested that it is related to cancer by the means of certain analogies. In particular, the process here studied is related to cell migration and its cancer analogous: metastasis.

To study this interaction, DNA microarray experiments at several time points upon hepatocyte growth factor (HGF) stimulation was performed by a cooperation partner group (Axel Szabowski), to obtain the gene expression kinetics from heterogonous co-cultures containing primary human keratinocytes and murine cjun-deficient fibroblasts. The latter were chosen trying to discriminate between human and murine mRNA, based on species-specific sequences and to provide an HGF-free background. This is a strong assumption; therefore a detailed quality control analysis will be explained for this data set.

In the global experimental setup keratinocytes were stimulated with HGF, which induces both proliferation and migration (Birchmeier et al, 2003). Three additional experiments using keratinocyte growth factor (FGF-7), granulocyte–macrophage

colony-stimulating factor (GM-CSF) and stromal derived factor-1 (SDF-1) as stimuli were conducted, all of them inducing cell proliferation, but not migration (Florin, et al., 2004). The particular experimental details are described below in order to explain the analysis and the obtained results, but the experiments where performed by the collaboration partner group.

Cell culture

Normal human skin keratinocytes and dermal fibroblasts (HDF) were derived from adult skin (Stark, et al., 1999). HDF obtained from the outgrowth of explant cultures were grown in Dulbecco's modified Eagle's medium (DMEM; Bio Whittaker) supplemented with 10% fetal calf serum (FCS), and cells from passages 4 to 8 were used. Mouse wild-type and cjun-/- fibroblasts were isolated from mouse embryos and immortalized according to the 3T3 protocol (Schreiber, et al., 1999) and used together with HDF as feeder cells. Normal human skin keratinocytes were plated on X-irradiated feeder cells (HDFi, 70 Gy[8]; MEFi, 20 Gy) in FAD medium (DMEM/Hams F12 3:1) with 100 U/ml penicillin, 50 mg/ml streptomycin and supplemented with 5% FCS, 5 mg/ml insulin, 0.1 ng/ml recombinant human EGF, 0.1 nM cholera toxin, 0.1 nM adenine and 0.4 mg/ml hydrocortisone (Sigma). For expression profiling, total RNA of co-cultured cells was isolated 1, 2, 3, 4, 6 and 8 h after stimulation with recombinant human cytokines (10 ng/ml HGF, 10 ng/ml GM-CSF, 10 ng/ml FGF-7 or 10 ng/ml SDF-1; all obtained from R&D Systems).

Migration assay

Immortalized human keratinocytes (HaCaT cells) were cultured in monoculture with DMEM (10% FCS, 100 U/ml penicillin, 100 mg/ml streptomycin). Subsequently, the cell monolayer was damaged with a 'scratch' using a pipette tip and the cells were

[8] In Dosimetry, which is a scientific subspecialty in the fields of health physics and medical physics that is focused on the calculation of internal and external doses from ionizing radiation, the absorbed dose is reported in gray (Gy) for the matter or sieverts (Sv) for biological tissue, where 1 Gy or 1 Sv is equal to 1 joule per kilogram.

treated with 5 mg/ml mitomycin c (Sigma-Aldrich) 3 h before the stimulation. The cells were stimulated at the indicated time points and periods with cytokines and/or inhibitors: 10 ng/ml HGF (R&D Systems), 1 ng/ml EGF (R&D Systems), 150 nM tyrphostin AG1473 (Biomol) (EGFR inhibitor) or 1 mM GW2974 (Sigma) (EGF-R inhibitor), 50 mM meloxicam (Biomol) (PTGS-2 inhibitor), 0.5 mM H-89 (Calbiochem) (PKA inhibitor) or 10 mM myristoylated PKI (14–22) amide, cell-permeable PKA inhibitor (Biomol), 200 mM 8-(4-chlorophenylthio) adenosine 30, 50 -cyclic mono-phosphate sodium salt (Sigma) (PKA activator) and incubated for further 30 h. The Relative migratory activity was determined by measuring the migration distance during the culture by using standard protocols.

In general, the experimental setup has three strong issues: the first is the fact that the cells were not synchronized in order to perform the measurements of they transcriptomic response after stimulation. This is crucial since the measurements are done over populations of cells, and it should represent the average behavior of the population. The other two are described on the next paragraphs.

Microarray data acquisition and analysis

Microarray measurements were recorded for four different stimuli from co-cultures, namely HGF, FGF-7, SDF and GM-CSF. For each stimulus, within one experiment six probes were taken (time points of 1, 2, 3, 4, 6 and 8 h after initial system stimulation) and further analyzed. Total RNA was isolated, labeled and hybridized to HG-U133-2plus (Affymetrix) according to the manufacturer's protocol. Raw microarray data were processed using the R environment (R Development Core Team, 2007) and the Bioconductor toolbox (Gentleman, et al., 2004). The Probe annotation was handled with the Bioconductor package hgu133-plus2cdf (Bioconductor Project). The Normalization was performed using variance stabilization available in the Bioconductor package vsn (Huber, et al., 2002). The gene fold expression was calculated according to the mean gene expression of two control measurements of an uninduced system at 0 and 8 h.

The second important issue from that experimental set-up appears in this section, where microarray measurements were performed over the mixture of co-cultured cells: Immortalized Human keratinocytes, irradiated Human Fibroblasts and irradiated Mouse Epidermal Fibroblast. Hence, cross hybridization occurs among different mRNA expressed by these cells. Moreover, there is no information on how strong and for which genes this cross hybridization could be.

Finally, the third important drawback of the experimental set-up resides on the lack of measurements of the errors and/or noise associated to the previous two drawbacks. In other words, there are no replications for the experiments because they are not reproducible (Szabowski, personal communication). As mentioned, despite repetitions could be expensive tedious and with low information content from an experimentalist point of view, they are of central relevance to develop models of biological systems.

Despite these three drawbacks invalidate the data for any scientifically based result, here will be shown as an example, what could be done with such data and some suggestions about how to improve future works that could face similar problems.

Quality control

As described in Methods, the first step to start the reverse engineering is to asses the quality of the data and to identify possible issues (Wilson and Miller, 2005). Figure 4.21 depicts the results of the preformed quality control of the keratinocyte arrays as described in Methods. The interpretation in this case, is based on the following explanations but, since these microarrays and quality control standards are thought for a single homogeneous cell line experiments, biochemical interpretation will be added when considered pertinent.

The figure is plotted from the bottom up, with the first chip at the base of the diagram and the last chip at the top. This corresponds to the order of the samples depicted in table 4.4.

Table 4. 4 Microarray time series sampling and input strength

Chip number	Hours	[µg/µl] HGF
b26	Control 8h	
a1.	Control 1h	0.5
2.	HGF 1h	0.5
3.	HGF 2h	0.5
4.	HGF 3h	0.5
5.	HGF 4h	0.5
6.	HGF 6h	0.5
7.	HGF 8h	0.5

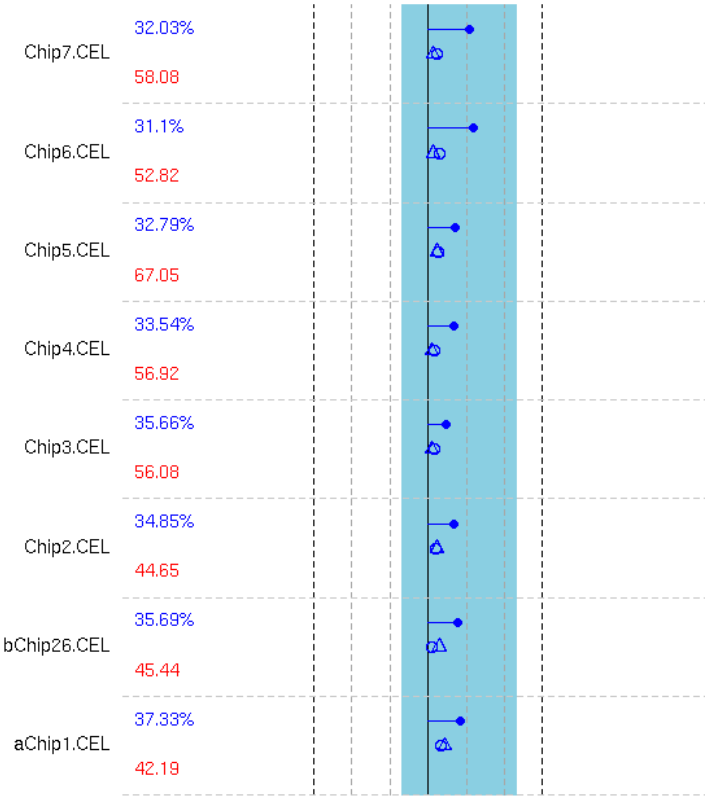


Figure 4.21 Quality control of the microarray data. Dotted horizontal lines separate the plot into rows, one for each chip. Dotted vertical lines provide a scale from -3 to 3 fold expressions. Each row shows the percentage of responding gene-probes in relation to the total gene probes in the chip, the average background, the scale factors and GAPDH / β-actin ratios for an individual chip.

The first indicator of the quality control pipeline is the average background; it is at the left of figure 4.21. The variation among the different chips is bigger than 10%,

therefore they are colored red. This parameter is usually more related to experimental problems, such as different concentration labeling or different efficiency of hybridization cocktails. Here, a stronger variation is found at the chip number 5, having a strong average background (67 units, compared to the 43 average of the two referential 1 and 26 chips). Usually this parameter has to be analyzed together with the rest of the quality control before obtaining false conclusions; however one has to bear in mind the particularities associated to the mentioned issues. Therefore, this strong background of the chip 5 will be analyzed later.

In figure 4.21 GAPDH 3':5' values are plotted as circles. According to Affymetrix, they should be about 1. The obtained values suggest a misbalance of the different transcripts sections (see methods section). However, no issues could be concluded here.

β-actin, 3':5' ratios are plotted as triangles. Because this is a longer gene, the recommendation is for the 3':5' ratios to be below 3; values below 3 are colored blue, those above, would be red. This result indicates no early degradation issues of the probes.

The percentage of present gene-probes is listed at the left side of the figure. The variation among the different chips is less than 10%; therefore, they are colored blue. However, it is important to notice that in general, the values are very low. This could have two explanations. The first is that this is a normal result, because it is not expected that all the genes are responding to the stimuli, instead just a small fraction (around 33% of the total gene-probes). The second could be that the low percentage of responsive probes is very likely an indicator of the mixture of cells. Though this parameter is designed for homogeneous cell population measurements, the normal interpretation does not apply. Instead, since probesets are flagged marginal or absent when the PM values for that probeset are not considered to be significantly above the MM probes, the number of mismatches could vary due to cross hybridization of the mixture with genes from different cells.

Finally, the blue stripe in the graph represents the range where scale factors are within 3-fold of the mean for all chips. Scale factors are plotted as a line from the centerline

of the image. A line to the left corresponds to a downscaling, to the right, to an up scaling. If any scale factors fall outside this '3-fold region', they would be colored red, otherwise they are blue. As shown in figure 4.21, there are no issues related to large scaling factors.

Dimensionality reduction: Data selection and interpolation

As explained previously, data selection and interpolation is applied to reduce the dimensionality of the data. The first step, data selection, is crucial to find the real response of the biological system and no bias should be applied here towards a particular set of genes. However, ideally, no genes balancing the overall system should be missed. It is a difficult compromise. For this data set, it was better to rely on the expertise of the experimentalist designer because the lack of controls could be in some extent compensated by some assumptions from the area of interest. In this sense, migration is the desired phenotypic response to be associated to a selected set of genes. Therefore, genes were selected by an experimentalist expert, to amplify the knowledge of their association to the migration process. Additionally, genes were selected due to their expression profile. In this sense, the idea behind was to select genes of three different kinds: early genes, possible target genes of the previous ones and late response genes that could be target genes of the two first groups.

In this way, the selected genes were PTGS2, CEACAM1, ETS1, EGR1, JUNB, FOS, PLAU, ITGAV, ITGB6, SERPINE, LAMA3, LAMC2 and additionally, I added PLAUR in order to check for its possible role as an autrocrine feedback loop. The expression profile of the selected genes is depicted in figure 4.22.

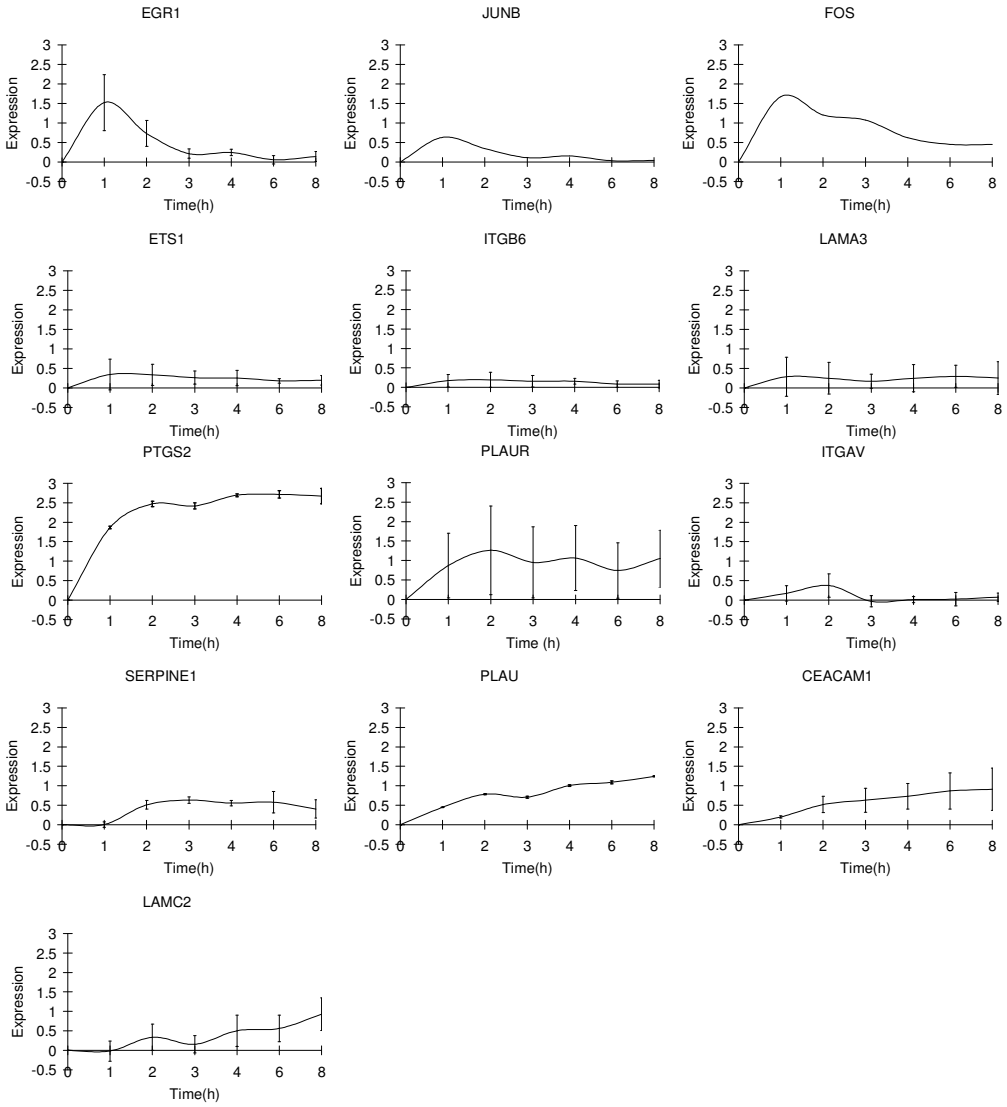


Figure 4.22 Selected data expression profile. Error bars does not represent experimental repetitions, instead represent standard deviation among probe sets of every gene at every sampling time point. Standard deviation represents the confidence of every selected gene data.

The genes in figure 4.22 are ordered by their expression profile. From top to bottom, there are from the early responding to the late responding genes. Here, it is important to notice that the standard deviation bars does not represent repetitions, instead it represents the variation of signal intensity for those genes which have more than one probe-set. Hence, this standard deviation could be used just to measure the specificity of every probe-set, but not to demonstrate the reproducibility of the experimental setup.

In this sense, genes without standard deviation bars as JUNB and FOS are represented by just one probe on the microarray chip. Even though, these two genes belong to an important family of genes with several homologous genes as the AP-1 system, there is just one probe for each member of the family. Therefore, one has to take special care, because the possibilities for a cross hybridization with the respective genes from the other cellular lines increase.

Genes like PTGS2 and PLAU exhibit a very small standard deviation among their different probes; this shows that some genes do not have cross hybridization among different cellular lines. Instead, their probe-sets are specific for them. By contrast, the rest of the genes exhibit a large standard deviation among their probe-sets. This situation is very likely due to cross hybridization.

The putative functionality of the selected genes is depicted on Figure 4.23. Notice that the early responding genes: EGR1, JUNB, FOS and ETS1 from figure 4.22 are all TF. Additionally, the apparently early responding genes from the upper panels of figure 4.22 (ITGV6 and LAMA3) are genes which encode for a cell receptor and do an extra cellular kind of Laminin proteins respectively.

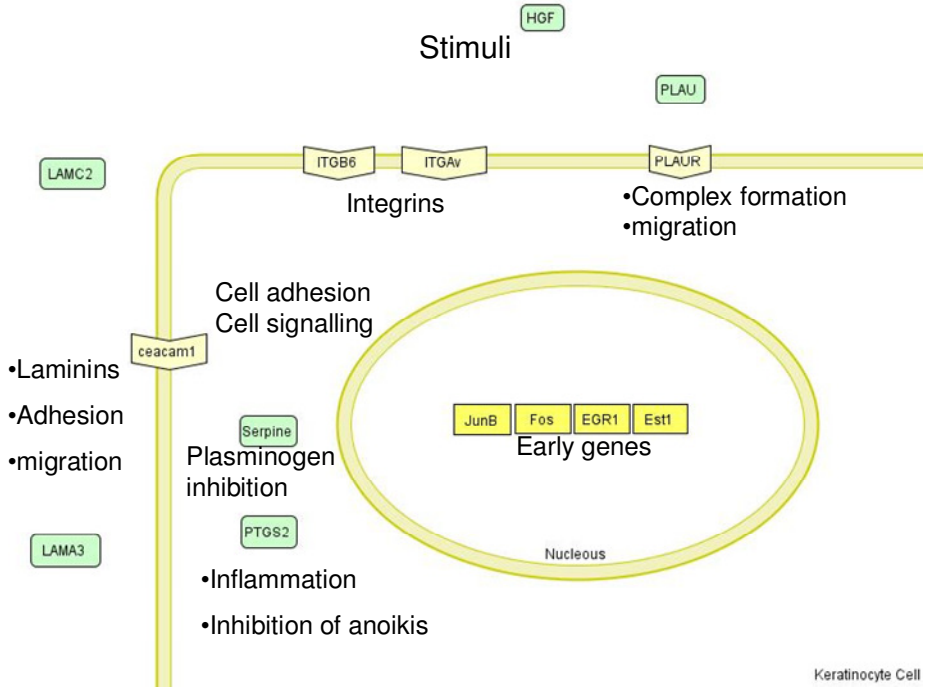


Figure 4.23 Thirteen selected genes, available information scheme.

The genes which maximum response – which happens after 2h (the middle panels of Figure 4.22)are the PTGS2, SERPINE1, ITGAV and PLAUR genes. The first two genes, PTGS2 and SERPINE1, are encoding for inhibitory proteins associated to cell signaling events, while the last two genes encode for cell receptors associated to the migration process as depicted in Figure 4.23. Lastly, there are genes which expression profile, in the lower panels of Figure 4.22 shows an increasing late response like CEACAM1, LAMC2 and PLAU. Their cellular functionality varies: CEACAM1 is a gene encoding for a protein receptor associated to processes as cell adhesion and signaling; LAMC2 is another gene encoding for a Laminin protein associated to cell-to cell adhesion and the migration process and PLAU is a gene encoding for the excreted protein PLAU that could display an autocrine feedback loop for this process.

Concerning the interpolation process, since there is no information about the associated noise, linear interpolation was utilized. However, two important aspects need to be taken into account to configure the final data set to be engineered. The first one is related to the quality control of Chip 5 and the strong background signal expression at 4 hours. A careful visual inspection of the data depicted in figure 4.22 coincides that this data point is showing very likely an artificial inflexion in almost all the selected genes. Inflexions in data are the means by which any correlation or multi regression model could associate putative interactions among genes, therefore a spurious inflexion could play a strong artificial role in the final result. By contrast, notice that the gene FOS at the three hours point shows a second wave that could not be associated to an experimental issue as demonstrated in the quality control and therefore it may remain there. Hence, the decision was taken to discard the data for the 4 hours point for the reverse engineering process as depicted in figure 4.24.

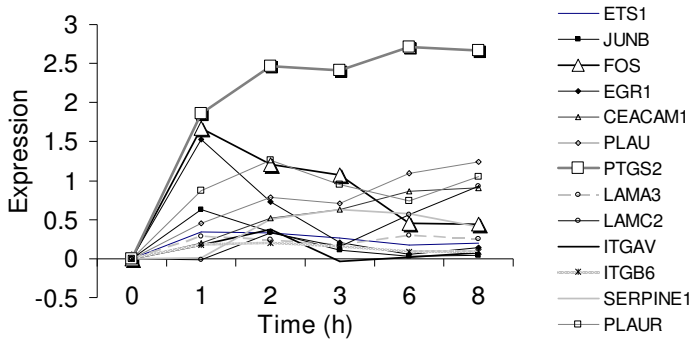


Figure 4.24 Linear interpolation of the expression profile of the selected gene data. Due to quality control issues, data of 4h were discarded. Interpolation data of the protein encoding PTGS2 gene appears to be first in reacting to the stimuli (stronger slope). The FOS transcription factor encoding gene presents a plateau between 2 and 3 h.

The second issue that could rise from interpolating the selected data is that an unexpected strong, early signal comes from some genes that do not belong to the early genes as PTGS2. Notice in figure 4.24 that the strong slope from this gene could direct the model to associate it as an initiator of the response. Hence, the importance of an adequate sampling rises because an intermediary 30 minutes point may easily show an even more abrupt response but at later time for this gene than the early genes.

Data fitting

As it can be seen in figure 4.24, the profile of the data to be fitted is relatively simple compared to the previous two experiments from sections 4.1 and 4.2. Therefore, the initial prescreening of the parameters showed that a much smaller population of only 100 individuals could be utilized to fit the data. Additionally, since the human cellular life span and the response of the selected set of genes is larger than the respective Yeast or E. coli of the two previous theoretical experiments, here a larger time delay parameter has to be utilized. However, as it can be seen from the results of section 4.1.1, an even short increment in the time delay range could have undesired results (see figure 4.2). Therefore, a maximum of 45 units' equivalents to minutes from the data set of figure 4.24 is utilized to fit this data. The selected range is of extreme importance, especially if one considers that this experimental setup includes an external input, because the fitting could be easily just driven by the external input and

when this is withdrawn, the selected functional module will withdraw its behavior, neglecting that even without strong external signals genes interact with each other at a basal level. Actually, this is the reason for the bias term and is the only source of activity for the repressilator modeled in section 4.1.

As mentioned before, this experimental setup includes the addition of HGF as a firing signal response. Therefore, here was used the equation (3.18):

$$\tau \frac{dY_i}{dt} = \sum_{j=1}^{N} W_{ij} \sigma\left(\left(Y_j\left(t-\delta_i\right)\right)-\theta_j\right)-\vartheta_i Y_i + w_i I \qquad \textbf{4. 4}$$

where the last term of the right side includes an external input *I*. However, the w_i parameter is $\neq 0$ only for those genes (represented by nodes) biologically able to receive the signaling transduced external input, in other words, only the nodes representing the TF genes receive the external signal. It makes no biological sense that all the nodes receive an input from the very beginning of the simulation runs, because it can easily impose a serious bias to the resulting network.

Moreover, the input here generated has to be the same for the selected 4 input nodes (JUNB, FOS, EST1 and EGR1) because no additional information exists. The only other possibility to change the input that I consider could be changing the intensity of the input that every one of the so-called input-nodes could receive. Therefore, the range for the w_i parameter ranges [-1,1] meaning that some genes are activated with different strength by the input and even some others could be (which not necessarily is the case) inhibited by the input. Changing the shape of every input is again a strong assumption that drives easily the results towards a desired bias. As mentioned before , this is especially true if a larger time delay is chosen, as for instance 2h., because then for the initial dynamics nodes are controlled mostly by the input. Therefore, here the general input is modeled by an exponential function giving a dynamics as depicted in figure 4.25 and as mentioned the w_i parameter was optimized to be $\neq 0$ only for the ETS1, JUNB, FOS and EGR1 genes. Notice that there is a basal input along the entire simulation runs.

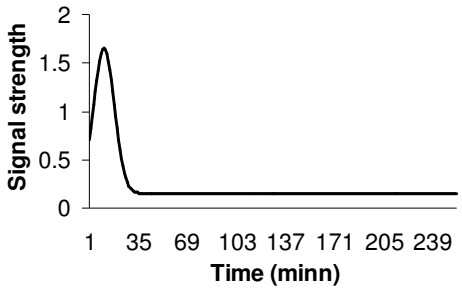


Figure 4.25 Fraction of the dynamics of the simulated external input. This simulated input represents a constant basal stimulus of 0.15 arbitrary units until the end of the optimization lapse of 480 units', equivalent to 8 hours.

However, control data fittings were performed with no input at all, and the same input without a basal signal. The rest of the parameters were chosen to be the same as those for the Yeast and repressilator data sets. MSE results of the 50 data fitting runs per each experiment, with and without input, are depicted in the Histograms of Figure 4.26 a and b respectively

Two important observations need to be mentioned from these histograms. The first is that no significantly different distributions were found among the two of them (t-test at p= 0.05). The second is that the fitting was performed to very low MSE (high fitness) ranges in all cases (no outliers). This results indicates that the data is very little restrictive and the fitting of the data has been very easily achieved. Unfortunately, this could indicate that the number of different solutions fitting the data equally well would be very broad.

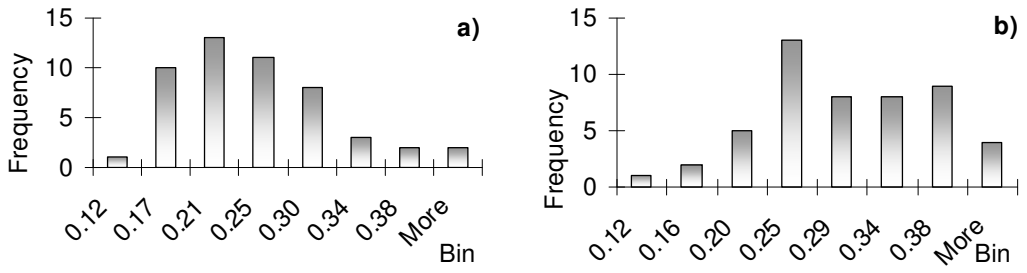


Figure 4. 26 Histograms of the fitting runs of the TDRNN model a) using an external input and b) without external input

Clustering and robust parameters identification

After clustering and performing the robust parameter identification as described in Methods, only poorly connected networks were found for both optimization groups, with and without input.

Therefore, the z-score threshold was lowered to 1 unit in other to impose a less restrictive criterion for the identification of robust parameters in the clusters. The resulting five networks encountered for the input bounded group are depicted in Figure 4.27.

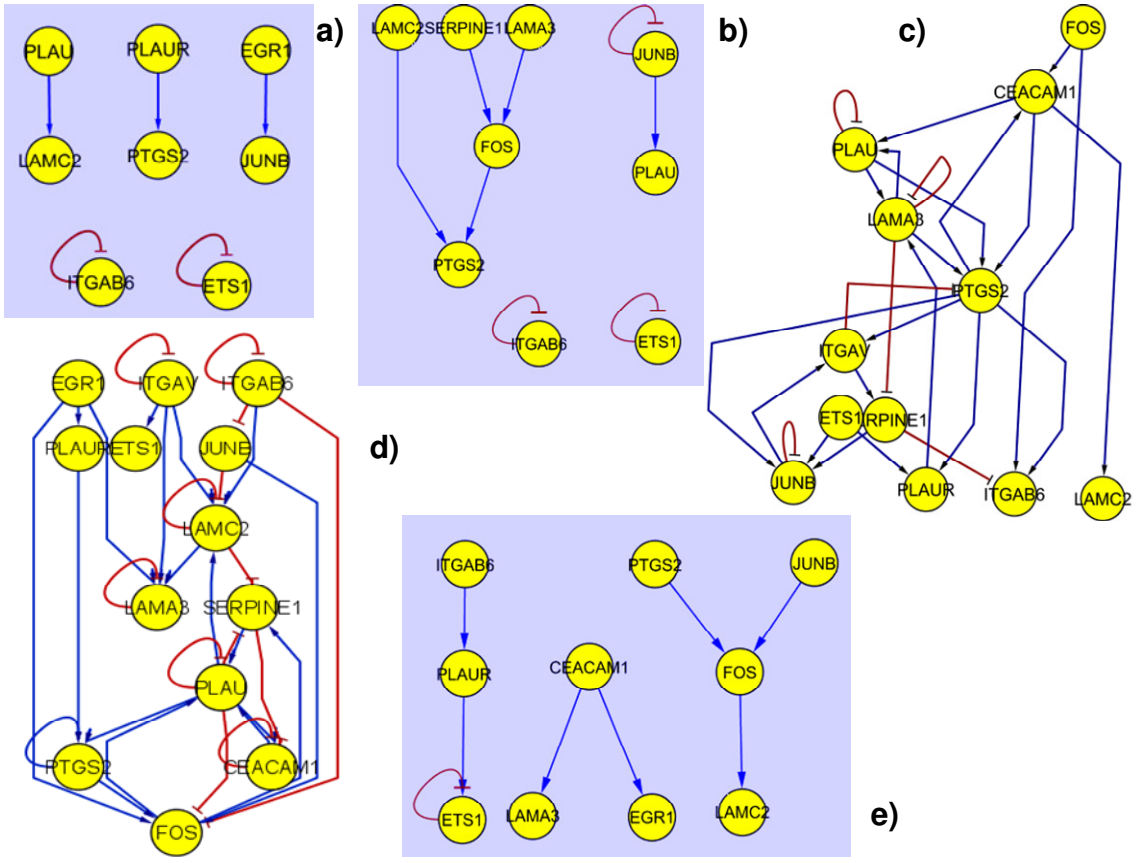


Figure 4. 27 Resulting inferred networks of the five clusters of the fitting runs: a, b and e are discarded because of lack of biological meaning; network c and d represent the potential solutions of the RE selected data.

The results depicted in figure 4.27 show that despite the fitting of the data were performed very well, no common pattern could be found as in the previous work (see Yeast in section 4.2). Notice that the index were not included since no significant difference among them was found (0.22, 0.20, 0.25, 0.23, 0.21 for the a, b, c, d and e networks). A similar situation was obtained for the case of the no input optimized group. Hence, it is not possible to elaborate serious conclusions about the different networks obtained.

However, the networks c and d exhibit a more realistic biologically founded scenario. Usually, these two networks would be the only two considered for further analysis, discarding the other three. Hence, these two networks would be an intermediary step on the iterative process of RE of GRN. One would extract some connectivity hypothesis of them and test them by simpler and faster experiment in the wet laboratory.

On the other hand, additional efforts to improve the identification of a common network pattern or robust parameters were performed. Analogous to the repressilator case section 4.1.2, the further dynamics of every optimization was simulated for every model by the equivalent of three times the optimized interval meaning 24h (1440 min). As expected, all the models of the no input run arrived to a steady state defined by their last activity states. Therefore, all the encountered models remain activated. This could be argued as the desired long term activity behavior found in migration. However, due to the optimization scheme, I consider this result just as an artifact from the indeterminacy of the system. Moreover, the models optimized including an external input behaved interestingly with a robust similar behavior, even though the prolonged simulations do not include any input. Both cases are exemplified by one randomly selected model of every group, depicted in figure 4.28.

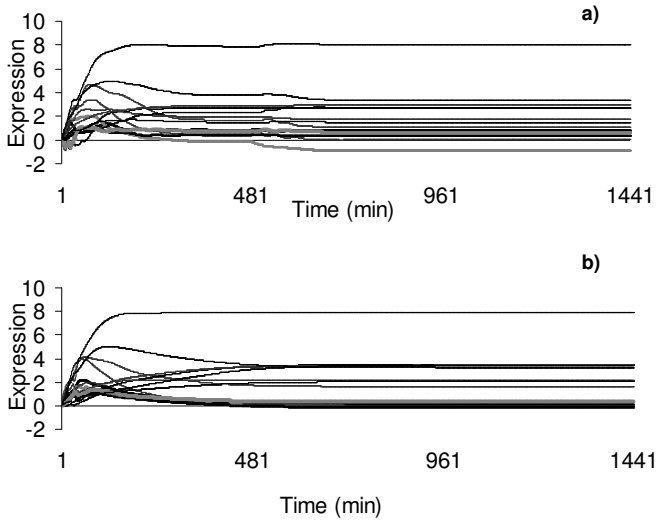


Figure 4. 28 dynamics stability analysis of two solutions of the simulated data; a) the simulation of the model using the external input until 480 min, b) simulation of the model without the external input

Notice that after withdrawing the input at the time of 8h (480 min), the model depicted in Figure 4.28 presents only a small perturbation, but it recovers by itself in the same steady state. This stable steady state was confirmed for longer intervals than 50h. Therefore, in my case, the fitting of this data drives the model to a stable steady state. However, despite this robust behaviors are logical and expected, it was still not possible to find a differential cluster in the solutions space of every group.

Definitively the modeling and process of reverse engineering GRN is possible and robust as previously shown in sections 4.1 and 4.2, but definitely it also has its limitations. As previously mentioned, the more important gain from this section is to notice the improvement opportunities for future works. For instance, for the secondly mentioned issue of measuring over the mixture of cells, an experimental improvement could be the use of the Flow cell Cytometry technique (Gray, et al., 2007). Then, quantitative information could be obtained about the proportion of every cell into the mixture. Then a numerical algorithm could be applied and signal deconvolution could be applied (Fellenberg, et al., 2001) to split the measured mRNA intensity signal according to the proportional contribution of every cellular line.

Additionally, to extend the cell system dynamics sampling would be another easy improvement for any data set like the here analyzed. As previously mentioned, having just one measurement after one hour after the supplied stimuli, it opens the possibility for behaviors like the showed by the PTGS2 gen. Notice that certainly in the network depicted in figure 4.27e this protein encoding gene appears at the top of an activation cascade. However, this simple improvement could be achieved only if the experimental setup is of a reproducible nature.

5. Discussion

In general, the reverse engineering of gene regulatory networks serves two purposes. It is concerned in the development of models with high *predictive power* to perform *in silico* experiments, as for instance theoretical knock outs. However, the main task of RE of GRN is to use such models into a broader pipeline to establish a frame work with the *inference power* to generate new knowledge about the biological networks under study. Therefore, this is a hot topic into modern Biological Sciences and new models and frameworks are intensively developed during the last ten years. However, since this is a multidisciplinary area, in fact it requires a deep knowledge on different areas such as molecular biology, mathematics, computer science, biochemistry etc. This situation has been source of limitations and even several misconceptions in the development of such models and frame works in the area. In this section, I will try to summarize the more important results obtained in this thesis, pointing out the critical steps for the area, and finally suggest some areas of improvement according to the experience gained along this work.

5.1 Model choice and data driven experiments

As explained in the Biological context, traditional enzymatic models have no chance to deal with the complexity of gene regulatory networks because the available data is by far not enough in quantity or in quality to perform such calculations. Instead, pragmatical models, whether black or grey could be useful ones, depending on particular research objectives.

In the introduction and along this thesis it was demonstrated that, the new technologically generated data are playing a double role in the area; on one hand they are the motivation for such an integrative theoretical work, but at the same time they are the bottleneck to obtain more accurate models and inference frame works. Therefore, in the last years the international community has started to deal with this situation and started to perform the so-called data-driven experiments. Data driven experiments are tedious and expensive from an experimental biologist point of view. They require, for instance, tightly sampled time-series for every stimulus-response experiment and with they respective (statistically significant) replicates. However, this data driven experiments are absolutely necessary to any serious effort in modern Systems Biology.

Sooner or later the data limitation will be solved and large amounts of data will need to be integrated into a new nascent biological epistemology. Then, I will come back to the problem of the choice of the adequate model, according to every particular research goal. In this sense, there are good and very interesting efforts to avoid systematically the misconceptions and misunderstandings that the bias from having different professional formations imposes over the choice of a model or a framework. In this context, I propose with this work, a semiautomatic (globally optimized) fast generation of models to cover, at a topological description level, two biological networks: Signal transduction and gene regulatory networks, through a general time delayed recurrent neural network. Additionally, I am proposing the innovative parameter clustering technique of the models solutions, to improve the actual inference power of different frameworks. However, there are still many issues to

fulfill in order to improve the reverse engineering of actual frame works. These issues are further discussed into next sections of this general discussion chapter.

5.2 Data selection

Following the order of the proposed workflow (see figure 2.1), the first critical step in the RE area is the data selection to reduce the dimensionality of the system to be engineered. Besides the different techniques described in the sections of related works and methods, a central problem needs to be solved: the so called data orthogonality. Ideally the selected genes or proteins (nodes, from a modeling perspective), data should be autonomous (or orthogonal) for the cellular function under study, with respect to the rest of the cellular components. These selected nodes should include sufficient information that none of the gene or proteins represented have missing information to explain its activation or inhibition, as it occurs in the reverse engineered network proposed by Li et al. where five auto inhibitory feedback loops needs to be added in order to make the module functional.

The gene regulatory goal network of the yeast cell cycle used in this study is a conceptual approximation based on experimental evidence. Here, I used just a single gene expression time series of unknown quantity of noise for reverse engineering. Hence, it would have been rather unlikely that all inferred connections would be correct. In this light, I consider that finding the 44% of the total correlations (15/34), or the 54% (7/13) of the indispensable weighted directed edges is a very good result. Obviously, I do not suggest testing experimentally for the false positives I encountered. Instead I suggest giving them a reading in the GED context, taking into account meaningful shortcuts.

In the section of the repressilator case study 4.1.4, the incompleteness of information in respect to the number of optimized nodes, I included one or two “blind” nodes into the model to fulfill the missing nodes. As this is valid for a benchmark to analyze the inference power of the models, it could not be the case for a normal pipeline as in the keratinocytes case study here analyzed. Usually, one has no precise information of the

percentage of missing nodes. The solution in this case was to use my bottom up proposal, starting from a core of nodes which was selected by using experts' knowledge and making it grow whenever more nodes needed to be added in the light of biological interest. However, the used approach in a related keratinocytes case study (Busch, et al., 2008) differs in this by the use of an expression strength ranking of the nodes (in this case genes). Since the expression range of the next added nodes is considerably smaller, for sure, they will not change the initial engineered topology, this solution imposes a strong bias to the original selected core based on the function one is interested to find.

I rather proposed to work under a bottom up approach, started with the minimum nodes information of biological interest, selected without a bias, by for instance the GeneSet enrichment platform (see Methods). Additionally, once an initial nodes core is selected and reverse engineered, one should make it grow by adding "blind nodes" for a < 25% range of the total initial nodes core and reengineer it until one can define a common topological pattern. Then, one can fix the encountered robust parameters; add new nodes data and restart the reverse engineering process. For sure, this is a lot of work to be performed, but imposes no bias towards a particular result. Therefore, the time consumption could be compensated by the knowledge one can obtain into this supervised iterative process.

5.3 Data interpolation, implications

Surprisingly, the TDRNN model is very robust against noise to perform the RE task. However, the different interpolation techniques are playing an important role to represent noisy data. My results are in agreement with : "You should not fit data with a parametric model after smoothing, because the act of smoothing invalidates the assumption that the errors are normally distributed" (Draper and Smith, 1998). Therefore, for higher quantities of noise (> 20%) the linear interpolation performs better on the synthetic data set. Consistently, the results from the experiments in section 3.2 suggest that interpolation plays an important role to facilitate the approximation of the data. Here, the MSE of the TDRNN using linear interpolation is

twice as low as when the same model is using another interpolation approaches. Therefore, the relationship between these MSE and the GED results suggests that this large difference is playing a significant role to the entire RE process. Hence, the results suggest the use of linear interpolation unless the percentage of noise is low.

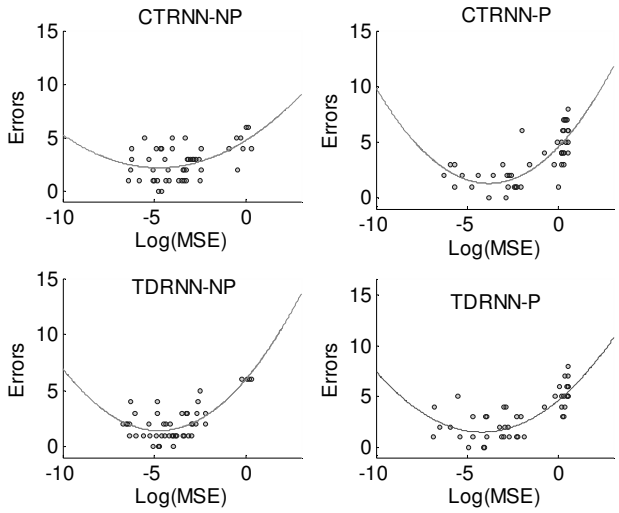
The superior reverse engineering performance of my TDRNN model compared to the CTRNN model on the synthetic network was obtained under different circumstances such as incomplete or noisy data. This achievement was obtained despite the fact that this data set does not have a different time delay between genes and is highly symmetric, which supposes an advantage for the CTRNN model because it already has synchronized node responses.

5.4 Data fitting and inference power relationship

The next critical step into any RE framework is, to define the optimal data fitting scheme used in every particular problem. Notice, this includes any additionally considered information as the sparsely mixed optimization function here introduced.

Since the MSE is not a normalized error measure (Battaglia, 1996) but depends on the range of the data, it is not possible to determine a MSE threshold on different data sets to distinguish good from bad data fitting. Therefore, we see for the shortest periods [0.2, 0.33] of the 4.1.2 repressilator case study, that the less information to measure the error, the lower is the MSE inter-group limit obtained (see lower whiskers from boxplots in figure 4.7. However, even though these optimizations with the lower MSE have a direct correlation with the errors (see figure 4.9), the provided information is still too few to restrict the solution space, and even several unrelated solutions are found (see figure 4.10, upper panels). In other words, for these low information conditions the GA get trapped in local maxima with fitness levels as high as the correct solution.

Moreover, I have realized that fitness alone is a poor indicator with variable correlation to the inference power (RE task) of a model. This is especially true for the high fitness (low MSE) [0.5, 0.75]period regions in section 4.1.2 and what should be seen as enough data [1, 3] period regions on the 4.1.2 experiment. Additionally, notice that the easy-to-fit zone does not correlate with the anti-correlation zone among 1 to 1.5 periods. Indeed, correlation depends on both: data structure and relative fitness values (as previously explained, the MSE is not an absolute error measurement). This is complex at a first glance, but it is easier to understand through some examples from the 4.1.2 experiments. Here, in figure 5.1 are plotted the scatterplots of the individual-errors vs. the fitness expressed by the logarithm of the MSE, for the smallest optimized interval (0.2 periods) of the four models.



5. 1 scatterplots of the relationship between data fitting (log MSE) and inference power (errors); for these fitness values, the relationship follows a quadratic function.

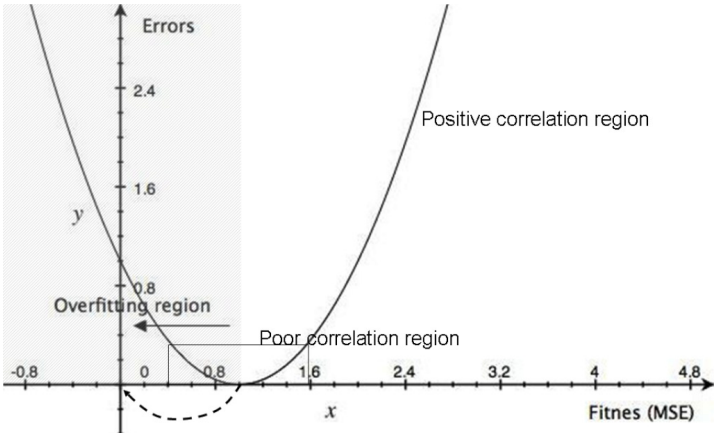
As it can be seen in figure 5.1, the correlation in all these cases does not follow a linear tendency but appears like a quadratic regression function. This is what is expected for this insufficient information data structure. The interpretation is, that for regions with poor fitting values (right side of the vertex of every parabola, MSE $\geq$ 1) the models are facing problems to fit the data, therefore there are several individual-errors and therefore, one faces a strong positive correlation.

For imperfect data (as for only 0.2 period), it occurs that fitting this imperfect data structure whit high accuracy (left side of the parabolas, negative log (MSE)) is

associated to a wrong network topology, a bad generalization which is an analogous of over-fitting. This occurs because the optimization process of the models easily finds different solutions than the expected one which could better fit that imperfect data structure. Therefore, the number of individual-errors increases according to the lowest mean value of the quadratic regression at the vertex of the parabolas. This parabolic behavior is exactly what is expected when over-fitting problems arise.

Obviously one can ask how it is possible then that on the figure 4.9 all the correlations for this period data show a clear positive correlation. The answer is in the boxplots of figure 4.7, this data structures (0.2 periods) have the largest distributions of MSE, ranging from very small MSE with high fitness, shown by the lower whiskers, to large MSE with few outliers, meaning covering from over fitting regions to regions with poor fitting. However, despite having few outliers the distributions are not normal and therefore, the Spearman correlation were used. Contrary to the Pearson correlation which is based on linear correspondence, Spearman is based on ranked data and therefore a relative strong correlation could still be observed for these data regions. As previously mentioned, one should not base important conclusions just on correlations without analyzing the scatterplots. However, here there are no contradictions but an answer to the complex behavior of the correlation.

In principle, one should expect that the vertices of this parabolas moves towards the origin of the graph (as depicted in figure 5.2 by a dashed arrow) with the increment of information (longer optimized period regions). Then, there should be only a positive correlation or no correlation when the fitness is around the vertex (as depicted by the box around the vertex in figure 5.2). In the case of the easy–to-fit-regions of figure 4.7 occurs that all the fitting runs have a good fitness (low MSE) without over-fitting and few fitting failure. Therefore, they correlation is around the vertex of this parabolic correlation regression, meaning close to cero or no Spearman correlation.



5. 2 Proposed parabolic behavior of the inference power and fitness relationship

Notice that a second issue for such kind of correlations is, that the individual-errors on the y-axis are measured in a discrete scale while the fitness (MSE) has a continuous scale. Hence, for small variations of the MSE around the vertex, there are no variations of individual-errors which means no correlation. This explains why a quadratic correlation regression would give the same results. But it is more important to take into account the fitness (MSE) region where the optimization runs are. These last two arguments partially explain the third fitness (MSE) optimization region,

Since the fitness alone is a poor indicator for the RE task, there should be other criteria to optimize the models additionally, to the fitting of the data. For instance, one should have an idea of the noise or incompleteness of the data in order to use the "early stopping" criteria according to some reference. This reference is usually not available, but could be or should be part of a routinely experimental setup. It could be established by the use of any previous knowledge of the network to be engineered. Another possibility is to perform the optimizations letting the model to explore for different possible solutions (or fitness regions), then to utilize the clustering technique to identify these different solutions and to evaluate them with the index here suggested.

Regarding the network sparsity, which was thought as another criteria to optimize the models, I found that the here introduced adaptive pruning function decreases the connectivity of the resultant networks, while no considerable alteration of their fitness occurs.

While the pruning function helps to split the different network solutions and consequently the clusters, I have found no evidence that pruning alone improves the RE task. Now it is clear that network sparsity is not a decisive criteria for the RE task. Instead, the connectivity characteristics as small world (Potapov, et al., 2005) or scale free (Balaji, et al., 2006; Chen, et al., 2008; Iguchi, et al., 2007; Kauffman, 2004; Wildenhain and Crampin, 2006; Zhou, 2005) networks could be taken into account . However, in my case the size of the studied networks was not suitable for such an analysis.

5.5 Reverse engineering framework, improving the robust parameter selection

An effective improvement to the reverse engineering is the clustering of different model solutions based on the identification of robust parameters. This clustering approach is innovative since it is applied directly to the parameter space and not to the dynamics of the models. This difference is exemplified in section 4.1.4. The use of Lyapunov exponents to restrict solutions to those with stable and similar dynamics could not find the two valid repressilators architectures, but the clustering applied to the parameters solution space does it. Additionally, the same clustering approach increased the number of correct edges (see. section 4.2.3), while not affecting the number of false positives.

Moreover, I found the cluster index to be useful to distinguish between possible solutions. Instead of a unique solution this strategy will usually narrow down putative solutions to few possibilities to be validated experimentally.

6. Conclusions

Reverse engineering of gene regulatory networks is an iterative process between experiments and modeling. In this process, the correct representation of the original system is a critical step. In this sense the TDRNN has been shown to constitute an improved approach as compared to existing CTRNN or DBN models. Additionally, the clustering of the reverse engineering solutions provides a novel method to identify robust parameters within the dynamic recurrent neural networks. Altogether, I presented a supervised learning framework that helps to provide novel insight into dynamic systems properties of genetic regulatory networks both from a biological and theoretical point of view.

If the only data source is of transcriptional nature, such as RT-PCR or microarray data, only transcriptional networks can be inferred. Cellular information processing, however, is a feedback entangled process between protein signaling and gene regulation for which combined transcriptomic and proteomic data are needed. I consider the TDRNN to be the ideal model to incorporate both types of data. Due to its incorporation of time delays that can potentially range from seconds to hours, it can naturally incorporate fast responses occurring in signal transductions and slow responses from the genetic regulatory network as we have demonstrated for the yeast cell cycle data set.

7. Bibliography

Akutsu, T., Miyano, S. and Kuhara, S. (2000) Algorithms for identifying Boolean networks and related biological networks based on matrix multiplication and fingerprint function, *J Comput Biol*, **7**, 331--343.

Albertini, F. and Sontag, E. (1993) Uniqueness of weights for recurrent nets, *Proceedings of the International Symposium Math. Theory of Networks Syst.*, **II**, 599-602.

Arkin, A., Ross, J. and McAdams, H.H. (1998) Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected Escherichia coli cells, *Genetics*, **149**, 1633-1648.

Auboeuf, D., Batsche, E., Dutertre, M., Muchardt, C. and O'Malley, B.W. (2007) Coregulators: transducing signal from transcription to alternative splicing, *Trends Endocrinol Metab*, **18**, 122-129.

Bar-Joseph, Z. (2004) Analyzing time series gene expression data, *Bioinformatics*, **20**, 2493-2503.

Bar-Joseph, Z., Farkash, S., Gifford, D.K. and Simon, I.R., R. (2004) Deconvolving cell cycle expression data with complementary information, *Bioinformatics*, **20 Suppl 1**, I23-I30.

Battaglia, G.J. (1996) MSE, *AMP Journal of Technology*, **5**, 31-36.

Bay, S.D., Shrager, J. and Pohorille, A.L., P. (2002) Revising regulatory networks: from expression data to linear causal models., *J Biomed Inform*, **35**, 289-297.

Bebis, G.G., Michael. Kaspalris, Takis. (1996) Coupling weight elimination and genetic algorithms. *Neural Networks, 1996., IEEE International Conference on*. Washington, DC, USA, 1115-1120.

Beer, R.D. (1995) On the dynamics of small continuous-time recurrent neural networks, *Adaptive Behavior*, **3(4)**, 469-509.

Beer, R.D. (2006) Parameter space structure of continuous-time recurrent neural networks, *Neural Comput*, **18**, 3009-3051.

Ben-Dov, C., Hartmann, B., Lundgren, J. and Valcarcel, J. (2008) Genome-wide analysis of alternative pre-mRNA splicing, *J Biol Chem*, **283**, 1229-1233.

Blackwood, E.M. and Kadonaga, J.T. (1998) Going the distance: a current view of enhancer action, *Science*, **281**, 60--63.

Blanco, A. and Delgado, M.P., M. C. (2001) A real-coded genetic algorithm for training recurrent neural networks, *Neural Netw*, **14**, 93-105.

Boutros, M. and Ahringer, J. (2008) The art and design of genetic screens: RNA interference, *Nat Rev Genet*, **9**, 554-566.

Busch, H., Camacho-Trullio, D., Rogon, Z., Breuhahn, K., Angel, P., Eils, R. and Szabowski, A. (2008) Gene network dynamics controlling keratinocyte migration, *Mol Syst Biol*, **4**, 199.

Cao, Y. and Gillespie, D.T.P., L. R. (2006) Efficient step size selection for the tau-leaping simulation method, *J Chem Phys*, **124**, 44109.

Cremer, T. and Cremer, C. (2001) Chromosome territories, nuclear architecture and gene regulation in mammalian cells, *Nat Rev Genet*, **2**, 292--301.

Chen, K. and Rajewsky, N. (2007) The evolution of gene regulation by transcription factors and microRNAs, *Nat Rev Genet*, **8**, 93--103.

Chen, T., He, H.L. and Church, G.M. (1999) Modeling gene expression with differential equations, *Pac Symp Biocomput*, 29--40.

D'Haeseleer, P., Liang, S. and Somogyi, R. (2000) Genetic network inference: from co-expression clustering to reverse engineering, *Bioinformatics*, **16**, 707-726.

D'Haeseleer, P., Wen, X., Fuhrman, S. and Somogyi, R. (1999) Linear modeling of mRNA expression levels during CNS development and injury, *Pac Symp Biocomput*, 41-52.

D´Haeseleer, P. (2000) Reconstructing gene networks from large scale gene expression data, *Ph.D. Thesis*.

de Jong, H. (2002) Modeling and simulation of genetic regulatory systems: a literature review., *J Comput Biol*, **9**, 67-103.

Downes, S.M. (2004) Alternative splicing, the gene concept, and evolution, *Hist Philos Life Sci*, **26**, 91-104; discussion 123-109.

Draper, N.R. and Smith, H. (1998) *Applied Regression Analysis*. John Wiley \& Sons, New York.

Edwards, R. and Glass, L. (2000) Combinatorial explosion in model gene networks, *Chaos*, **10**, 691--704.

Elowitz, M.B.L., S. (2000) A synthetic oscillatory network of transcriptional regulators, *Nature*, **403**, 335-338.

Fellenberg, K., Hauser, N.C., Brors, B., Neutzner, A., Hoheisel, J.D. and Vingron, M. (2001) Correspondence analysis applied to microarray data, *Proc Natl Acad Sci U S A*, **98**, 10781-10786.

Feyzi, E., Sundheim, O., Westbye, M.P., Aas, P.A., Vagbo, C.B., Otterlei, M., Slupphaug, G. and Krokan, H.E. (2007) RNA base damage and repair, *Curr Pharm Biotechnol*, **8**, 326-331.

Florin, L., Hummerich, L., Dittrich, B.T., Kokocinski, F., Wrobel, G., Gack, S., Schorpp-Kistner, M., Werner, S., Hahn, M., Lichter, P. and Szabowski, A.A., P.

(2004) Identification of novel AP-1 target genes in fibroblasts regulated during cutaneous wound healing, *Oncogene*, **23**, 7005-7017.

Friedman, N., Linial, M., Nachman, I. and Pe'er, D. (2000) Using Bayesian networks to analyze expression data, *J Comput Biol*, **7**, 601--620.

Funahashi, K. (1989) On the Approximate Realization of Continuous Mapping By Neural Networks, *Neural Networks*, **2**, 183-192.

Gillespie, D.T. (1977) Exact stochastic simulation of coupled chemical reactions, *J. Phys. Chem.*, **81**, 2340-2361.

Gillespie, D.T. (1992) A rigorous derivation of the chemical master equation, *Physica A*, **188**, 404-425.

Gray, A.C., McLeod, J.D. and Clothier, R.H. (2007) A review of in vitro modelling approaches to the identification and modulation of squamous metaplasia in the human tracheobronchial epithelium, *Altern Lab Anim*, **35**, 493-504.

Guet, C.C., Elowitz, M.B. and Hsing, W.L., S. (2002) Combinatorial synthesis of genetic networks, *Science*, **296**, 1466-1470.

Hancock, E.R. (2005) Graph Edit Distance from Spectral Seriation, *IEEE Trans. Pattern Anal. Mach. Intell.*, **27**, 365--378.

Hasty, J., Pradines, J., Dolnik, M. and Collins, J.J. (2000) Noise-based switches and amplifiers for gene expression, *Proc Natl Acad Sci U S A*, **97**, 2075--2080.

He, L.H., G. J. (2004) MicroRNAs: small RNAs with a big role in gene regulation., *Nat Rev Genet*, **5**, 522-531.

Hill, T. and Lewicki, P. (2006) *Statistics : methods and applications : a comprehensive reference for science, industry, and data mining*. Tulsa, Oklahoma.

Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P. and Kummer, U. (2006) COPASI--a COmplex PAthway SImulator, *Bioinformatics*, **22**, 3067-3074.

Hu, X., Maglia, A. and Wunsch, D. (2005) A general recurrent neural network approach to model genetic regulatory networks, *Conf Proc IEEE Eng Med Biol Soc*, **5**, 4735-4738.

Hu, X., Maglia, A., Wunsch, D. (2005) A general recurrent neural network approach to model genetic regulatory networks. , *In Conf Proc IEEE Eng Med Biol Soc*, **5**, 4735 – 4738

Huber, W., von Heydebreck, A., Sultmann, H., Poustka, A. and Vingron, M. (2002) Variance stabilization applied to microarray data calibration and to the quantification of differential expression, *Bioinformatics*, **18 Suppl 1**, S96-104.

Itzkovitz, S., Tlusty, T. and Alon, U. (2006) Coding limits on the number of transcription factors, *BMC Genomics*, **7**, 239.

Juliano, R., Alam, M.R., Dixit, V. and Kang, H. (2008) Mechanisms and strategies for effective delivery of antisense and siRNA oligonucleotides, *Nucleic Acids Res*, **36**, 4158-4171.

Kastner, J., Solomon, J. and Fraser, S. (2002) Modeling a hox gene network in silico using a stochastic simulation algorithm, *Dev Biol*, **246**, 122--131.

Kauffman, S. (1971) Gene regulation networks: a theory for their global structure and behaviors, *Curr Top Dev Biol*, **6**, 145-182.

Kauffman, S., Peterson, C. and Samuelsson, B.T., C. (2004) Genetic networks with canalyzing Boolean rules are always stable, *Proc Natl Acad Sci U S A*, **101**, 17102-17107.

Keasling, J.D., Kuo, H. and Vahanian, G. (1995) A Monte Carlo simulation of the Escherichia coli cell cycle, *J Theor Biol*, **176**, 411--430.

Kim, S.-S. (1998) Time-delay recurrent neural network for temporal correlations and prediction, *Neurocomputing 20* **20**, 253—263

Lanctôt, C., Cheutin, T., Cremer, M., Cavalli, G. and Cremer, T. (2007) Dynamic genome architecture in the nuclear space: regulation of gene expression in three dimensions, *Nat Rev Genet*, **8**, 104--115.

Lewis, B.A. and Reinberg, D. (2003) The mediator coactivator complex: functional and physical roles in transcriptional regulation, *J Cell Sci*, **116**, 3667--3675.

Li, F., Long, T., Lu, Y. and Ouyang, Q.T., C. (2004) The yeast cell-cycle network is robustly designed, *Proc Natl Acad Sci U S A*, **101**, 4781-4786.

Liang, S., Fuhrman, S. and Somogyi, R. (1998) Reveal, a general reverse engineering algorithm for inference of genetic network architectures, *Pac Symp Biocomput*, 18--29.

Liao, X. and Wang, J. (2003) Global dissipativity of continuous-time recurrent neural networks with time delay, *Phys Rev E Stat Nonlin Soft Matter Phys*, **68**, 016118.

Lipan, O.W., W. H. (2005) The use of oscillatory signals in the study of genetic networks, *Proc Natl Acad Sci U S A*, **102**, 7063-7068.

Ma, J. (2004) The capacity of time-delay recurrent neural network for storing spatio-temporal sequences, *Neurocomputing* **62** 19 – 37.

MacQueen, J.B. (1967) Some Methods for classification and Analysis of Multivariate Observations, *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability", Berkeley, University of California Press*, **1**, 281-297.

Mathayomchan, B.B., R. D. (2002) Center-crossing recurrent neural networks for the evolution of rhythmic behavior, *Neural Comput*, **14**, 2043-2051.

McAdams, H.H. and Arkin, A. (1997) Stochastic mechanisms in gene expression, *Proc Natl Acad Sci U S A*, **94**, 814--819.

Misteli, T. (2001) Protein dynamics: implications for nuclear architecture and gene expression, *Science*, **291**, 843--847.

Mjolsness, E. and Sharp, D.H.R., J. (1991) A connectionist model of development., *J Theor Biol*, **152**, 429-453.

Murphy, K.a.M., S. (1999) Modelling Gene Expression Data using Dynamic Bayesian Networks, *Computer Science Division, University of California, Life Science Division. Lawrence Berkeley National Laboratory*.

Paddison, P.J. (2008) RNA interference in mammalian cell systems, *Curr Top Microbiol Immunol*, **320**, 1-19.

Pe'er, D., Regev, A., Elidan, G. and Friedman, N. (2001) Inferring subnetworks from perturbed expression profiles, *Bioinformatics*, **17 Suppl 1**, S215--S224.

Perkins, T.J., Jaeger, J. and Reinitz, J.G., L. (2006) Reverse engineering the gap gene network of Drosophila melanogaster, *PLoS Comput Biol*, **2**, e51.

Pilpel, Y. and Sudarsanam, P.C., G. M. (2001) Identifying regulatory networks by combinatorial analysis of promoter elements, *Nat Genet*, **29**, 153-159.

Potapov, A.P., Voss, N., Sasse, N. and Wingender, E. (2005) Topology of mammalian transcription networks, *Genome Inform*, **16**, 270--278.

Radde, N. and Kaderali, L. (2007) Bayesian Inference of Gene Regulatory Networks Using Gene Expression Time Series Data. In. Springer Berlin / Heidelberg, 1-15.

Rao, A., Iii, A.O.H., States, D.J. and Engel, J.D. (2007) Inferring time-varying network topologies from gene expression data, *EURASIP J Bioinform Syst Biol*, 51947.

Reinitz, J.S., D. H. (1995) Mechanism of eve stripe formation, *Mech Dev*, **49**, 133-158.

Robles-Kelly, A. and Hancock, E.R. (2005) Graph edit distance from spectral seriation, *IEEE_J_PAMI*, **27**, 365--378.

Savageau, M.A. (1969) Biochemical systems analysis. II. The steady-state solutions for an n-pool system using a power-law approximation, *J Theor Biol*, **25**, 370-379.

Schreiber, M., Kolbus, A., Piu, F., Szabowski, A., Mohle-Steinlein, U., Tian, J., Karin, M., Angel, P. and Wagner, E.F. (1999) Control of cell cycle progression by c-Jun is p53 dependent, *Genes Dev*, **13**, 607-619.

Shannon, P., Markiel, A., Ozier, O., Baliga, N.S., Wang, J.T., Ramage, D., Amin, N., Schwikowski, B. and Ideker, T. (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks, *Genome Res*, **13**, 2498-2504.

Slepoy, A., Thompson, A.P. and Plimpton, S.J. (2008) A constant-time kinetic Monte Carlo algorithm for simulation of large biochemical reaction networks, *J Chem Phys*, **128**, 205101.

Smolen, P. and Baxter, D.A.B., J. H. (2000) Mathematical modeling of gene networks, *Neuron*, **26**, 567-580.

Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D. and Futcher, B. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization, *Mol Biol Cell*, **9**, 3273--3297.

Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O. and Botstein, D.F., B. (1998) Comprehensive identification of cell cycle-

regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization, *Mol Biol Cell*, **9**, 3273-3297.

Stamm, S. (2002) Signals and their transduction pathways regulating alternative splicing: a new dimension of the human genome, *Hum Mol Genet*, **11**, 2409-2416.

Stoll, G. and Rougemont, J.N., F. (2006) Few crucial links assure checkpoint efficiency in the yeast cell-cycle network, *Bioinformatics*, **22**, 2539-2546.

Sturn, A., Quackenbush, J. and Trajanoski, Z. (2002) Genesis: cluster analysis of microarray data, *Bioinformatics*, **18**, 207-208.

Subramanian, A., Tamayo, P., Mootha, V.K., Mukherjee, S., Ebert, B.L., Gillette, M.A., Paulovich, A., Pomeroy, S.L., Golub, T.R. and Lander, E.S.M., J. P. (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles., *Proc Natl Acad Sci U S A*, **102**, 15545-15550.

Swain, M., Hunniford, T., Dubitzky, W., Mandel, J. and Palfreyman, N. (2005) Reverse-engineering gene-regulatory networks using evolutionary algorithms and grid computing, *J Clin Monit Comput*, **19**, 329-337.

van Nimwegen, E. (2003) Scaling laws in the functional content of genomes, *Trends Genet*, **19**, 479-484.

van Someren, E.P., Wessels, L.F. and Reinders, M.J. (2000) Linear modeling of genetic networks from experimental data, *Proc Int Conf Intell Syst Mol Biol*, **8**, 355--366.

van Someren, E.P., Wessels, L.F.A. and Backer, E.R., M. J. T. (2002) Genetic network modeling, *Pharmacogenomics*, **3**, 507-525.

Wahde, M.H., J. (2000) Coarse-grained reverse engineering of genetic regulatory networks, *Biosystems*, **55**, 129-136.

Weaver, D.C., Workman, C.T. and Stormo, G.D. (1999) Modeling regulatory networks with weight matrices, *Pac Symp Biocomput*, 112--123.

West, A.G., Gaszner, M. and Felsenfeld, G. (2002) Insulators: many functions, many mechanisms, *Genes Dev*, **16**, 271--288.

Whitley, D. (1993) *"A genetic algorithm tutorial," Tech. Rep. CS-93-103*. Department of Computer Science, Colorado State University, Fort Collins, CO 8052.

Wilson, C.L. and Miller, C.J. (2005) Simpleaffy: a BioConductor package for Affymetrix Quality Control and data analysis, *Bioinformatics*, **21**, 3683--3685.

Wu, C.C., Huang, H.C., Juan, H.F. and Chen, S.T. (2004) GeneNetwork: an interactive tool for reconstruction of genetic networks using microarray data, *Bioinformatics*, **20**, 3691-3693.

Wuensche, A. (1998) Genomic regulation modeled as a network with basins of attraction, *Pac Symp Biocomput*, 89--102.

Erklärung

Ich, David Camachio Trujillo, habe diese Dissertation selbst verfasst und mich dabei keiner anderen als der von mir ausdrücklich bezeichneten Quellen und Hilfen bedient. Experimentelle Daten bzw. Materialien, die nicht von mir selbst erhoben bzw. hergestellt wurden, habe ich besonders kenntlich gemacht.

Ich habe an keiner anderen Stelle ein Prüfungsverfahren beantragt und diese Dissertation auch nicht anderweitig in dieser oder anderer Form bereits als Prüfungsarbeit verwendet oder einer anderen Fakultät als Dissertation vorgelegt.

Teile der vorliegenden Arbeit wurden im Vorfeld in Absprache publiziert:

- David Camacho-Trujillo, Hauke Busch, and Roland Eils (2008): Reverse Engineering Gene Regulatory Networks with Time Delay Recurrent Neural Networks. Accepted for publication at Bioinformatics

- Busch, H., Camacho-Trullio, D., Rogon, Z., Breuhahn, K., Angel, P., Eils, R. and Szabowski, A. (2008) Gene network dynamics controlling keratinocyte migration, *Mol Syst Biol*, **4**, 199.

- Schnickmann,S., Camacho-Trujillo, D., Bissinger, M., Eils, R., Angel, P., Schirmacher1, P., Szabowski, A., Breuhahn, K. (2008): 1AP-1 controlled hepatocyte growth factor (HGF) activation promotes keratinocyte migration via CEACAM1 and uPA/uPAR, Accepted at Journal of Investigative Dermatology

- Camacho, D., Busch, H., Eils, R., Angel, P., Sbawoski, A.. (2008) Means and mehtods for diagnosing metastasizing potentials of tumor cells. Patent Pend.

Heidelberg, den _____

David Camacho Trujillo