TECHNISCHE UNIVERSITÄT WIEN

VIENNA UNIVERSITY OF TECHNOLOGY

MASTERARBEIT

Reconstruction of Functional Context from Heterogenous Data Networks

Ausgeführt am
Institut für Computersprachen
Arbeitsgruppe für Theoretische Informatik und Logik

unter der Anleitung von

Ao.Univ.-Prof. Mag.rer.nat. Dipl.-Ing. Dr.techn. Rudolf Freund

durch

Andreas Bernthaler
Burggasse 51 / 9 / 65
1070 Wien

Datum Unterschrift(Student)

Acknowledgments

First of all I am deeply indebted to my family for cheering me up, their emotional and financial support and especially for believing in me over all these years.

Furthermore grateful thanks to all emergentec staff for invaluable help and advice, for having the patience in giving me answers to my uncountable questions and for the unique and unforgettable working environment. Without their backing and assistance this work would not have been possible.

In particular my gratitudes go to Bernd Mayer for his expert advise and mentorship in scientific working, and especially for the time he invested in supporting me during the final stage of this thesis.

Special thanks as well go to Johannes Söllner. He gave me the chance to develop interest in the highly interesting field of computational biology by carrying forward his enthusiasm and offering insight into unique perspectives during the last years.

Finally I want to express my sincere thanks to my friends for giving me invaluable social support and spending a great time with me over the past years, and first and foremost I would like to thank Robert Csapo for being a life-time friend.

This work was performed at and financed by emergentec biodevelopment GmbH, Rathausstrasse 5/3, 1010 Vienna.

Abstract

Background

Research in molecular biology has significantly changed towards utilizing explorative analysis on the basis of vast, but heterogeneous data sets - in contrast to purely hypothesis driven approaches. This development has grounded on the availability of novel experimental technologies (the 'omics' revolution) for monitoring cell-wide events, complemented by computational procedures for data prediction and systems modeling.

Present procedures for understanding cellular processes and identification of relevant key players (biomarkers) have utilized explorative analysis and classical statistic approaches in a sequential manner, with the ultimate goal of identifying functional context and associated biomarkers. With upcoming approaches on the basis of Systems Biology a more integrated procedure on the basis of given, but heterogeneous data sets has emerged. Under these premises the sequential linking of data will change towards a parallel, integrated analysis approach: Available large scale data on cellular processes covering gene expression, protein abundance, protein location and their functional interplay are contextually analyzed in contrast to a sequential analysis procedure which focuses on a 'from gene to protein to function' approach. This thesis outlines a novel computational methodology aimed at deciphering functional context and biomarkers: Our concept is following the framework of dynamical hierarchies and emergence as organizational principle of the underlying biological processes. This framework allows a parallel and integrative analysis of heterogeneous sources from 'omics'.

Results

Our implementation is grounded on an object representation of the cellular proteome, complemented by a set of object interaction functions for realizing an iterative reconstruction of functional dependencies between the objects based on initially provided data and parameters. In the present implementation objects are characterized via gene expression data, transcriptional regulation factors, gene ontology terms, intracellular location information and protein-protein interaction data.
Interaction functions were designed for describing the 'similarity' of objects on the individual data level, reaching an interaction graph representation of the system where weighted edges encode similarity and context between nodes (molecular objects).
Various update scenarios were tested considering different neighborhood functions and transition matrices, complemented by stochastic update functions including the Metropolis criterion in a Monte Carlo - like procedure.

Next to analyzing the dynamic properties of the system representation Proof of Concept test series were designed. We evaluated the propensity of our simulation framework for allowing de novo reconstruction of functional dependencies by utilizing the apoptosis as well as the MAPK signaling pathway.

Conclusion

We established a novel simulation concept capable of handling and analyzing heterogeneous biological data sources given for intracellular processes. Analysis of the dynamical properties of our object framework identified stable solutions, and comparison to the well established cellular apoptosis framework revealed partial de novo reconstruction of correct functional dependencies.

In a next step our concept will be broadened for covering the entire human proteome, additionally including further 'omics' data for further refining protein interaction network and biomarker discovery.

Zusammenfassung

Hintergrund

In Kontrast zu rein hypothesengetriebenen Ansätzen verlagerte sich die Forschung in der Molekularbiologie in signifikanter Weise in Richtung explorative Analyse auf Basis reichhaltiger Datensätze. Diese Tatsache gründet auf der Entwicklung von neuen experimentellen Technologien (Stichwort: 'omics' - Revolution) zur Beobachtung von gesamtzellulären Prozessen, ergänzt durch computergestützte Algorithmen für Datenvorhersage und Systemmodellierung.

Verfahren, welche zelluläre Prozesse zu erklären und relevante Schlüsselstellen (Biomarker) zu identifizieren versuchen, bedienen sich der explorativen Analyse und der klassischen Statistik in sequentieller Art und Weise, um letztendlich funktionalen Kontext aus gegebenen Daten ableiten zu können.

Aus grundlegenden Ideen der neuen Disziplin Systems Biology entstanden mittlerweile integrativere Ansätze auf Basis dieser gegebenen, jedoch sehr heterogenen Datensätze. Unter diesen Umständen ist anzunehmen, dass eine Entwicklung - weg von einer sequentiellen und hin zu einer parallel getriebenen - integrativeren Analyse stattfinden wird: Die verfügbaren 'large-scale' Daten umfassen Genexpression, Proteinexpression, -lokalisation und -interaktion. Funktionale Abhängigkeiten werden im Kontext analysiert - im Gegesatz zu dem sequentiellen Ablauf - folgend der Herangehensweise 'from gene to protein to function'.

Im Rahmen dieser Diplomarbeit erfolgte die Implementierung einer neuen Methode, welche darauf abzielt, Biomarker im funktionalen Kontext zu entschlüsseln: Unser Konzept folgt dem Ansatz der Dynamischen Hierarchien respektive der Emergenz als zugrundeliegendes organisatorisches Prinzip biologischer Prozesse und erlaubt eine parallele und integrative Analyse der heterogenen 'omics'-Daten.

Ergebnisse

Unsere Implementierung basiert auf einer Objektrepräsentation des zellulären Proteoms, erweitert durch einen Satz an Bewertungsfunktionen für Objekt-Interaktionen zur Realisierung einer iterativen Rekonstruktion der funktionalen Abhängigkeiten zwischen den Objekten, basierend auf gegebenen Daten und Parametern. In der gegenwärtigen Implementierung werden die Objekte durch Genexpressionsdaten, Transkriptionsregulationsfaktoren, Gen Ontologien, intrazellulären Lokationen und Protein-Protein Interaktionsdaten beschrieben.
Bewertungsfunktionen wurden formalisiert, um ein Ähnlichkeitsmaß für Objekte auf jedem individuellen Datenlevel zu schaffen, woraus ein - das System repräsentierender - Interaktionsgraph hervorging, in welchem gewichtete Kanten Ähnlichkeit und Kontext zwischen den Knoten (den molekularen Objekten) darstellen.
Verschiedene Szenarien für die Systempropagation wurden getestet. Veränderliche Nachbarschaftsfunktionen und Bewertungsmatrizen, ergänzt durch eine stochastische - dem Monte Carlo Verfahren ähnelnde - Updatefunktion (Metropolis Kriterium) wurden evaluiert. Neben der Untersuchung der dynamischen Eigenschaften des Systems wurden 'Proof of Concept' Testreihen erstellt. Das Apoptose und das MAPK Netzwerk wurden als Referenzsystem herangezogen, um eine mit dem System generierte de novo Rekonstruktion funktionaler Abhängigkeiten damit zu vergleichen.

Interpretation

Um in heterogenen, biologischen Daten funktional zusammenhängende Netzwerke erkennen zu können, wurde ein neuartiges Simulationskonzept realisiert. Die Analyse der dynamischen Eigenschaften des Modells ergab die Identifikation von stabilen Lösungen. Im Vergleich zu dem

als Referenz gewählten Apoptose Netzwerk zeigten die Resultate eine starke Affinität, und unter anderem wiesen die Ergebnissen auch korrekte de novo Abhängigkeiten auf. In einem nächsten Schritt wird dieses Konzept in grösserem Maßstab angewandt werden, um das gesamte humane Proteom abzudecken zu können. Zukünftig werden zusätzliche 'omics'-Datenschichten mit eingebunden, um den Selektionsprozess von Protein-Interaktionsnetzwerken und Biomarkern weiter zu verfeinern.

Table of Contents

Part I
Introduction and Motivation

1 Introduction

1.1 General Background of the Thesis

One of the biggest challenges in the area of computational biology at the moment is handling the huge amount of data which was collected in the past few years and which is still enormously increasing - representative datasources and institutes gathering such data are (EBI, 2006; EMBL, 2006; NCBI, 2006; OPHID, 2006; BIOCarta, 2006; PANTHER, 2006; PDB, 2006; KEGG, 2006) just to name the most popular ones.

Besides considerations regarding performance (like physical and algorithmic management of these data sets) the efficient use and interpretation is an important task as well. High-throughput methodologies have turned the problem from 'how to get the information' towards the question 'how to gain knowledge from the information'. See (Nikolsky *et al.*, 2005) and (Roos, 2001) for further reading and applications of data integration in (von Mering *et al.*, 2005) and (Ng *et al.*, 2006). The lack of techniques focusing this deficiency keeps us from having an appropriate benefit in relation to the heap of already generated data sources.

Besides the wet lab experiments the focus on gaining more knowledge from the given data is increasingly important. Systems Biology approaches try to understand the cell as a whole again and go away from the very specialized approaches focusing on details of cellular events.

The idea for an application of this work arose from the need of holistic perspectives on biochemical data in the search for **disease associated biomarkers**. Nevertheless the basic principle is applicable for many different problems with similar characteristics.

Thesis Statement 1 *This work is related to the areas of Systems Biology, formal informatics and applied informatics. The aim is to study the option of reconstructing functional networks from strongly heterogenous data sources with the mathematical concept of dynamical hierarchies and hyperstructures. For a successful reconstruction the process is embedded in State of the Art workflows of marker identification to improve identification quality.*

1.2 Thesis Overview

The thesis consists of four parts, each of it giving a structured insight in following topics:

- **Introduction and Motivation**
  This part outlines the thesis and introduces and explains why this project was triggered.
- **Biological and Mathematical Background**
  Here the areas of biology, genetics and molecular biology, necessary for understanding the

decisions taken in the choice of data sets and in building the theoretical model are covered. This section also gives an insight into the state of the art methods in data analysis focusing especially on 'omics data', data creation in biological and biochemical experiments, and the statistical methods used to classify them. Common strategies in integrating these steps are presented.

Furthermore we explain the mathematical background which forms the model of the proposed theory, namely emergent properties and dynamical hierarchies. The advantages and disadvantages of the different existing notations are clearly pointed out and one notation is chosen to be extended and adapted to this theory.

- **The Approach: Planning of a solution**
  The selected mathematical notation is transferred to the workflow in the selection process of disease associated marker proteins, but nevertheless it is our far-sight goal to make this approach applicable to any other workflow which shows simlilar dynamic characteristics in data analysis. From the statistical view first considerations of the impact of noise in the data sets are taken into account when designing requirements and architecture.

  A requirements analysis then assures the fullfillment of the demands to the prototype. Due to the fact that the model's behaviour needs to be studied intensively, flexibility is be one major requirement. This step is not covered in detail in this thesis. Only the major aspects in implementing the software are mentioned (especially import- and the flexibility-related issues of the system).

  The last section of this part deals with performance considerations. Leaving the test set data and switching to a full data set - which means all proteins of the human genome - needs a study of technical feasability to ensure the possibility of a real life application of this system.

- **Application and Results**
  This part is addressed to the application and the practical experiences we collected in adopting the system to real data.

  The first section will deal with the topic data import and data prearrangement. It also explains why certain data was accepted for calculation, and why certain data was not. It explains the decisions how the results are validated and compared to a control group.

  Another important topic was the choice the data sets the system will be based on. In this section all information about the import data's origins is documented.

The next section holds details on the different conditions and enviroment under which the model was studied. Noise of incomplete and incorrect biological measurements in the data sets tend to make validations of models very difficult, so many characteristics of the model dynamics are studied and documented to restrict the dynamic behaviour in the workflow. Together with the data import this was one of the most complex and time consuming parts of the practical work, hence it is actually one of the major parts of this thesis.

Data visualization is needed to interpret the behaviour of the model and to establish an easy and fast platform for deciding about parametrization and statistical calculations for the next runs. The methods used are summarized briefly.

Based on the results the conclusion gives a survey on the success of this method when used with -omics data. Suggestions for future work are provided.

1.3 Scope and Goals

In the past few years researchers started continuously embedding more and more techniques from software developement, algorithm theory, data management, machine learning, simulation systems and statistics into the process of bioengineering (Altschul *et al.*, 1990; You, 2004; Taylor *et al.*, 2006; Vass *et al.*, 2006; von Mering *et al.*, 2005; Yalamanchili *et al.*, 2006) and their usage (McGinnis & Madden, 2004). Applying those methods became an indispensable tool for analyzing data and for planning high throughput experiments. Their use influences the quality of the results of predictions and wet lab experiments. (Biron *et al.*, 2006) studies bioinformatics tools various settings and documents the quality of the outcome.

Standards try to avoid data incompatibilities, but methodologies might be very different. Researchers might find similar functional information, and interpret and store them in different ways. Problems arise due to different syntax in data representation for the same information. A natural bias further complicates the situation. Researchers might perform similar experiments and get different data thus deriving different hypotheses. Data standardization in biotechnology (generally in fast growing research communities) is a difficult task.

From the IT-supported Systems Biology view there exist several approaches to integrate data into systems and models for a better understanding of the cellular process. Some experiments base on simulations and/or predictions (E-Cell, 2006; Korber *et al.*, 2006; Regenmortel, 2006; Nakai & Horton, 1999). This is a topic where many researchers already come very close to models and formal mathematics.

This thesis meeting point between formal mathematics and the biological workflow is following: Cells - like all living systems - show the phenomena of emergent properties which will be studied in detail in part II. Emergence is an pattern arising from constituent parts and an observer can interpret a function into this pattern. If data is measured in different cell experiments and hypotheses were derived from this experiments then a view on both - hypothesis and data from experiments - should give a consistent view. Dynamical hierarchies are a proper mathematical framework to examine these causalities giving a better understanding of the cellular process. The influence of noise and incompleteness of the biological data will also be studied. The result are a networks of interrelating proteins which can support the decisions in selecting disease associated biomarkers.

As a far sight goal the resulting framework will be designed to examine every emergent system (whenever only a part of the full information is known) constraint by its complexity.

1.3.1 Biomarker Selection Processes in Industry

To explain the relevancy of workflow-design for selection processes, a short introduction into business models is given. Mainly there exist two kinds of business models, which can be split up into more fine-grained models as below:

1. Biotechnological Experiments
    - Establish theory and concentrate on organism focused drug design.
    - High throughput methodologies to gain experimental data.
      In those cases companies usually consist of biologists and/or chemists, who establish experiments on a biological and medical base. The tasks mentioned under computational supported models below usually tend to be outsourced.
2. Computationally Supported Models
    - Statistical analysis of experimental data provided by a third party.
    - Support in biological workflow design.
    - **Selection process of biomarkers** by computer aided analysis of experimental data and literature. These biomarkers are sold or forwarded to preclinical trials.

1.3.2 Standard Workflows and Substantial Shortcomings

Information technology already plays an important role in molecular biology. LIMS (Laboratory Information Management Systems) have already become standard workflow supporters in managing information generated in biological workflows, where some of them integrate a wide range of services. However, usually they solely keep track of informations and do not make predictions

about targets.

In principle computationally supported selection processe for biomarkers can be divided into two different approaches. A data warehouse approach that concentrates on data analysis, and a constructing approach whose aim is to build and simulate a system in a bottom-up design. The problem of the commonly existing systems identifying targets is that data from different experiments and theories is not examined in a coherent way. The first step is mostly a purely statistical analysis which already 'overselects' from the given population (Perco *et al.*, 2006). One approach to this shortcoming is the STRING tool, developed by (von Mering *et al.*, 2005) at the EMBL (EMBL, 2006). STRING is a good tool to explore the context of a given gene, but not when analysing big amount of data for target identification. Another drawback is that common workflows do not deal with the incompleteness of biological data.

1.3.3 Goals

For this thesis' approach the computational supported workflows are focused on. In this concept analysis and construction constitute an integrative system.

Thesis Statement 2 *We want to overcome three special problems:*

- *avoid an early overselection*
- *close or create data gaps if implied by already existing data*
- *reduce the false positive rate of the data.*

This will result in a new system and the picture of proteins standing in a functional context will change. This procedure is repeated until the system converges to a stable system state.

Figure 1 shows the general concept of a common workflow (the blue part) *extended* by our solution (red part). The rules are derived from theories and hypotheses in literature, and the initial configuration of the model is represented by already gained data from analysis of experimental results. The part 'computational model' can stand for any complex approach in the candidate selection process. In this special case, it represents the model of a complex system simulating a behaviour similar to a multidimensional cellular automata. A detailed explanation on this behaviour is covered in part III.

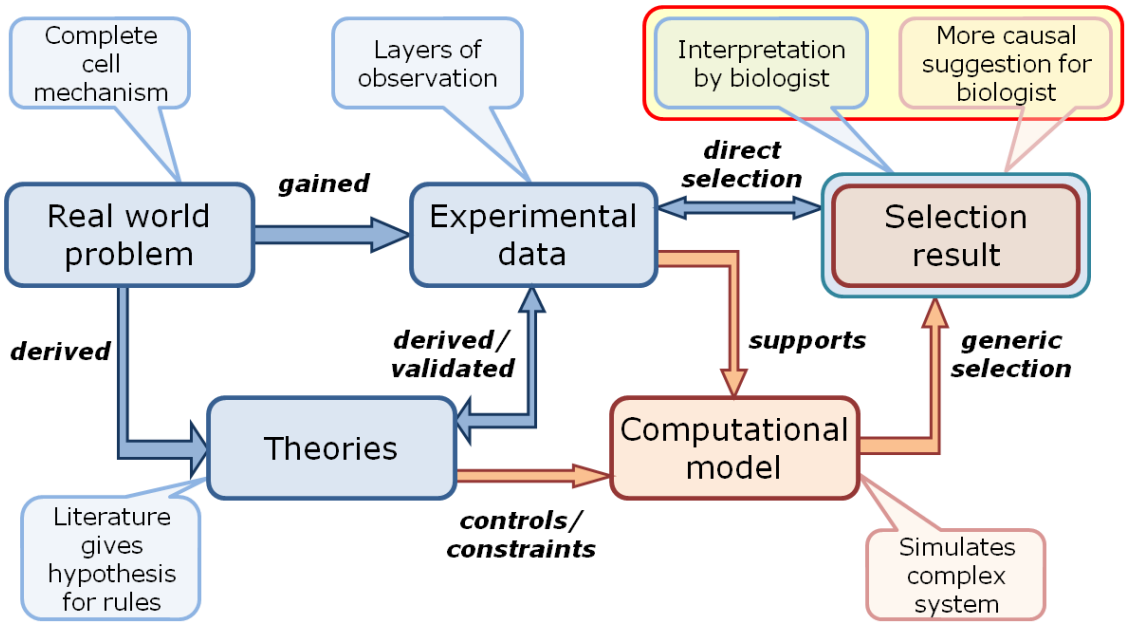

Figure 1: Scheme of a process, identifying biomarkers. The blue part is the common approach and the red one symbolizes the extension.

Part II
Biological and Mathematical Background

2 Biological Context and Basic Concepts of Systems Biology

2.1 Introduction

This section gives a briefly summarizes topics of computational biology and Systems Biology. The State of the Art in Systems Biology, bioinformatics methods and data integration is covered. The data integration part has substantial relations to omics data and its underlying experiments and is carried out in detail.

The final part of this section gives a deeper insight into an often used State of the Art workflow to select biomarkers. Chosen biomarkers are validated in wet lab experiments for verification of hypotheses.

2.2 Systems Biology and Modeling Approaches

In an excerpt of history of molecular biology Mesarovic (Mesarovic *et al.*, 2004) states out why research became so focused on details. He argues that

> ”With 30 000+ genes in the human genome the study of all relationships simultaneously becomes a formidably complex problem.”

which is a rather trivial observation. But this fact was surely the reason for need of a discipline like Systems Biology. What he meant to say is, that biologists first had to break down the system to easier subparts. Because of the complexity in cellular processes a basic understanding had to be obtained, before studying the cellular system studied as a whole. Researchers invented highly sophisticated experiments to make a start in discovering basic processes of cells. Based on the knowledge gained, scientists discovered the possibility to increase the speed of research and developed high throughput methodologies. With the permanent growth of data another approach became more relevant again. With continuous data creation the complexity of the holistic perspective is strongly reduced and there is the possibility to design new, more integrative experiments. Besides tools which concentrate on only one aspect (like on the tissue specificity or the subcellular location of a certain protein) other approaches integrating all these information become more important again. One of these approaches is the computational Systems Biology. The Systems Biology approach aims at designing experiments in an all-embracing view, its major goal is to see the organism, cell or the information derived as a whole. There is exactly the point where computational Systems Biology hooks in. These computational approaches try to find mathematical and computational models to discover new information or to explain new functional aspects when considering knowledge from many different sources, e.g. as described in detail in (Joyce & Palsson, 2006). A profound overview on computational Systems Biology can be found in (You, 2004).

Mesarovic et al. also focused on interdependencies of system views and detailed systems. For further reading on system theories, this very first article by (Mesarovic *et al.*, 2004) in the IEE Systems Biology Journal is an excellent and highly recommended reference. For more information on systems biology and especially its applications in drug discovery see (Brent, 2004; Butcher *et al.*, 2004; Apic *et al.*, 2005).

This thesis is categorized to be a systems biology approach, because different data sources are supposed to give one conclusive picture of the target organisms gene functionalities.

Just to be mentioned here, another systems approach is the e-cell project (E-Cell, 2006), where scientists try to build a working cell 'in silico' with software engineering knowledge. This project is an engineering approach, a framework which can be filled with data once there exists enough to describe a complex system like a cell sufficiently.

2.3 Experimental Data Sets and Omics Data Integration

The experimental data and the methods for extracting valuable and significant information are essential, in computational as well as in experimental biology (for further information on this see the section 2.4, "Existing Workflows and Selection Processes"). The exact methods are part of another field of study, namely bioinformatics.

When working with sets of data generated by different institutes, organizations and people standardization is an important, yet hard to establish. One difficulty is that the standardization process is still not complete in this young scientific field. The troubles with this incompatible data sets create a big overhead and are - among other difficulties - pointed out in this section. One of the major problems are the various naming conventions and identifiers. Identifiers might not be unique over different nomenclatures, thus losing their property to 'uniquely identify' a compound.

Imagine two different research groups finding a gene functionality at the same time. Each group names the gene after the naming conventions familiar to their own research group. That's how many aliases were born in the naming of cellular parts.
Institutions take care of standardizations, like the (NCBI, 2006; EBI, 2006; EMBL, 2006) (see (Hucka *et al.*, 2004) who gives an excellent overview), but new findings in research make it hard to find a timeless common denominator.
This is even harder in the area of molecular biology and genetics, because there is a natural ambiguity, e.g. when naming splicing forms, or when common believings are withdrawn on the base of newer and more conclusive theories, which causes changes in existent biochemical databases and data.

NOTE: Furthermore, if the word 'official'- or 'public'-identifier is used in this context, this addresses the well-known GeneSymbol, GeneID, or other standardized identifiers in biochemistry.

A very good resource compiling genome wide -omics data generation, analysis and prediction are (Perco *et al.*, 2006) and (Rapberger, 2007).

The following sections briefly summarize the background of the data our system is based on.

2.3.1 Gene Expression

The aim of these experiments is the detection and quantification of RNA concentration in cells under a given influence. This information gives evidence about the coexpression of genes. Coexpression **can** imply co-regulation and interaction networks on protein level. Conclusions on the functionality can be made, if some certain genes roles are already known.

Companies producing the gene expression arrays for sale base the decision which technology to use on a State of the Art in scientific literature. The experimental design and the outcome is driven by a combination of chemical, statistical, and technical methods, conjunct in one device, which supports electronic output in form of a formatted file. Results are strongly defined by the product chosen. Figure 2 illustrates a gene expression array.

An example on gene expression data is shown in figure 3. The first column is an ID, then there are some columns holding results from 'normal' samples followed by a part of columns, existing of the same experiment made on diseased samples. Every row represents one gene (or sequence) and holds results for each of the experiments in the columns.
The exact value contained by each field of this matrix depends on the color of the measured spot on the array, detected by special equipment. The color of each spot indicates how strong a certain gene is expressed, i.e. the concentration of a certain RNA fragment to its related gene. In most microarrays technologies this basic procedure is enriched by sophisticated techniques to overcome bias problems and reach a certain fault tolerance. To give an understanding on the problems resulting from the diversity and complexity of the different gene expression formats an example format description from Affymetrix™ is outlined in figure 4.

There are basically two very common practices in gene expression experiments (Lipshutz *et al.*, 1999; Knippers, 2001) :

- **Oligo microarrays**
  Oligo microarrays are e.g. produced by Affymetrix, named GeneChip™. The arrays use fragmented short DNAs strains, which have several occurrences on one measuring spot of

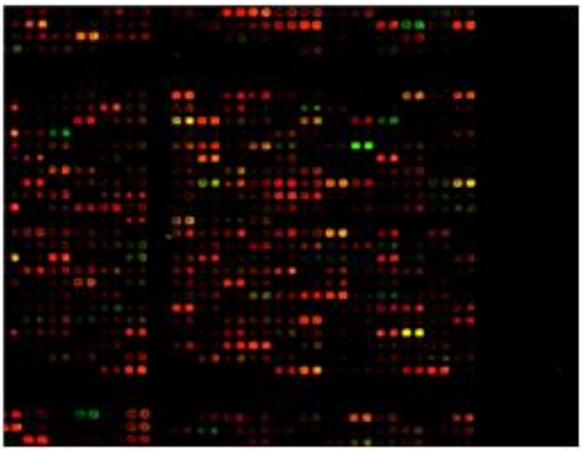

Figure 2: An example for a gene expression array after experimental processing.

Entrez ID	GeneSymbol	Probe Set Nam	OVR1	OVR2	OVR3	OVR4	OVR5	OVR6	OVR8	OVR10	OVR11	OVR12	OVR13	OVR14	OVR15	OVR16	OVR17	OVR18	OVR19	Normal1	Normal2	Normal Epith
2556	GABRA3	A28102_at	571,4	600,1	590,7	390,4	601,1	390,1	533,3	698,5	361,4	471,4	359,8	386,6	363,8	325,6	250,1	415	537,5	794,6	477,4	463,8
4958	OMD	AB000114_at	47,8	48,9	44,8	59,7	5,3	47,8	32,2	74,7	23,5	36,8	26,7	70,1	48,1	33,3	21,6	9,8	16,5	64,2	108,4	173
10964	IFI44L	AB000115_at	183,6	605,1	369,8	656,6	145,2	587,4	126,9	864,9	95	141,5	126,2	271,2	598,3	291,7	140,2	224,1	122,3	264,5	161,4	312,2
10512	SEMA3C	AB000220_at	273,7	413,5	237,1	321,5	99,6	656,1	351,9	280,8	398,5	619,5	746,2	311,2	316,7	180,2	209,8	898,6	275,4	1923,7	2046,3	1294,4
8569	MKNK1	AB000409_at	203,1	310,4	1,4	106,9	28,1	163,7	17,8	84,1	53,9	66,3	219,8	97,4	118	125,4	87	95,5	133,1	123,3	78,1	121,8
7443	VRK1	AB000449_at	66	180,8	122,3	147,8	132,3	79,3	130,7	105,2	63,2	288,5	125,4	81	145,9	65,6	45,7	92,7	107,6	61,1	62,7	83,8
7444	VRK2	AB000450_at	81,8	152,8	169	84,4	107,1	147,5	116,9	129,3	110,1	170,7	79,8	94,6	125,4	116,3	111,9	116,5	126,4	77,2	43,7	77,5
8603	C4orf8	AB000460_at	633,5	643,6	467,4	340	725,4	575,8	554,9	782,4	500,1	1833,5	565	585,8	576,7	684,3	524,3	687,1	629	918,4	761,3	749,5
6452	SH3BP2	AB000462_at	100,7	36,1	15,2	34,6	-4,3	53,8	60,5	51,1	25,7	83	31,5	19,8	51,7	98,1	52,8	93	60,7	29,6	42,4	73,6
317648	C4orf10	AB000464_at	560,1	596,2	459,3	305,1	511,3	555,9	665,5	573,9	522,6	567,4	505,7	604,5	586,6	624	522,7	745,6	627,2	853,2	769,6	812,3
317648	C4orf10	AB000466_at	105,8	-0,4	30,1	-111	49,9	70	-65,8	18,6	91,6	46,3	12,3	42,2	82,9	-72,2	4,3	-248,6	59,6	73,7	-83,2	-108
8602	C4orf9	AB000467_at	-48,3	-81,2	-154,9	-118,1	-326,8	-126,6	-275,9	-431,5	-155,2	-11,9	-251,5	-168	-171	-155,5	-183,3	-354	-224,2	-217,2	-194,6	-216,5
6047	RNF4	AB000468_at	292,3	213,2	287,9	129,6	272,1	201,9	132,6	140,4	220,7	734	393,7	128,4	180,5	186,7	240,2	227,6	146,3	240,9	200,6	140,9
9518	GDF15	AB000584_at	166,3	282,8	602	110,2	-110,3	96,3	240,3	195,2	219	52,4	121,6	35,7	264,5	282,7	116,3	265,4	344,3	-4,8	-41,6	-5,4
8642	DCHS1	AB000895_at	46,1	-8,5	-151,3	45,1	-160,3	76,8	11,5	88,2	42,3	-178,5	74,1	16,7	53,9	-277,8	-2	16,2	-418	-176,7	-224,4	-131,7
8641	PCDHGB4	AB000896_at	98,3	53,8	36,4	25,5	32,4	54,3	4,5	56,2	57,8	41,5	58	32,9	66,3	35,2	47,1	5,9	52,5	76,5	34	19,9
26025	PCDHGA12	AB000897_at	12,9	-12	10,6	-5	-29,9	5,9	36	87,2	8,2	-0,2	15,3	-4,1	55,6	36,7	23,4	3,2	41,9	22,9	33,2	20,1
8294	HIST1H4I	AB000905_at	29	9,6	43,6	35,9	66	19,2	120,2	87,4	50,9	20,4	42	83,9	70,9	41,5	17,3	56,1	37,2	47	16,6	58
2764	GMFB	AB001106_at	614,2	397,8	384,5	250,6	429,4	558,3	904,5	332,7	329,1	373,5	242,7	275,5	438,7	301	351,3	420,4	295,9	358,3	265,4	330,2
360	AQP3	AB001325_at	472,2	252,4	327,3	225,2	261,8	343,9	548	457,5	405,6	307,9	334,6	307,6	508,3	461,1	323,1	448,1	399,1	480,1	311	300,7
---	---	AB002314_at	101,3	34,1	82,4	25,2	18	26,4	74,8	190	100,8	112,6	69,1	65,3	123,9	150	95,2	61,2	-29,9	51,6	114,4	43,7
9870	KIAA0317	AB002315_at	426,6	381,3	401,7	308,2	431,5	470,2	388,2	440	241,3	590,6	258,3	175,5	263,4	314,7	149,1	316,7	264	364,2	241,6	305,4
---	---	AB002318_at	427,8	530,6	311,3	358,4	538,1	129,8	494,7	530,9	348,3	467,8	473,6	470,2	322	128,3	304,6	130,6	559,5	639,3	408,7	699,9

Figure 3: A minimal excerpt from results of a gene expression result file from an Affymetrix™ array.

A28102_at	chrX:151087279-151370326 (-) // 99.57 // q28	gamma-aminobutyric acid (GABA) A receptor, alpha 3	GABRA3	chrXq28	ENSG00000011677
AB000114_at	chr9:94216358-94226381 (-) // 99.47 // q22.31	osteomodulin	OMD	chr9q22.31	ENSG00000127083
AB000115_at	chr1:78858736-78880657 (+) // 91.59 // p31.1	interferon-induced protein 44-like	IFI44L	chr1p31.1	ENSG00000137959
AB000220_at	chr7:80209790-80386603 (-) // 96.87 // q21.11	sema domain, immunoglobulin domain (Ig), short basic dom	SEMA3C	chr7q21-q31	ENSG00000075223
AB000381_s_	chr8:143913218-143925264 (+) // 99.86 // q24.3	GPI anchored molecule like protein	GML	chr8q24.3	ENSG00000104499
AB000409_at	chr1:46795677-46842496 (-) // 99.92 // p33	MAP kinase interacting serine/threonine kinase 1	MKNK1	chr1p33	ENSG00000079277
AB000410_s_	chr3:9766707-9774087 (+) // 99.49 // p25.3	8-oxoguanine DNA glycosylase	OGG1	chr3p26.2	ENSG00000114026
AB000449_at	chr14:96333467-96417693 (+) // 97.05 // q32.2	vaccinia related kinase 1	VRK1	chr14q32	ENSG00000100749
AB000450_at	chr2:58127335-58240507 (+) // 99.84 // p16.1	vaccinia related kinase 2	VRK2	chr2p16-p15	ENSG00000028116
AB000460_at	chr4:2596956-2704077 (+) // 96.73 // p16.3	chromosome 4 open reading frame 8	C4orf8	chr4p16.3	ENSG00000125386
AB000462_at	chr4:2790338-2810142 (+) // 68.81 // p16.3	SH3-domain binding protein 2	SH3BP2	chr4p16.3	---
AB000464_at	chr4:2907075-2922577 (+) // 77.51 // p16.3	chromosome 4 open reading frame 10	C4orf10	chr4p16.3	---
AB000466_at	chr4:2907075-2913394 (+) // 73.32 // p16.3	chromosome 4 open reading frame 10	C4orf10	chr4p16.3	---
AB000467_at	chr4:2910333-2928249 (-) // 100.0 // p16.3	chromosome 4 open reading frame 9	C4orf9	chr4p16.3	ENSG00000087269
AB000468_at	chr1:229223741-229226572 (+) // 93.87 // q42.2 /// chr4:2440643-248	ring finger protein 4	RNF4	chr4p16.3	ENSG00000063978
AB000584_at	chr19:18357967-18360987 (+) // 99.58 // p13.11	growth differentiation factor 15	GDF15	chr19p13.1-13.2	ENSG00000130513
AB000816_s_	chr11:13313020-13353858 (+) // 94.34 // p15.2	aryl hydrocarbon receptor nuclear translocator-like	ARNTL	chr11p15	ENSG00000133794
AB000895_at	chr11:6617710-6618121 (-) // 100.0 // p15.4	dachsous 1 (Drosophila)	DCHS1	chr11p15.4	ENSG00000166341

Figure 4: A minimal excerpt from the format conversion file of Affymetrix™ (Affymetrix, 2000)

.

each sequence. These are called perfect-match and mismatch probes and the calculated value from all these samples represents the gene expression.

The Affymetrix results and files do NOT come with public identifiers for the genes, but with Affymetrix internal unique IDs mapped to the official nomenclature. A mapping file is provided by the company, including various additional information extractable from the file. The data integration in part 7 uses RNA data from those kind of arrays for calculations and predictions.

- **cDna microarrays**
  The cDna system an alternative to oligo arrays. It has been first established by (Liang & Pardee, 1992) and further developed in the last years. This thesis does not use data originating from cDna microarrays, so literature on principles of cDna microarrays is left to the interested reader under (Knippers, 2001) and (Liang & Pardee, 1992).

As given above microarray results strongly depend on the experimental setup making it a hard task to integrate gene expression data automatically.
Further problems addressing current challenges in microarray technologies are available in e.g. (Microarrays, 2006). The problems of gene expression experiments (like noise or the throughput) is subject of intensive research in the area of molecular biology or biotechnology.

2.3.2 Protein Protein Interaction

Another important evidence for interrelating gene functionality on the level of proteins are experiments regarding to PPI (protein protein interactions), which are driven experimentally as well as in computational predictions. PPI are based on the information shown in figure 5. The most commonly used techniques are the following.

- **Yeast-Two-Hybrid experiments**
  This is the classical experiment for detecting interactions on protein level, originally pioneered by (Fields & Song, 1989). In the meantime, other simliar processes for bacteria etc. were proposed as in (J.K. Joung & Pabo, 2000). A very profound overview on the basics of this topic can be found in the review articles (van Criekinge & Beyaert, 1999) and (Phizicky & Fields, 1995).

  Say we want to test if protein A and B bind each other. The Gal4 protein, a strongly expressed transcription factor in yeast, is taken and 'divided' into two parts. Only these two parts together can act as a transcription factor and express a 'reporter-protein'.
  The clue is the following: after the Gal4 is divided, protein A and B are each attached at one half of the divided Gal4. So if protein A and B bind each other, they act as a glue and

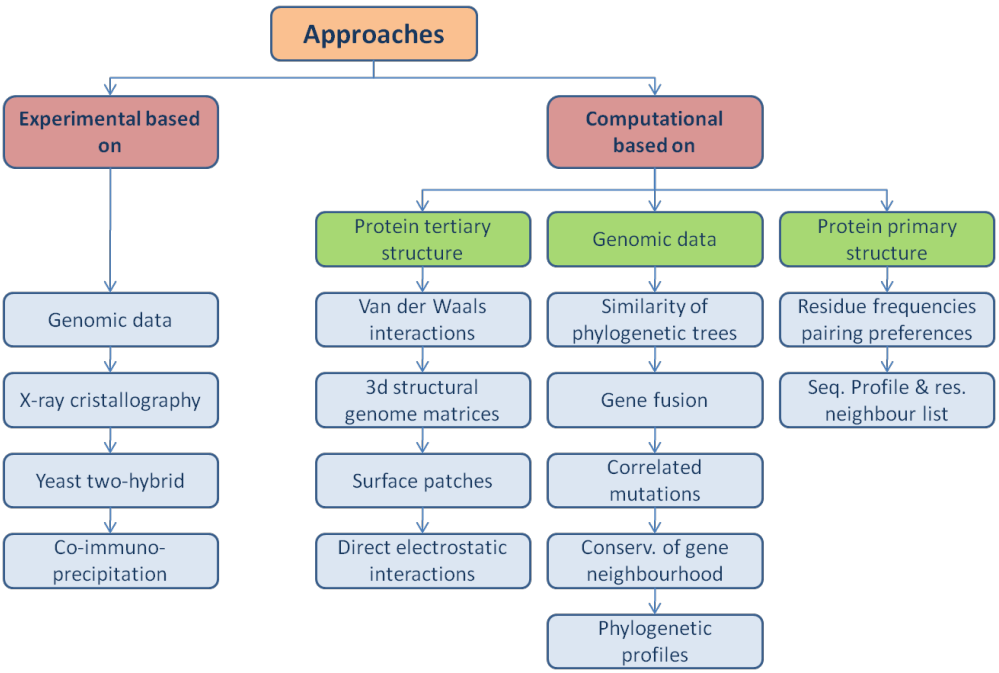

Figure 5: Bases for gaining protein protein interaction data according to Porollo (Porollo, 2006) based on (A. Valencia, 2002; Marcotte *et al.*, 1999; Li, 2006; Chen, 2005).

combine the splitted Gal4 protein parts and the reporter gene will be expressed, otherwise not. The conclusion now is, that the reporter gene is expressed, because A and B bound each other, and thus completed the Gal4 protein, making it an active transcription factor again.

Most data in the frequently used interaction databases (e.g. (OPHID, 2006)) is derived from Yeast-Two-Hybrid experiments.

- **Computational Predictions**
  In that case the calculations are mostly based on the primary, secondary and tertiary structure, or they are gained by mining already existing information and building classifiers with bayesian networks. A detailed overview on techniques is given in (A. Valencia, 2002; Marcotte *et al.*, 1999; Li, 2006; Chen, 2005).

A list of data sources for information on PPI are the (HPRD, 2006; BIOGrid, 2006; OPHID, 2006; Brown & Jurisica, 2005; MIPS, 2006).

2.3.3 Transcription Factors

Research on transcription factor binding sites became very intense over the last few years. Every gene has a region upstream which acts as control sequence for the strength of transcription.
The main idea is that co-regulated genes share similar promoter regions and therefore a similar

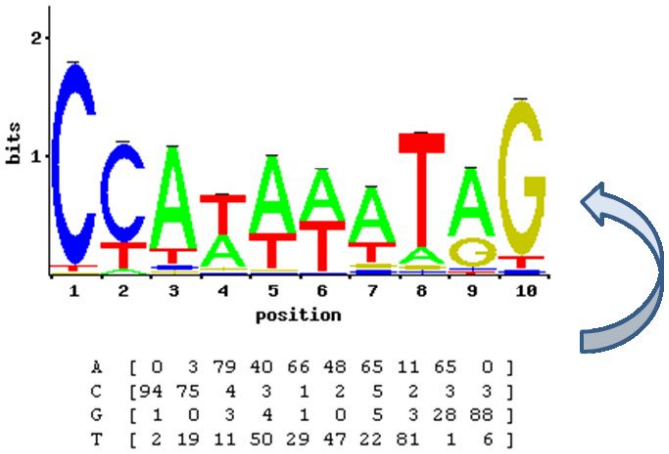

Figure 6: An example for a transcription factor matrix created by JASPAR (Sandelin *et al.*, 2004). The x-axis shows the position on the transcriptions factor binding site while the y-axis counts the occurrences of the DNA alphabet.

behaviour on the level of transcription. The search for promoter regions has been intensively supported by computational methods. The transcription factor data used in our system results from calculations based on promoter motif databases like JASPAR (Sandelin *et al.*, 2004) and TRANSFAQ (Matys *et al.*, 2003). Another work on over-represented transcription factor binding sites in co-expressed genes is oPOSSUM (Ho *et al.*, 2005). The main problem in this area is the extremely high false positive rate, because of short length of promoter regions of about ten to 20 nucleotides. Phylogenetic footprinting is named the golden standard when trying to reduce the false positive rate (D.L. Gumucio & Goodman, 1993). An example on applying a transcription factor to a sequence is given in figure 6.

2.3.4 Gene Ontologies

Gene Ontologies refer to a classification of genes depending on their function, location and process. One of the most popular resources is the (GeneOntology, 2006). However, there are still more resources based on other ontology classifications like the MONET ontology (da Silva *et al.*, 2006). The gene ontology regarding to (Ashburner *et al.*, 2000) is a hierarchy where all genes are classified in a category describing their known characteristics. From a mathematical view, the ontology is a non cyclic directed graph as shown in figure 7. Usually, the deeper the level of the hierarchy in the ontology, the more specific are the genes assigned to this ontology term.
The gene ontologies can give evidence if genes are responsible for similar functions, according (Popescu *et al.*, 2006) it is possible to calculate distances between GO terms (confirmed by wet lab experiments).

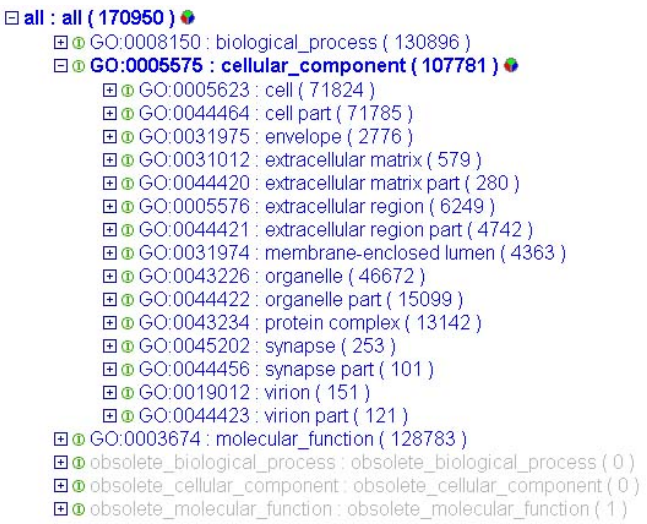

Figure 7: A gene ontology representation of (GeneOntology, 2006).

2.3.5 Systems and Pathway Information

The pathway information is usually already very well defined and experimentally validated. A pathway database holds information on known processes in cells on a functional level. The information can often be requested in a graphical mode, shown in figure 8. At the moment out model contains this information just for verification purposes, but later on those data will be a substantial source, too. The most well known databases for pathways are the (KEGG, 2006) and the (BIOCarta, 2006).

2.3.6 Subcellular Location

The subcellular location is a characteristics of proteins and specifies a protein's location in the cell. The occurrence of the intracellular location of the protein is an important indicator of its function. Usually the physically established experiments are very cumbersome, so a handful prediction tools have been developed like PSORT (Nakai & Horton, 1999), which is the most prominent representative in this family. The trigger in development of those prediction tools was the discovery of the correlation of protein's amino acid composition with its structural and biological characteristics (K. Nishikawa, 1982). PSORT in detail uses amino acid composition and N-terminal targeting signals, as well as sequence motifs to determine the subcellular location of an expressed protein. Other applications are (Bina *et al.*, 1997; Kumar *et al.*, 2000; Kenri *et al.*, 2004).

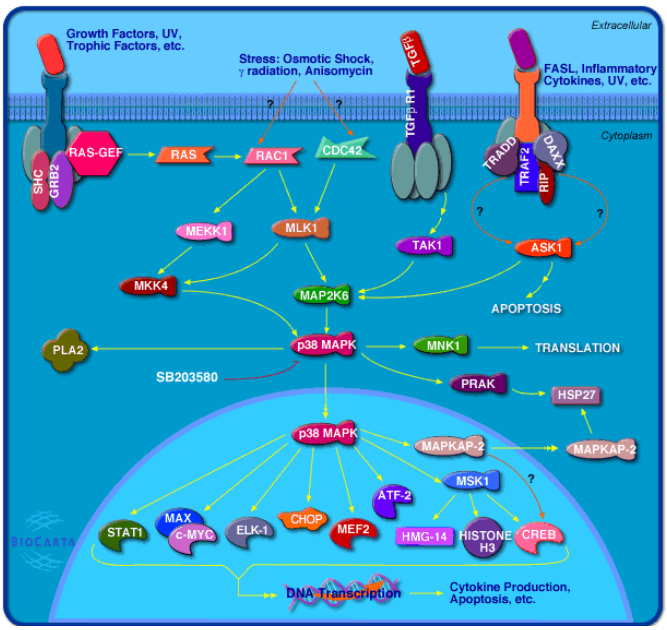

Figure 8: An example pathway from the BioCARTA (BIOCarta, 2006).

2.4 Existing Workflows and Selection Processes

2.4.1 LIMS

LIMS are **L**aboratory **I**nformation **M**anagement **S**ystems, and they came up in the early 1980's. First used for just keeping track of the information created in laboratories, those systems soon became an integrated solution in laboratory environments. A typical LIMS supports a researcher in keeping track of information in the laboratory environment like shown in figure 9, a LabVantage solution (LabVantage, 2006) (LabVantage is a distributor in the area of LIMS in the life science cluster). LIMS can be connected to electronic devices and might be able to directly store information generated by devices, or partly automatize the workflow through handling the device in a half automated-way. **Nevertheless this is not a workflow in this thesis' sense.** A LIMS is a supportive tool for researcher's necessary worksteps in a laboratory. For the approach of this thesis, of course, also information collected by LIMS systems is used, but the main focus will be explained in the next section.

2.4.2 A Common Computational Supported Marker Identification Process

A common approach to identify disease associated marker proteins is described in (Perco *et al.*, 2006) where different levels of information are taken into account on a sequential base. Until not proven experimentally, that a protein or gene represents a disease associated marker, this compound is called a **candidate**.

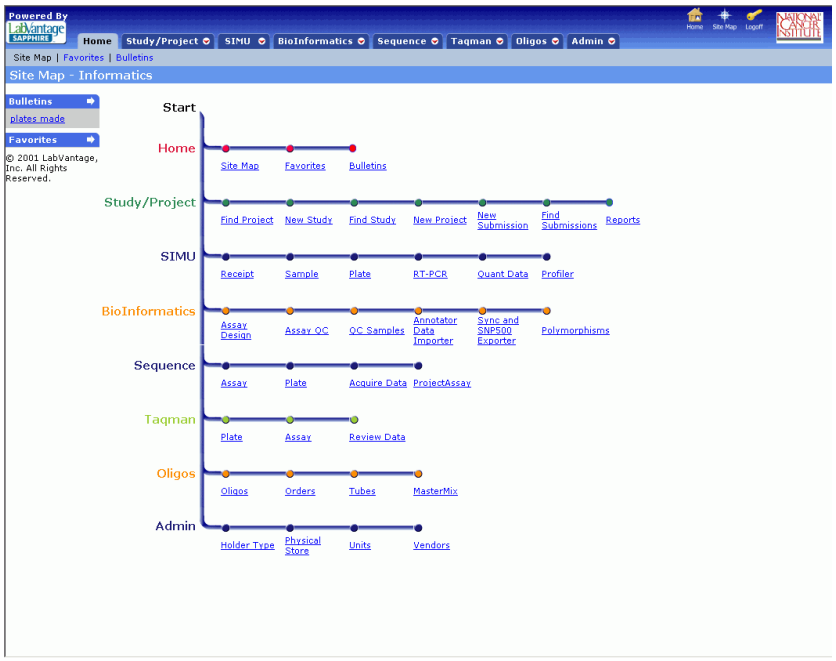

Figure 9: An overview of an example of a LIMS customization in the life science cluster by LabVantage (LabVantage, 2006).

The first thing of a typical -omics workflow is an analysis of raw microarray data - gained by one or more different microarray techniques - where statistical significances of two probes (let's say *Group A* and *Group B*) are examined. *A* and *B* often refer to a Group of 'healthy' and 'diseased' samples' gene expression (but they can also be the same samples evaluated with different treatment). Nevertheless, in both cases the differences between the resulting gene expression of *A* and *B* is of interest. After the significances - of this mostly huge amount of data - have been registered to reduce the data to a starting set of candidates, other analysis steps are taken into account. Figure 10 shows this in a detailed manner. *1* represents an explorative analysis of raw microarray data, the result is a core vector of candidates (*2*). After that co-regulation analysis takes place, enriching the core vector by co-regulated candidates which could be relevant in the context of the core samples. Finally, this set received from *1 and 2* is broadened by a network analysis, using sources like the (OPHID, 2006) or (KEGG, 2006) (*3, 4*). The core vector can be enriched and depleted by candidates during the steps of analysis. But as a result a minimal set of candidates is required, because the necessary validation of the identified candidates is very expensive and time consuming. In every analysis step candidates are filtered out by certain constraints until only a few candidates, the final pathway *5*, remain. This method has some major drawbacks:

- the initial building of the core vector filters too many possible canditates.

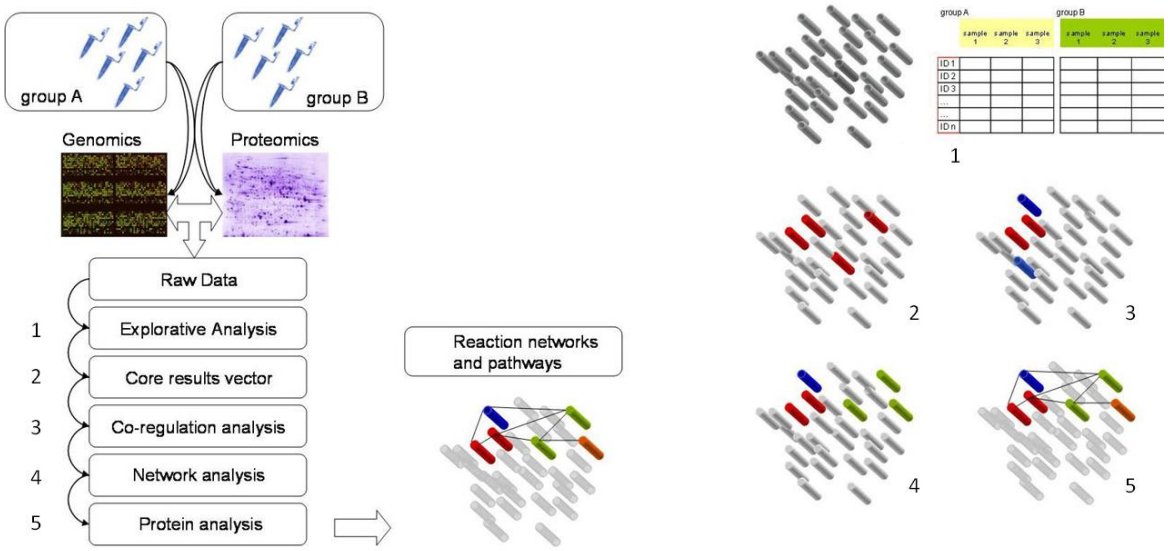

Figure 10: A common selection process for the identification of disease associated markers (Perco *et al.*, 2006).

- always when a candidate is filtered out in one step, a whole possible chain of candidates might be lost, just because of the considerations of one analysis step.
- disqualified candidates are likely not to be considered again.

One important thing to say is that data sets analyzed in each step are extremely incomplete, it is very likely that candidates are filtered out more often because of data gaps and not because of the fact, that they do not fit in a data profile. The workflow of course strongly depends on the techniques used, this workflow shall just act as an example. For a collection on methodologies used for the identification of markers based on computational biology, see (Ambesi-Impiombato & di Bernardo, 2006).

2.4.3 STRING - View All at Once

STRING (von Mering *et al.*, 2005), developed at the (EMBL, 2006), aims to give a integrative view of gene related data. Among (KEGG, 2006) and (BOND, a) the tool has a broad base of information to represent, filters GeneIDs for redundancy and creates data *de novo* by comparing genomes and searching them for conserved genomic neighbourhood, gene fusion events, and co-occurrence of genes across genomes. For a later step it would be very attractive to let the assessment of the edges from STRING flow into our system model. Mering could prove the enhance of quality of the data, by having a scoring method for the interactions.

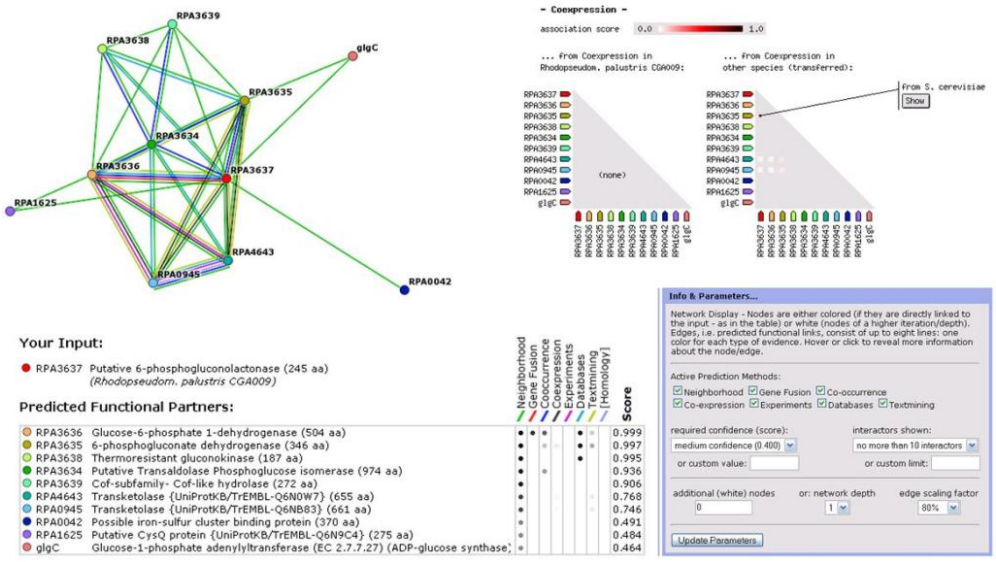

Figure 11: An output from the STRING tool. The tool assesses various informations at once and 'weights' the edges between genes (von Mering *et al.*, 2005). Additional information to the graph is also available.

Overall, it is a tool for representing collected information, but not a solution for a selection workflow for biomarkers.

2.4.4 Subsumption

STRING is not adequate for the search of disease associated marker proteins with selection workflow characteristics, because it does not *filter* data heaps, it just assesses them and represents collected information in a structured and graphical way. All data is assessed at once, so nothing is lost, but in a filtering workflow the focus of interest is also the **exclusion of non-relevant information**, concretely the reduction of the **false positive rate**.
The common workflows (like proposed by (Perco *et al.*, 2006)) have major drawbacks as shown right above which are addressed in this thesis approach.

Thesis Statement 3 *The aim of this thesis is to*

- *combine a selection workflow like (Perco* et al.*, 2006) and a parallel assessment as in (von Mering* et al.*, 2005) and*
- *reduce the false positive ratio*
- *without excluding candidates or chains of connection until all possible information is evaluated and the most likely configuration is found*

3 Emergent Properties and Dynamical Hierarchies

3.1 Introduction and General Considerations

The following section ins structured as given below:

- **Introduction to this Section** 3.1
  Gives an overview on emergence and dynamical hierarchies and the link between those concepts.

- **Emergent Properties** 3.2
  Introduces emergent properties, briefly summarizes the history and the State of the Art in research and application of emergent properties. Furthermore concrete examples of emergent properties are given, which will be helpful (especially to clearly separate the topics of emergence and dynamical hierarchies) when structuring the topic in a more fine grained classification. At last an overview to nowadays definitions of emergence is provided.

- **Mathematical Approaches to Formulate Emergent Properties**
  At the beginning this part gives a survey of concepts of dynamical hierarchies and introduces approaches to these definitions in detail (mainly according to the constructors view, described in the emergent properties section 3.2).

Emergence is a ubiquitous principle in nature owning a very philosophical component (which is not the focus of this work), as well as a profound theory for modeling practical issues. A lot of work has been done so far, reaching from the area of complexity theory, chaos theory and evolutionary theories (Rosenberg, 1985; Hartmann, 1940) down to dynamical hierarchies, which establish a mathematical framework for simulating complex systems.

The concept of emergence is also found in artificial life, which in fact has constituted the largest community in doing research on this topic. This community did not just focus on theoretical aspects, but also on real life adaptions (like the implementation of an ant optimization algorithm in package delivery with autonomously acting agents from (Wolf & Holvoet, 2005a), or the application of theories derived from emergent behavior to election algorithms in distributed systems with self-organizing agents (Anthony, 2004)). Other communities in the the NASA and the MIT conducted research on distributed self-organizing systems and agents. Those groups established models for self-organizing hierarchies in sensor- and communication networks (Prokopenko *et al.*, 2005). One group for example develops autonomous robots for the exploration of difficult terrain. Those robots show a complex behavior in bulk usage (Dubowsky *et al.*, 2006).

The following sections introduce emergent properties and give an overview of definitions of emergent properties. Finding a way to classify a property definitely as an emergent one by a formal

definition is the most argumentative part in emergent properties research, so this is outlined in detail. Another necessity is the understanding of the circumstances under which emergent properties can actually be used in calculable models for solving practical issues. Therefore, it is essential to break down the complexity of the problem into stable subsystems and avoiding an accumulation of the subsystem's errors. Then only the rules of the dynamics of the subsystem, which form the complex behavior, have to be known. When starting to evaluate models under various settings, only some rules can be taken for granted. Most rules come from literature search and derived hypotheses.
Additionally examples for emergent properties are given to improve the understanding what an emergent property can - and what it can not be.
After studying definitions of emergent properties we will agree on one definition which we will formalize in a mathematical notation as a base of this work.

Another section covers the mathematical notations the different definitions, which arose since the introduction of the concept of emergence. The reason for the manifold different formalisms grounds on the different views on emergent properties, as well as the influence of applications solved by using this method.

3.2 Emergence

Emergent properties have been studied since a long time. According to (Goldstein, 1999), the identity of a 'Gestalt' (since the ancient Greek) conforms to something as a whole, thus representing a pattern not describable as just the sum of its parts. The notations of emergence appeared in the writings of the English philosopher G.H. Lewes in 1875, the first definitions close to formal descriptive ones were introduced and formulated by the German philosopher Nicolai Hartmann with its original term *'categorial novum'* used in his book about the construction of the real world (Hartmann, 1940). The principle of emergence was more philosophical, than beeing a topic studied in natural or technical sciences. Emergence came up with the question of life and self-organizing systems. With the rise of computer science and the possibility of simulation scientists became more interested in this area again and developed simulation systems to study complex behavior, self- organizing systems and applications like cellular automata. Computerized simulation allows scientists to show the dynamics of a crowd of people in panic situations and thus helping to provide better infrastructure. Especially at the Los Alamos National Laboratory a lot of work in this field was done (Rasmussen *et al.*, 1996; B. Mayer, 1998; Baas, 1992).

There are in principle two big communities dealing with emergence. One part focuses more on the philosophical, observing view on emergence, the others more on the constructive, engineering or mathematical view. To simplify this constellation the terms *explorers* for the first group,

constructors for the second one is introduced and used from now on. The *explorers* tend to find explanations to describe real life circumstances - mostly in words and precise verbal definitions. Their major interest is to understand life in a deeper meaning, and to find explanations why certain patterns or phenomena arise, and how they fit into a consistent philosophy of life. They are usually not interested in finding the notion or formalisms for inventing mathematical tools, that give the possibility to start calculating abstracted situations from bottom-up. Explorers leave this part of work to the *constructors*, who identify definitions, notations, formulas and lemma providing the possibility to gain benefit from the experiences and research made and stated by the *explorers*. Of course *constructors* are always *explorers* as well, but the major focus and characteristic of the constructors view is the outcome of a formalized approach, giving the possibility to perform calculations, mostly in a reduced scope. Nevertheless, both views give highly important input for opening the chance to take advantage of these insights. One important thing is not to mix up these two positions. Of course, both views in their ideal definition should meet each other, but at the moment it is most likely, that the constructive view only covers a restricted scope of overall emergence, because emergence itself is a complex topic and simulations of more sophisticated models often go far beyond the limits of today's computer powers.

3.2.1 An Introduction on Emergence

Emergent properties relate to **complex pattern formation evolving from more basic constituent parts or behavior**. This complex pattern could take shape in a formation like a large number of neurons forming a brain capable of the complex behavior of thinking or in the process of genetic evolution.
A gas particle for instance has several physical properties like momentum, its position etc., however, the phenomena sound can not be observed for one single molecule. In accumulation it is suddenly possible to give the new construct a not expected property. In the definitions later on, the term 'not expected' will take a key role for the chosen definition. These properties are irreducible and can not be observed on a lower level.
Like the formation of pattern in emergent structures in nature when talking about genes and cells, no one would conclude the structure skin from just observing the behavior of a single cell. It can be a sensor for temperature, air pressure and much more. So an emergent property can also be considered as a property, which makes 'sense' on a higher level.
Another good example is color. It is impossible to find this property in electrons or protons, but at a larger scale like atoms light is emitted in different wavelength and allows us to define a color.

The path finding system of ants for example does not make sense for a single ant. However, if there is a large colony it always describes a good way to a food source. Ants are consequently spreading pheromones. Usually other ants walk randomly, but if they notice a pheromone trail

they tend to follow it with increased probability, thus making the intensity of the pheromone trail stronger. If the food at the end of this trail is transported away, the ants do not return the same way and by that the intensity of the trail gets weaker and another routes are marked. For a biologist (when examining the ant's food organization behavior) it is possible to interpret a function, a common sense in this pattern. The ants do not spread those pheromones on individual purpose, the phenomena of ant pathfinding is an evolutionary concept helping the ants to survive. A multi-agent implementation of an 'Automatic Guided Vehicle System' (a corporation between the company 'Savant Automation' and the Dept. Of Computer Science KU Leuven, Belgium) can be found (De Wolf *et al.*, 2002) and an exact tract on ant-based pathfinding algorithms is worked out in (Gordon, 2003). On the general part on how to engineer emergent properties and self-organization on a model based view more research has been conducted by De Wolf et al. (Wolf *et al.*, 2005; Wolf & Holvoet, 2006; De Wolf *et al.*, 2002).

Applications of these principles from emergence are widely spread. Anthony (Anthony, 2004) calls emergence a paradigm when talking about the development of robust and scalable distributed applications, or scientific applications for models of self-replicating cells capable of self-maintenance (Ono & Ikegami, 1999). A good collection of methodologies using hyperstructures and taking advantage of emergence in the internet search was compiled by Qiu (Qiu, 2004). An excellent and very comprehensive state-of-the-art on the occurrences of emergence in areas of computer science, robotics and artificial intelligence was done by (Nitschke, 2005).

The rule 'complexity comes from simplicity' implies an at least geometric growth of calculation afford for a linear growth of components. For already simple rules the application can result in a nearly infinite state space. But one advantage in emergent systems is the occurrence of higher-level rules which seem to be stable again, so that the rules can be used without the necessity to consider lower-level behavior. For a certain scope and restricted environments, it is sufficient to take higher-level observations for granted view underlying levels as a 'black-box'. It is not always required to recalculate all underlying details in a system, if these underlying properties do not influence the behavior on higher levels in the given environment.

3.2.2 Definitions of Emergent Properties

The Journal Artificial Life is one major journal for researchers on this topic. Especially Lenaerts (Lenaerts *et al.*, 2005; Lenaerts & Gross, 2002) was one of the driving forces, when trying to formalize emergent phenomena, no question these definitions of emergent properties are consequently very closely related to the definition of dynamical hierarchies (and thus to the *constructors* view). It makes sense to give the interpretation of an emergent property in a way to allow an easy formulation and mathematical representation. But the definitions on emergent properties and their mathematical

notations are very ambiguous.

The first reason for such an ambiguity is that researchers have no common agreement in deciding if a phenomena is an emergent property or not. Definitions of (Baas, 1992) require an observer to to form an emergent property. This case could be defined in an observation function. For example a pile of sandgrains as discussed by (McGregor & Fernando, 2005). Of course a pile of sandgrains can possess 'novel' properties, but can they be defined as necessarily emergent properties? The choice, of what is an emergent property or not, will have a remarkable influence on the possibilities of the defined formal mathematical system and in consequence the calculation of the model.

The second reason for the different definitions of emergent properties is the different interpretation of the property as a process or a structure.

The third reason why different mathematical formalisms for a picture of reality may exist is a formal one. Ambiguity may come from a different notation for the same problem. This is comparable to the different descriptions of one and the same problem in e.g. graph theory and algebra. Both formalisms can sometimes describe the same problems, but they have different strength and weaknesses to solve the problems. One formalism might be better to lead a proof, and the other one might be better for an algorithmic calculation, due to e.g. efficiency reasons.

To summarize the reasons for ambiguity in formal definitions from above:

- **'What is an emergent property?'**
  Researches have different opinions which phenomena represent emergent properties.
- **Emergent properties can be seen as process or structure**
  The need to find mathematical definitions describing emergent properties established in processes or in structures.
- **Different mathematical notations for the same problems**
  One and the same problem can be defined by different notations.

This turns out as a problem, because literature in emergence at this time has not stated this situation in clear words until now and thus not found a common formalization. So the question is, if there can be found one definition, which is sufficient for all forms of emergent properties. Anyway, this work is not interested in finding an overall definition for emergent properties, but to find a way to adapt the dynamical hierarchies system to a special workflow process in biotechnology. To come to a definition of dynamical hierarchies fitting our needs, the following paragraphs will now deal with different definitions of emergence first.

Definitions by De Wolf (Wolf & Holvoet, 2005a).
De Wolf picked out the major characteristics of emergence occurring in today's literature. Many of his referenced papers have analyzed the topic from an *explorers* view. Subsequent characteristics have been taken from De Wolf's work. It seems that modern literature found a more and more consistent way for a description. However, the actual conclusion of his studies - the completion of emergence and self-organization - is not adopted (respectively not of interest). We just attempt to excerpt characteristics and definitions of emergence which have been found to accord to this work.

- *The Micro-Macro Effect*
  Overall this is the most important characteristic, and mentioned in nearly all articles considering emergence. Phenomena arise in higher-level structures as a result from (inter)actions at a lower-level in the system. This interaction between higher-level systems and lower-level system is called the micro-macro effect.

- *Radical Novelty*
  This characteristic is linked to the micro-macro effect, because it already uses the terminology of levels, meaning that macro-level behavior can not be expected (McGregor especially concentrated on the issue of 'novelty' (McGregor & Fernando, 2005) studies this property in a statistical way) from observations (observation in this sense is a term borrowed from Baas (Baas & Emmeche, 1997)) on micro-levels, even though the information about the possibility of this behavior is already implicitly contained in the micro-level parts. A very expressive quote, based on Bar-Yam's Book (Bar-Yam, 1997) explains this in short words:

  > *"For many, the concept of emergent behavior means that the behavior is not captured by the behavior of the parts. This is a serious misunderstanding. It arises because the collective behavior is not readily understood from the behavior of the parts. The collective behavior is, however, contained in the behavior of the parts if they are studied in the context in which they are found."*
  >
  > *"...emergent properties cannot be studied by physically taking a system apart and looking at the parts (reductionism). However, they can be studied by looking at each of the parts in the context of the system as a whole."*

- *Coherence*
  The coherent whole forming an identity over time, consisting of (interactions of) parts.

- *Interacting Parts*
  The parts interact in some way, pure parallelism is not enough.

- *Dynamical*
  Emergence arises as a phenomenon as the system evolves over time. In this case the emergent

behavior is seen as a process, building patterns or showing behavior identifiable as a pattern over time.

- *Decentralized Control*
  No central control mechanisms occur, control of the system is established in a local manner. Local control is conscious, local actions are taken through direct influence, but the whole is not directly controllable. Centralized control is per definition not possible in this case, because in that situation the behavior would not be an evolving one from the parts anymore, but a planned global one. This is already implied by radical novelty.

- *Two-Way Link*
  Another point is not only the micro-macro effect, but also the one vice versa. Actually, very often the micro-macro characteristic is said to be bidirectional, the whole (or macro entities) are influencing the behavior of the single entities, imaginable as feedback loops.

- *Robustness and Flexibility and Scalability*
  Like mentioned above in the aspects of coherence and decentralized control, a decentralized control (distributed over many single entities) does not lack the risk of a single point of failure, explanations are very similar to distributed systems, which implies also high scalability and flexibility (Tanenbaum & van Steen, 2003).

Definitions by Lenaerts et al. (Lenaerts & Gross, 2002).
More examples on definitions of emergence are in Lenaerts et al. (Lenaerts & Gross, 2002), intensively taking the challenge to establish a profound explanation for dynamical hierarchies. Lenaerts et al. try to find tentative definitions of emergence to distinguish different hierarchy levels from one another. Unlike de Wolf et al., Lenaerts et al. do not focus on finding general characteristics, but more on hierarchies and novelty (characteristics by de Wolf et al. represent both) and how to explain especially them in a mathematical and well defined way.
Lenaerts et al. bring some examples of hierarchial constructs, as the hierarchial structure and dynamical behavior of monomers and polymers. But after all, in his opinion there are - especially in the field of artificial life - concepts which

> "...seem to have a definite reality, but also successfully evade any attempts of definition."

. Lenaerts et al. restrict his definitions to the area of agent based models to get a working definition.

- **Tentative Definition 1** *If two or more objects, of which at least one is of order N-1, but none is of order higher than N-1, engage in indirect or direct interaction with one another, then they form an object of order N.*

- **Tentative Definition 2** *An object A is of order N if it is an assembly of directly or indirectly interacting lower-order objects of which at least one is of order N-1 and if A has a new kind of property that cannot be found at lower-order objects.*
- **Tentative Definition 3** *An object A is of order N if it is an assembly of directly or indirectly interacting lower-order objects of which at least one is of order N-1 and if A can engage in at least one new type of interaction in which objects of order $< N$ cannot engage.*
- **Tentative Definition 4** *AB is an object of order N, if*
  - *A, B are of order $< N$,*
  - *A or B are of order N-1,*
  - *AB has a property that cannot be found at lower-order objects,*
  - *At least one element of η or ω does not label an interaction in which subcomponents of A or B are engaging, whereas $\eta = \{$set of all indices that label active types of input interactions between A and B$\}$ and $\omega = \{$set of all indices that label active types of output interactions between A and B$\}$.*

As one conclusion Lenaerts et al. state that fundamental entities seem to be simpler for description than higher-order structures, and that object complexity rises with the level of emergence. Important to notice is the fact, that Lenaerts et al. clearly differentiate between *object* complexity and the complexity of *behavior*. In the case of *object complexity* they find different perspectives on complexity. In the latter one the growing complexity of systems containing higher levels sounds intuitive, because the number of possible feedback-loops in the system is rising.

Definitions by Baas et al. (Baas, 1992; Baas & Emmeche, 1997).
Baas was very active in publishing articles, from both, the *constructors* and the *explorers* view. His concepts strongly influenced the work done by Rasmussen and Mayer in the area of hyperstructures later on. Going away from the idea of seeing pure reductionism in the concept of emergence, and purely deductive explanations, he and Emmeche suggest, that the explanations themselves can be seen as emergent explanations, thus finding better methodologies to work with the phenomena of complex systems, shifting the topic into the area of philosophy. They emphasize themselves that they try to (Baas & Emmeche, 1997)

> *"...present a framework that may help to circumvent a pure dilemma of reductionism and holism - of gaining knowledge and retaining the richness of a world of emergent structures."*

Therefore Baas introduces the concept of hyperstructues and the notation of an observer as defined in (Baas, 1992). Details on the notation of hyperstructures can be found below in section 3.3 Mathematical Concepts to Formalize Emergence. A good example for an emergent property given by Baas

in (Baas & Emmeche, 1997) is the temperature in thermodynamics. From a reductionistic point of view the temperature is well understood in physics, but it was introduced as an 'ad-hoc' parameter. A more comprehensive explanation on this would then cover the whole process, namely the laws of thermodynamics derived from the laws of Hamiltonian mechanics.
Hence some special definitions - including one general approach of de Wolf listing general characteristics of dozens of definitions and Baas' approach (which actually builds the base for the chosen mathematical framework for this work) - have been presented now. These verbal definitions now instantaneously lead us to a definition of a more formalized representation, expressible by machine languages, the dynamical hierarchies.

3.3 Mathematical Concepts to Formalize Emergence

3.3.1 General Issues

Dynamical Hierarchies have been studied - especially in the area of artificial life - for the past 20 years, and different mathematical representations have been identified. (Lenaerts *et al.*, 2005) delivers insight into the most important publications on dynamical hierarchies and states, that there are rarely simulations where more than one level of emergence is demonstrable, but also that a complete synthetic framework

> *"...will not only provide an understanding of the organization and origin of the complexity in biological systems, but also influence all fields that have adopted biological theories or appeal to some form of emergence to create complexity out of simplicity."*

The problem lies in representation. As in the concept of dynamical hierarchies in (Rasmussen *et al.*, 1996), where objects are very central, it is not a bijective representation of nature. Because objects on a lower-level view probably do not have the same properties on a higher-level view anymore or vice versa. Significant research is done in defining different level-structures, and under which circumstances new levels emerge. The Artificial Life VIII Workshop in Sydney (Bilotta *et al.*, 2002) dealt especially with the theories how complexity evolves from simplicity. In the Journal Artificial Life Volume 11, a profound collection on articles can be found that cover the trail towards formal frameworks of the definition of dynamical hierarchies. They all deal with levels of emergence, self assembly, self-organization or dynamical hierarchies and give a very good insight into the research adapted to this topic over the last years, the relevant articles regarding to dynamical hierarchies outlined below:

- (Prokopenko *et al.*, 2005) deal with practical issues, the research on a self-organizing system - called the 'Ageless Aerospace Vehicle Project' - capable of detecting defects on a surface and repairing itself. With the definition of *impact-boundaries* and the communication by *impact-networks* connecting remote cells they are able to detect the impacts on the surface. This

scenario involves emergent and complex behavior manifesting in the behavior of the problem solving process through cell communication. Shannon's entropy measurement (Shannon, 1948) is then used to evaluate, if the dynamic behavior is chaotic or stable. Lenaerts and McGregor (Lenaerts & Gross, 2002; McGregor & Fernando, 2005) criticize this - at least the definition part of the emergent properties - because of the 'triviality' of the higher-level properties. They state that a higher-level emergent property has to be unpredictable from the perspective of the lower properties.

- The central topic of (Watson & Pollack, 2005) is the (short term) independence of subsystems of a system. When starting a system it might take some time, until an action in subsystem A shows effect in subsystem B. For this timespan it is possible to handle these systems independently. Watson et al. examined this behavior of evolvability in systems. Furthermore they dealt with the scope of systems. A subsystem might have something like a local and stable scope, which makes it possible to define rules for the subsystem without knowing about the whole system, or the other subsystem. In the scope of the subsystem calculations done by the definitions of this subsystem are always correct under certain preconditions. Nevertheless, in that case still unexplainable situations can arise when subsystems interact with each other, because these would be interactions on a higher-level forming a higher-level system. This is then said to be the *modular interdependence*. Based on the work about evolution of complexity in modular systems and the concept of nearly decomposable systems by (Simon, 1996) and (Simon, 1973). Watson et al. discuss these overall system effects.

- The next article (McGregor & Fernando, 2005) discusses an alternative approach to the hyperstructure model proposed by Rasmussen et al. (Rasmussen *et al.*, 1996), Mayer et al. (B. Mayer, 1998) and Baas et a.(Baas, 1992). The clue about this newly suggested model is the definition of emergence of new hierarchies through information theory. McGregor et al. see problems in adapting the hyperstructure model to some special situations. The definition of dynamical hierarchies by information theory traces back to the work of (Dorin & McCormack, 2002). Dorin et al. try to find information based indications in new levels of hierarchies. Shannon's information theory (Shannon, 1948) seemed to be a proper approach when assuming that new levels must contain unexpected properties.

- Rowe et al. (Rowe *et al.*, 2005) again tried to find methods how to group interacting elements (basic units) in dynamical higher-level macroscopic states in a way that is compatible with the dynamics of the underlying system. Rowe et al. found a way to formalize this theories and, presumed linear dynamics, they proved the appearance of this effects using the algebraic group theory. Finally Rowe et al. present some examples and occurrances in artificial life where this behavour is observable - in a very figurative and comprehensible way.

- The last and final contribution (Jacobi, 2005) focused strongly on the relations between the

degrees of freedom (describing the different levels) and the system dynamics. Jacobi et al. try to define hierarchial structures in dynamical systems and to find definitions for the allowance of multiple simultaneous levels of descriptions. Jacobi et al. found definitions expressing *levels within hierarchies* and procedures that can identify such levels. Tools helping to work with these so called *'smooth'* hierarchies (based on *projective maps*) are presented. At last Jacobi draws an interesting correlation between interaction networks and hierarchies. A network representation can also be a valid view on complex systems and Jacobi tried to embed this perspective as one step in the definition process of dynamical hierarchies.

Besides from dynamical hierarchies several other formal mathematical models find their application in the area of biology, just to mention the *Matrix Formalism to Describe Functional States of Transcriptional Regulatory Systems* in Gianchandani's work (Gianchandani *et al.*, 2006) describing 'switches' in the transcriptional regulatory system by the use of boolean algebra. Then a formalism is found transforming the system into a matrix, which then represents the regulatory rules of transcription. Having a matrix perspective on the transcriptional regulatory system of a cell allows a systemic characterization of the system and the computational process of determining the transcriptional state of the genome.
Zambronelli et al. tried to find a model-driven approach in developing multi agent systems and to clarify a concept in this process. The perspective on emergence is strongly practical and influenced by software engineering. A formalization of mathematical concepts not covered, but a associations - between software engineering and very dynamic, proactive and autonomous multi agent systems - are drawn. This is a very similar context, but Zambronelli et al. actually does not care about the definitions of emergence, or the definition of a formalism, although those topics are strongly related.

As an even more restricted model (for engineering purposes) in the area of autonomous agents, de Wolf brings up suggestions to formulate the micro-macro behavior in the Unified Modeling Language UML (by the OMG (OMG, 2006)). Integrating the important aspect of finding models into an engineering process, making it partly possible to automatize parts of the implementation of such systems, this research area is of high relevance as well. Details can be obtained from (Wolf & Holvoet, 2006; Wolf & Holvoet, 2005b).

But what this thesis is searching for is a more formalized, integrated solution. A solution considering more than just one layer of information, giving the answer in the application of dynamical hierarchies to -omics data and using biological knowledge to fill these data's lack of information. We aim to find mechanisms under which the information gained from the evolved system is optimally exploited.

As the definition in differentiating between the levels of emergence has been taken in the sec-

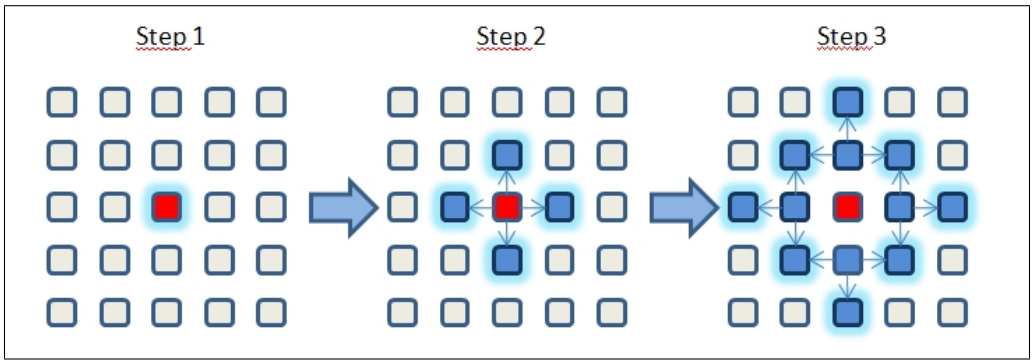

Figure 12: A simple cellular automata

tion Emergent Properties 3.2, now the choice of the definition of dynamical hierarchies to work with has to be taken. The definitions used for representing emergent behavior in mathematical models differ between each other. We decided to use the concept of dynamical hierarchies, because the thought of objects and states can be embedded in an easy way in that case. Dynamical hierarchies can be seen as a discrete dynamical systems similar to multidimensional cellular automata on a n-dimensional lattice. All objects can be all object's neighbors. Each object can potentially influence every other object directly and indirectly and each higher-level compound influences all other compounds (**in an non-reductionistic environment**). The principle of a 2-dimensional cellular automata is shortly presented in figure 12. The function is quite simple. There is a lattice of cells, where the cells can take up different states. The cell's states depend on a predefined starting state set. The actual cell's states are then calculated by the states of its surrounding cells. These set of surrounding cells is defined by a neighborhood function and called the neighborhood. The system propagates over time, updating every cell in a particular order (defined again by the neighborhood function). This model represents a discrete dynamical system. The rules in this example are quite easy and someone can easily imagine what happens with proceeding propagation. Each cell colors all direct neighbors blue for each step. After each cell was 'touched' once, all cells will be blue. This is a very simple example of a complex system. But with already very simple rules complex pattern formation can be observed as in Conway's game of life.

The following sections will give an insight into today's mathematical formulations of emergent properties:

3.3.2 Dynamical Hierarchies and Hyperstructures

For dynamical hierarchies this concept was first introduced by Baas (Baas, 1992) and later found its applications in (Baas & Emmeche, 1997; B. Mayer, 1998; Rasmussen *et al.*, 1996).

Let $\{S_i|i \in I\}$ be a family I of general systems S. Let Obs^1 be an observation mechanism and Int^1 be interactions between systems. The observation mechanism measures the properties of the agents to be used in the interactions. The interactions then generate a new kind of structure

$$S^2 = R(S_i^1, Obs^1, Int^1)$$

which is the result of the structure. S^2 is an *emergent structure* which may be subject to new observational mechanisms Obs^2. This leads to

Definition 1 *P is an emergent property if*

$$P \in Obs^2(S^2) \text{ and } P \notin Obs^2(S_i^1).$$

Somewhat eye-catching is the circumstance, that Baas explicitly emphasizes *internal or external* occurrence of observational mechanisms, but states, that both can take the place of an observation function. For a useful representation Baas suggests *category theoretical* notation, where the objects in a category are systems and interactions between objects are morphisms.

Baas differentiates between two types of observations:

1. **Deductible or computational emergence**
   Which means, that deductional processes can be represented by the formal theory shown above.

2. **Observational emergence**
   There exists an emergent property, but the process can not be understood as in deductible emergence. As example Baas mentions the Gödel sentences or the property of the membership of Mandelbrot sets..

Definition 2 *A **hyperstructure** of order N is given by*

$$S^N = R(S_{i_{N-1}}^{N-1}, Obs^{N-1}, Int^{N-1}, S_{i_{N-2}}^{N-2}, ...)$$

For further research done on these algebraic structures see (Lygeros & Vougiouklis, 2005; Bayon & Lygeros, 2006). Baas then continues to describe the function of the observer in a tentative way, followed by formulating emergence through *category theory* in detail showing aspects how to differentiate between reductionism and non-reductionism (emergence or holism) with category theoretical arguments. Just to be mentioned, Baas refers to the topic of formal logics applying this concept for deduction in classical logics. The final part holds an ongoing discussion around the ideas in emergence in a more general and prospective way which will be omitted at this point, because it is of no further necessity for the definition of hyperstructures.

At this point Mayer et al. (B. Mayer, 1998) hooked in with a model for an object hierarchical representation of the definitions mentioned above. Proposing a model for the self-reproduction of dynamical hierarchies in chemical systems, Baas' definitions of hyperstructures were used to study emergent properties of molecules. In detail they present objects on three different hierarchies, namely

- Monomers - 1^{st} level
- Polymers - 2^{nd} level
- Micelles - 3^{rd} level

structures, all of them having different emerging properties arising. This system was then implemented in form of a *lattice molecular automata* - shortly called a **LMA** - to examine the properties of this system. They were then able to identify emergent properties when starting the simulation with simple propagation rules. Mayer et al. referred to **time** $t = 0$ when starting to propagate the system and each step evolved the system by $t + 1$. However, it is important to notice, that **the time parameter is a virtual one in this thesis' model**, because there is no evolving system over time, just an attempt to let incomplete information evolve until the data profile shows systematic patterns representing functional gene networks. In this context the term 'time' is better substituted by the term **'evolvement'** measured in **'iterations'**.
The hyperstructure model of Mayer et al. is very specific, highly tuned to a molecular dynamics context, but nevertheless very suitable to derive a generalized model. The hyperstructure's objects hold different information describing the system, respectively describing the other objects states. In figure 13 this hyperstructure model is shown, as it was used in (B. Mayer, 1998) and the following paragraph explains the single properties used in this structure:

1. $X_1 = \{x_1\}; x_1 \in N_0$; type-state:
   molecular types (including vacuum) at site (i,j).
2. $X_2 = \{x_{2,1}, ..., x_{2,8}\}; x_{2,l} \in N_0$; rec-type:
   molecular types (including vacuum) in the neighborhood of site (i,j).
3. $X_3 = \{x_{3,1}, ..., x_{3,8}\}; x_{3,l} \in Z$; send-state:
   outgoing force particles along q lattice directions.
4. $X_4 = \{x_{4,1}, ..., x_{4,8}\}; x_{4,l} \in Z$; rec-state:
   incoming force particles from q lattice directions.
5. $X_5 = \{x_{5,1}, ..., x_{5,8}\}; x_{5,l} \in N_0$; kin-state:
   local kinetic energy at location (i,j) in q directions.

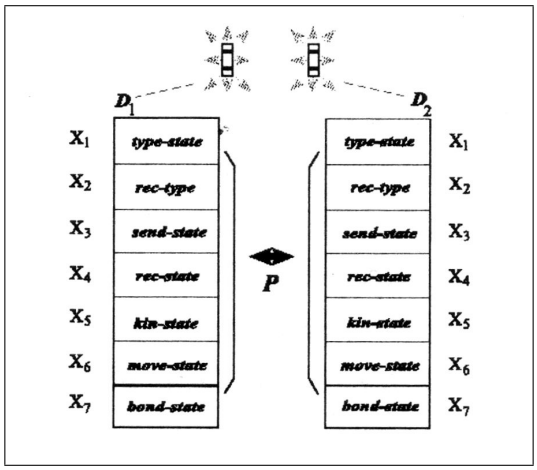

Figure 13: The hyperstructure model used by (B. Mayer, 1998)

6. $X_6 = \{x_{6,1}, ..., x_{6,8}\}; x_{6,l} \in Z$; move-state:
   list of net energetic states (including potential and kinetic energies).

7. $X_7 = \{x_{7,1}, ..., x_{7,8}\}; x_{7,l} \in \{0, 1\}$; bond-state:
   maintain bonds with polymers.

With this model Mayer et al. evoke complex behavior without implementing it explicitly. They found emerging properties and complex patterns which arise from simple rules.

3.3.3 An Approach by Probability

Rowe et al. (Rowe *et al.*, 2005) focused on the aggregations of lower-level units to higher-level units, and if the higher-level units can be reconstructed from the equations describing the lower-level units. In that case the partitioning of the lower-level units is said to be compatible with the higher-level units.
He uses a probability matrix Λ with the following constraint to the matrix:

Definition 1

$$\Lambda = \{x \in \mathbb{R}^{\aleph} : \sum_{k} x_k = 1 \textit{ and } x_k \geq 0 \forall k\}$$

whereas x denotes the column vectors(states) and k denotes the number of rows(action for the next state) in the matrix.

If the dynamics of the system are known, he can then define a mapping $G : \Lambda \rightarrow \Lambda$ from which he can reconstruct the dynamics in a dynamic or stochastic way depending on the underlying problem. After that the state space Ω (all possible states in Λ assumed the state space is finite) is divided into partitions by an defined equivalency relation and following definition is proposed:

Definition 2 *A map $G : R^n \rightarrow R^n$ is said to be compatible with an equivalence relation $\equiv$ if*

$$x \equiv y \Rightarrow G(x) \equiv G(y)$$

$\forall x, y \in \mathbb{R}^{\ltimes}$.

This groupings are then proposed to be emergent properties of the system. Rowe et al. model this system in the context of a linear process respectively a linear Markov process. The equations describing the dynamics through the suggested aggregation technique could be dramatically simplified and systems with higher- and lower-level structures evolved from their implementation simlilar to a black-box definition. Dynamics from higher-level units can be used (if they are in the same equivalency class) without worrying about the microscopic behavior. To emphasize the characteristics it has to be said that the system is *aggregable and decomposable* and a bijective mapping. The state space is well defined and finite in this case.
Rowe et al. proceed to study the compatibility of state aggregation with terms of algebraic group theory. Note that the focus of this work lies on aggregation and classification to the hierarchy and not in defining a working data structure for an application.

3.3.4 An Entropic Approach

McGregor et al. (McGregor & Fernando, 2005) suggested a correction of Baas' and Mayer's framework. McMullin and Gross criticize Mayer's and Rasmussen's work (Gross & McMullin, 2002), because they declare that the hyperstructure formulation allows trivial new properties on higher levels. McGregor et al. compares the third order structures of Mayer and Rasmussen (B. Mayer, 1998) with a pile of sand grains and says that the micelles (the third-order structures) could have been formed from the monomers (first order structures) directly without an interplay of polymers (second-order structures). That makes the micelles actually second level structures. Rasmussen (Rasmussen *et al.*, 2002) counters, that new hyperstructures with no new properties would be trivial anyway and that piles of sand would fall into this category. Like mentioned above, Lenaerts tentative Definitions (Lenaerts & Gross, 2002) deal with this issue and propose that new emergent properties can only arise if there are new interactions between fundamental particles. Here McGregor declares this definition as overexclusive. However, McGregor argues that piles of sand **do have** new properties, but that they might **not be of interest**. That is where McGregor et al. try to compensate the shortcoming of a lack of definition of a 'trivial' property. The approach they aim at is closely related to an observer function, but with an additional aspect. They ask for the 'new' in the property using Shannon's information theorem (Shannon, 1948). The information entropy is used to determine the existence of new properties. In other words: a new property is a property that can not be expected from knowledge of lower levels.

For this thesis the decision fell for the model of hyperstructures by Baas (Baas & Emmeche, 1997; Baas, 1992) and Mayer, Rasmussen (Rasmussen *et al.*, 1996; B. Mayer, 1998), because there is no necessarity to define levels of emergence dynamically. Primarily an object representation fitting the application very well was needed.

Part III
The Approach: Planning of a Solution

4 Basic Approach

4.1 General

This thesis' approach is - as outlined in the previous sections - an attempt to reconstruct biomolecular functional networks from heterogenous data in the concept of dynamical hierarchies. Therefore the method of hyperstructures from section 3.3.2 was chosen to complete this task.

After an exact description on hyperstructures above an analysis of the work of Mayer et al. (B. Mayer, 1998) took place and the key issues of their model were extracted to be able to allow generalization: a **hyperstructure** is an algebraic structure with at least one multivalued operation called hyperoperation. Informally spoken there are objects which hold states about the different *objects' properties*. In the application of a discrete dynamical system these properties' states are then defined by

1. a neighborhood function
   who are the objects influencing the current object's next property states.

2. propagation rules
   how are those selected objects influencing the current object's next property states.

The term 'function' in this context invariably refers to the mathematical meaning and not the programming construct. Besides in the *time parameter* from Mayer et al. (B. Mayer, 1998) this model differs by the complete absence of spatial aspects, respectively a lattice structure. Our system is absolutely independent from spatial terms. The object's spatial position is not the base measure for the calculation of a distance. From now on the term *edge* will be used to define the distance between any two objects. This *edges* have *weights* (also denoted *strength*).
Overall our system has a similar working principle, but because of the lattice (cell) definition is replaced by an object definition (derived from the concept of Baas (Baas & Emmeche, 1997)), the term *object* is used instead of *cell*.

The main focus of this work is to determine the characteristics of such a system under different conditions. In a first step the implementation was filled and tested with a small set of randomly generated data to assure the correctness of the data structure. Afterwards effective data sets were selected like outlined in section 7 'Empirical Data Selection' and filled into the data structures. Then a state diagram for the system was designed like in figure 14.

At first *imports* of different types of data were performed. The import system allows handling of heterogenous data and to annotate the newly created data entry of the objects with a variable name.

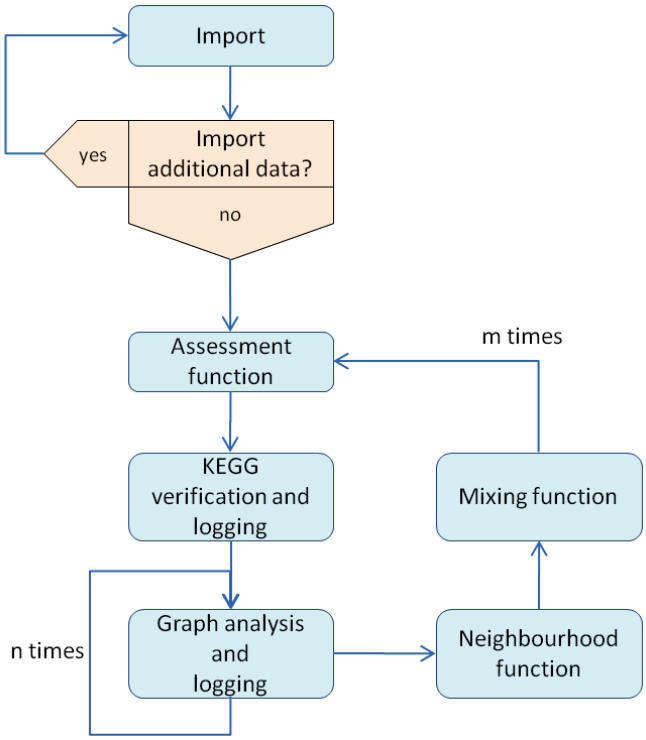

Figure 14: A state diagram representation of the system.

After the import is complete the system adapts an *assessment function* to the objects and builds a data matrix where each matrix entry describes the distance between two objects. Details on the *'Assessment function'* are following in section 4.2.

Now two different steps are inserted. The first one is the *KEGG verification*, where an evaluation on the degree of overlap of the network reconstruction with respect to the interactions contained in chosen KEGG pathways is done. Details are given in section 7.2.1 'Experimental Settings'. The next procedural step is *graph analysis*. The network is evaluated with different *cutoffs* (the granularity of the cutoff can be chosen with the parameter *m*) as described in section 7.2.1 ('Experimental Settings'). Logfiles are written for subsequent analysis. During the *KEGG verification* and the *graph analysis* a variety of information is logged, especially diagrams reflecting the state of the system.

The next procedure applied is a *neighborhood function* determining the update sequence of the objects and which objects are taking part in the *mixing function*. Details are given right below in section 4.3 (*'Neighborhood Function'*).

The *mixing function* is responsible the object alteration. Because of the very strong influence of this function on the system's behaviour, only two objects take part in mixing, and only the way of altering these two objects differ. A description on the different variations can be found in section 4.4 *'Mixing Function'*.

4.2 Assessment Functions

The *assessment function* calculates a distance between two objects. To be concrete, we used information shown in figure 15. Beforehand one important fact should be mentioned, namely the treatment of 'incomplete' information. If an entry set has never been initialized the question arises on how to deal with this fact. For assessment, the way on how to treat interactions where one participating partner had a non existent information data entry are treated, is of importance. This issue is outlining below in the *meta function*.

The concept holds *single functions* which are then combined to a *meta function* f_{meta} afterwards. Until now the following *single functions* influence the *meta function*. In principle all *single functions* scale between $]0, 1[$ and O_1 and O_2 are the two objects assessed.

- $f_{assess_{rna}}(O_1, O_2)$
  For this assessment absolute value of the Pearson coefficient of correlation was taken to evaluate similarity on the level of gene expression. Sometimes the measurement rows of the gene expression samples are not complete or mutually missing. Therefore a *completeness* parameter was introduced. Two measurement rows must have a percentage of valid data entries at the same positions for at least ¿ *completeness*.

- $f_{assess_{OPHID}}(O_1, O_2)$
  Calculate the protein interaction between two objects. If there is an interaction between two objects the result is 1 otherwise it is 0.

- $f_{assess_{PSORT}}(O_1, O_2)$
  Evaluates the fitness between two objects regarding to their intercellular location. The PSORT algorithm gives a certain probability in percent, where a protein could be located in a cell for each location. The sum of all is 100%.

  $$abs\left[a_1 - b_1, a_2 - b_2, ..., a_n - b_n\right]$$

  , where a and b are entries of percentage values of the PSORT prediction.

- $f_{assess_{TF}}(O_1, O_2)$
  Evaluates the fitness between two objects regarding to their transcriptional co-regulation. Calculates the overlap of the transcription factor terms of both objects in the manner

  $$\frac{c}{a+b}$$

  where a is the number of transcription factor terms in the first object, b is the number in the second object and c is the number of occurrences in both.

- $f_{assess_{GO}}(O_1, O_2)$
  The fitness of this function is evaluated like the fitness of $f_{assess_{TF}}(O_1, O_2)$, except that the used terms represent GO identity terms.

- $f_{assess_{KEGG}}(O_1, O_2)$
  This function is **NOT** considered for assessment of the similarity of objects, because we use this information for validaton.

The *single functions* do not necessarily have to be calculated only from GO to GO or RNA to RNA, functions which conclude from one data level to other ones may also be included. The *meta function* combining the results of the different single parts is given as follows:

$$f_{meta}(O_1, O_2) = \frac{f_{assess_{rna}}(O_1, O_2) + f_{assess_{OPHID}}(O_1, O_2) + f_{assess_{PSORT}}(O_1, O_2)}{adjustment} +$$

$$+ \frac{f_{assess_{TF}}(O_1, O_2) + f_{assess_{GO}}(O_1, O_2)}{adjustment}$$

whereas $adjustment$ can vary as explained below.

4.2.1 Assessment function 1 - Division by one

The *single functions* are just added without a scaling parameter which means $adjustment = 1$.

4.2.2 Assessment function 2 - Division by All

In the first case $adjustment$ is taken as the total number of arguments, so its a straight normalization to an interval $]0, 1[$. *Single functions* not calculable are set to zero.

4.2.3 Assessment function 3 - Division by Average Filling

Many object pairs have data lacks and are not calculable. That is why the $adjustment$ is set to the average number of objects holding data. For example, if only one factor out of five can be evaluated the 'Assessment function 2 - Division by All' scales down the interaction between those two objects to a very low value. Even if the (only) calculable *single function* shows a high similarity. As proved in the result section this change influenced the system's behavior substantially. In the case of a not calculable *single function*, it is set to zero as in 'Assessment function 2 - Division by All'. But the scaling value for $adjustment$ for the average filling of five evaluated *single functions* is as obtained as shown below:

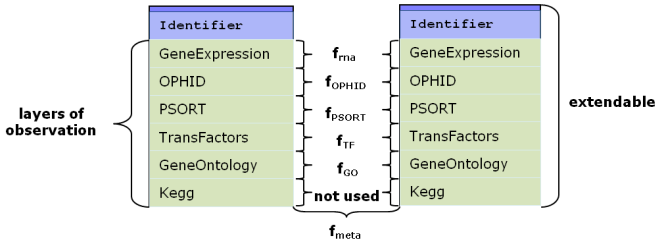

Figure 15: A state diagram representation of the system.

A: No of entries holding data in an object B	No of objects B with A holding data	$A * B$
0	0	0
1	2	2
2	8	16
3	57	171
4	114	456
5	135	675
	$\sum$=316	$\sum$= 1320

So $adjustment = 1320/316 = 4.177 \approx 4.1$.

4.2.4 Assessment function 4 - Division by Evaluated

This function was chosen to observe the systems behavior when omitting non-initialized entries. The *meta function* f_{meta} only adds the values of *single functions* which could be calculated and sets $adjustment$ to the number of *single functions* which could be calculated.

4.3 Neighborhood Functions

This function is responsible for selecting the neighborhood. At the moment only two objects take part in an update step.

4.3.1 Highest Edge Choice 1 - Reconsider Random Failures (RF)

Here a sorted list of all edges - constructed through our *assessment function* $f_{meta}(O_1, O_2)$ - is created with the strongest edge top ranked. The algorithm is as following: Start with the strongest edge and if the value of the edge's strength is smaller than a random number in the interval $]0, 1[$ mark the two participating objects for mixing. Once an object is *effectively altered*, the combination

(O_1, O_i) or (O_2, O_i) will be *altered* again when occurring in another edge later on until mixing and a new assessment takes place.

4.3.2 Highest Edge Choice 2 - No Reconsideration

Here a sorted list of all edges - constructed through our *assessment function* $f_{meta}(O_1, O_2)$ - is created with the strongest edge on top. The algorithm is as following: Start with the strongest edge and if the value of the edge's strength is smaller than a random number in the interval $]0, 1[$ mix the two participating objects. Once an object was *considered for alteration*, (checked against the random number) the combination (O_1, O_i) or (O_2, O_i) will be neither *considered for being altered* nor *altered* again when occurring in another edge later on until mixing and a new assessment takes place.

4.3.3 Incident Rotation Edges - Get Highest then Random - Reconsider RF

For every object the sum of all incident edges is calculated and the results are kept in a sorted list A with the strongest edge on top. For every object in this list starting from top: *get the highest incident edge of this object and if this the value of the edge's strength is smaller than a random number* in the interval $]0, 1[$ mix the two participating objects. Then the next object from list A is taken and so forth. Once an object was *effectively altered*, no combination (O_1, O_i) or (O_2, O_i) will be *altered* again when occurring in another edge later on until mixing and a new assessment takes place.

4.3.4 Incident Rotation Edges - Check Random for All Unsorted - Reconsider RF

For every object the sum of all incident edges is calculated and the results are kept in a sorted list A with the strongest edge on top. For every object in this list starting from top: *get the strongest incident edge of this object which strength is smaller than a random number* in the interval $]0, 1[$ and mix the two participating objects. Then the next object from list A is taken and so forth. Once an object was *effectively altered*, no combination (O_1, O_i) or (O_2, O_i) will be *altered* again when occurring in another edge later on, until mixing and a new assessment takes place.

4.3.5 Incident Rotation Edges - Check Random for All Sorted - Reconsider RF

For every object the sum of all incident edges is calculated and the results are kept in a sorted list A with the strongest edge on top. For every object in this list starting from top: get all incident edges and put them in a sorted list with the strongest edge on top. Then *get the strongest edge from this list which strength is smaller than a random number* in the interval $]0, 1[$ and mix the two participating objects. Then the next object from list A is taken and so forth. Once an object was *effectively altered*, no combination (O_1, O_i) or (O_2, O_i) will be *altered* again when occurring in another edge later on until mixing and a new assessment takes place.

4.3.6 Incident Rotation Edges - Check Random for All Sorted - No Reconsider

For every object the sum of all incident edges is calculated and the results are kept in a sorted list A with the strongest edge on top. For every object in this list starting from top: get all incident edges and put them in a sorted list with the strongest edge on top and process this list top down. *The first edge which strength is smaller than a random number* in the interval $]0, 1[$ is taken to mix the two participating objects. Then the next object from list A is taken and so forth. Once an object was *considered for alteration* (checked against the random number), the combination (O_1, O_i) or (O_2, O_i) will be neither *considered for being altered* nor *altered* again when occurring in another edge later on until mixing and a new assessment takes place.

4.3.7 Incident Rotation Edges - Check Random for All Sorted with Metropolis - No Reconsider

This *neighborhood function* is an extension of *'Incident Rotation Edges - Check Random for All Sorted - No Reconsider'* including the Metropolis criteria (Metropolis *et al.*, 1953). If a chosen edge's value is higher than the random number R_1 in the random number test, an additional chance to accept this edge for mixing is given. A rejected edge is still accepted if another random number R_2 is smaller than $e^{-\frac{\Delta E}{k*T}}$. In a first attempt T is constant with a value of 300, k is the *Boltzmann constant* and $\Delta E = strength_{edge} - R_1$ substracted from the edges value.

4.4 Mixing Functions

The *mixing function* determines how the objects are altered. These *single functions* are **never** applied directly. There is always an *overall function* combining them. At the moment a maximum of two objects take part in a the mixing process with the following behavior:

- $f_{mix_{rna}}(O_1, O_2)$
  This entry holds gene expression data is is not altered.

- $f_{mix_{OPHID}}(O_1, O_2)$
  Build the intersection of the two object's OPHID-entries.

  $$f_{mix_{OPHID}} = O_{1_{OPHID}} \cap O_{2_{OPHID}}$$

  whereas $O_{1_{OPHID}}$ denotes the OPHID-entry of the first object and $O_{2_{OPHID}}$ the one of the second one.

- $f_{mix_{PSORT}}(O_1, O_2)$
  The PSORT-entries of the two object's are taken, the value where they have the highest common percentage is set to 100% and all other values are set to 0%.

- $f_{mix_{TF}}(O_1, O_2)$
  Build the intersection of the two object's transcription factor entries.

  $$f_{mix_{TF}} = O_{1_{TF}} \cap O_{2_{TF}}$$

  whereas $O_{1_{TF}}$ denotes the TF-entry of the first object and $O_{2_{TF}}$ the one of the second.

- $f_{mix_{GO}}(O_1, O_2)$
  Build the intersection of the two object's GO-entries.

  $$f_{mix_{GO}} = O_{1_{GO}} \cap O_{2_{GO}}$$

  whereas $O_{1_{GO}}$ denotes the GO-entry of the first object and $O_{2_{GO}}$ the one of the second.

- $f_{mix_{KEGG}}(O_1, O_2)$
  This function is **NOT** considered for assessment, because we use this information for validation.

- $f_{mix_{AddOPHID}}(O_1, O_2)$
  Add the object IDs of the two object's to each others OPHID-entries.

This was on how the **single** entries are supposed to change. Afterwards **always** an overall altering function is applied. The following methods show how these functions are composed.

4.4.1 Intersect from All

$$f_{mix}(O_1, O_2) = f_{mix_{OPHID}}(O_1, O_2) \circ f_{mix_{PSORT}}(O_1, O_2) \circ$$
$$\circ f_{mix_{TF}}(O_1, O_2) \circ f_{mix_{GO}}(O_1, O_2)$$

where $\circ$ is the associative and commutative composition and O_1 and O_2 are the objects to change.

4.4.2 Intersect from Highest

$$f_{mix}(O_1, O_2) = \begin{cases} f_{mix_{OPHID}}(O_1, O_2), & \text{if } f_{assess_{OPHID}}(O_1, O_2) = max\{f_{assess_{OPHID}}(O_1, O_2), \\ & f_{assess_{PSORT}}(O_1, O_2), f_{assess_{TF}}(O_1, O_2), f_{assess_{GO}}(O_1, O_2)\} \\ f_{mix_{PSORT}}(O_1, O_2), & \text{if } f_{assess_{PSORT}}(O_1, O_2) = max\{f_{assess_{OPHID}}(O_1, O_2), \\ & f_{assess_{PSORT}}(O_1, O_2), f_{assess_{TF}}(O_1, O_2), f_{assess_{GO}}(O_1, O_2)\} \\ f_{mix_{TF}}(O_1, O_2), & \text{if } f_{assess_{TF}}(O_1, O_2) = max\{f_{assess_{OPHID}}(O_1, O_2), \\ & f_{assess_{PSORT}}(O_1, O_2), f_{assess_{TF}}(O_1, O_2), f_{assess_{GO}}(O_1, O_2)\} \\ f_{mix_{GO}}(O_1, O_2), & \text{if } f_{assess_{GO}}(O_1, O_2) = max\{f_{assess_{OPHID}}(O_1, O_2), \\ & f_{assess_{PSORT}}(O_1, O_2), f_{assess_{TF}}(O_1, O_2), f_{assess_{GO}}(O_1, O_2)\} \end{cases}$$

4.4.3 Intersect from Lowest

$$f_{mix}(O_1, O_2) = \begin{cases} f_{mix_{OPHID}}(O_1, O_2), & \text{if } f_{assess_{OPHID}}(O_1, O_2) = min\{f_{assess_{OPHID}}(O_1, O_2), \\ & f_{assess_{PSORT}}(O_1, O_2), f_{assess_{TF}}(O_1, O_2), f_{assess_{GO}}(O_1, O_2)\} \\ f_{mix_{PSORT}}(O_1, O_2), & \text{if } f_{assess_{PSORT}}(O_1, O_2) = min\{f_{assess_{OPHID}}(O_1, O_2), \\ & f_{assess_{PSORT}}(O_1, O_2), f_{assess_{TF}}(O_1, O_2), f_{assess_{GO}}(O_1, O_2)\} \\ f_{mix_{TF}}(O_1, O_2), & \text{if } f_{assess_{TF}}(O_1, O_2) = min\{f_{assess_{OPHID}}(O_1, O_2), \\ & f_{assess_{PSORT}}(O_1, O_2), f_{assess_{TF}}(O_1, O_2), f_{assess_{GO}}(O_1, O_2)\} \\ f_{mix_{GO}}(O_1, O_2), & \text{if } f_{assess_{GO}}(O_1, O_2) = min\{f_{assess_{OPHID}}(O_1, O_2), \\ & f_{assess_{PSORT}}(O_1, O_2), f_{assess_{TF}}(O_1, O_2), f_{assess_{GO}}(O_1, O_2)\} \end{cases}$$

4.4.4 Intersect from All - Add to OPHID Everytime

$$f_{mix}(O_1, O_2) = f_{mix_{OPHID}}(O_1, O_2) \circ f_{mix_{AddOPHID}}(O_1, O_2) \circ f_{mix_{PSORT}}(O_1, O_2) \circ$$
$$\circ f_{mix_{TF}}(O_1, O_2) \circ f_{mix_{GO}}(O_1, O_2))$$

where $\circ$ is the not associative and not commutative composition and O_1 and O_2 are the objects to change.

4.4.5 Intersect from All - Add to OPHID Restricted

Apply the $f_{mix_{AddOPHID}}(O_1, O_2)$ function only if $f_{assess_{RNA}} > 0.9 \wedge f_{assess_{TF}} > 0.99 \wedge f_{assess_{PSORT}} > 0.99$

4.4.6 Intersect from All - Add to OPHID Hard

Apply the $f_{mix_{AddOPHID}}(O_1, O_2)$ function only if $f_{assess_{RNA}} > 0.9 \wedge f_{assess_{TF}} > 0.99 \wedge f_{assess_{PSORT}} > 0.9$

4.4.7 Intersect from All Except OPHID - Add To OPHID Restricted

$$f_{mix}(O_1, O_2) = f_{mix_{AddOPHID}}(O_1, O_2) \circ f_{mix_{PSORT}}(O_1, O_2) \circ f_{mix_{TF}}(O_1, O_2) \circ f_{mix_{GO}}(O_1, O_2)$$

where $\circ$ is the associative and commutative composition and O_1 and O_2 are the objects to change. Apply the $f_{mix_{AddOPHID}}(O_1, O_2)$ function only if $(f_{assess_{RNA}} > 0.9 \wedge f_{assess_{TF}} > 0.99) \vee (f_{assess_{PSORT}} > 0.9) \wedge f_{assess_{RNA}} > 0.9 \wedge f_{assess_{TF}} > 0.9)$

4.4.8 Non Intersecting when Empty

Another switch was introduced to influence the mixing behavior. The fact if empty sets are also intersected had a strong influence on the systems dynamics. Empty sets are entries which had

information during import, but during the course of updates they have become empty. Data entries not inizialized during import are not changed at all. The interpretation of not initialized entries plays a crucial role in the in section 4.2 'Assessment Function'.

- Situation A - Empty Set Intersection
  if one of the objects has an empty set for OPHID, TF or GO the intersection is built and the objects are changed.
- Situation B - No Empty Set Intersection
  if one of the objects has an empty set for OPHID, TF or GO the intersection is **not** built and the objects are **not** changed.

5 Considerations Regarding the Implementation

5.1 General

This section specifies requirements of urgent need with respect to architecture and implementation. The major goal in the implementation during the proof of concept is openness, easy to understand programming style and a software, which keeps changes local. This results in a tradeoff with performance, see section 6 for further information on why this is acceptable.

5.2 Import

As already mentioned in the introduction, flexibility and especially openness with respect to import data is a major requirement in this context. Data in the area of computational biology and Systems Biology is extremely versatile and originates from various different sources.

The system therefore has to exhibit a well defined input interface (on GUI level as well as on data level) to assure the compatibility to other data sources. This is of importance if there is a need to include various data origins for the calculation. We propose a special input format (EmergeElect input format) allowing the definition of classes of imports (like arrays of 'floats', 'doubles', 'strings', or other types of variables).
Each line in a file contains a GeneID a tabulator and a list of tabulator separated entries of the type 'double', 'String', 'double-array' or 'String-array' as specified during the import routine. A header can optionally be specified. An example on this format for protein protein interactions based on OPHID (OPHID, 2006) follows:

> GeneId GoTerms
> 1387 GO:0007582 GO:0043226 GO:0005623 GO:0009987 GO:0003824 GO:0050896 GO:0005488

6531 GO:0005215 GO:0005623 GO:0005488 GO:0007582
7022 GO:0007582 GO:0043226 GO:0005623 GO:0009987 GO:0030528 GO:0005488
1638 GO:0005623 GO:0007582 GO:0043226 GO:0008372 GO:0005488 GO:0003824
GO:0007275

5.3 Evaluation and Visualization

5.3.1 Visualization

After each update propagation the system is evaluated and can be seen as a graph. It is not necessary to implement a full framework for the visualization of the graph network. It is sufficient to implement an interface to a common graph visualization tool. Examples of these programs are ProteoLens (Proteolens, 2006) or Cytoscape (Cytoscape, 2006).

5.3.2 Evaluation

Another special part is the evaluation of the graph network. Therefore the graph analysis software of (Platzer *et al.*, 2006) is used. This software is able to take any graph and calculate parameters describing the graph and giving evidence on its characteristics. An interface to this graph analysis software had to be established.

5.4 Update Propagation and Neighbourhood Functions

Due to the fact that this part is subject to change very frequently, it is absolutely necessary to find a flexible architecture for this part where changes stay **local**. This is an urgent need to implementation and must not be ignored.

5.5 Implementation Details

The choice programming of the language for the proof of concept is not of relevance, because of an restricted input size. However, when calculating the full human proteome, this fact will have remarkable influence on the computability. More on this in the performance section 6.

6 Performance Considerations

6.1 General

For the proof of concept performance considerations are not of high importance. The proof of concept can be done on a small subset of the human proteome. Nevertheless, if the proof of concept comes out to be a valid approach, we plan to apply the software to real size problems. In that case

performance is a highest-priority requirement and this section gives a small outline, on what the resource comsuming parts are and how they can be handled to keep a good solution in sight.

6.2 Data Import

The data import can be a critical part regarding to memory issues. Keeping all objects in memory at the same time will not be possible.

6.3 Efficiency and Memory Usage in a Real World Scenario

Input candidates: 20 000
Data sources: 6

For the assessment function a general statement on performance issues can be provided, because the calculation follows a scheme which will not change. The validation is done as a 'n-to-n' relation of objects. It is a symmetric matrix which has to be calculated and which complexity function is in $O(n^2)$ for assessment. Assessment is done once for each update propagation. This is a time consuming part, only compensatable through high performance processors.
The trickier part is the memory usage of the matrix. 20 000 objects result in 20 000*20 000 = 400 000 000. A not directed graph results in a symmetric matrix representation. We can take it half and get 400 000 000 / 2 = 200 000 000 entries for the matrix. Saving one float value for each entry would already result in 200 000 000 * 4Byte = 800MByte. There will be at least five to six entries in a realistic scenario, which means an overall matrix size of 800MByte * 6 = 4.6GByte. This means it will be necessary to keep the program very slim only primitive datatypes implemented preferably in a high performance programming language because swapping behaviour will likely not be avoidable.

For the neighbourhood function and mixing a general statement on performance issues can not be given, because efficiency varies with the methodology chosen.

6.4 Update Propagations and Neighbourhood Function

This issue is a similar one to the update functions in cellular automaton. As soon as the validation is done the update can be established in a parallel way.

Part IV
Application and Results

7 Empirical Data Selection

7.1 Overview

This section holds the details on the decisions taken regarding to the choice of the biochemical data set. The following steps were planned and executed:

1. Select three biological pathways (PI, PII, PIII) from the KEGG database (Control Group)(KEGG, 2006).
2. Extract genes from PI, PII, PIII and analyse the overlap between the genes.
3. Extract gene expression data corresponding to the genes from PI, PII, PIII from the Welsh ovarian cancer data set (J.B. Welsh, 2001).
4. Statistical analysis of the frequency distribution of the Pearson coefficients of correlation of the gene expression data (in and between the pathways PI, PII, PIII).
5. Complete the chosen genes from PI, PII, PIII with randomly selected genes (PIV) excluding the genes from PI, PII, PIII.
6. Extract gene expression data from Welsh (J.B. Welsh, 2001) for randomly picked genes PIV.
7. Statistical analysis of the frequency distribution of the Pearson coefficients of correlation for the randomly selected genes from PIV.
8. Extract data from OPHID (OPHID, 2006) for PI, PII, PIII, PIV.
9. Extract data from KEGG (KEGG, 2006) for PI, PII, PIII, PIV.
10. Consolidate transcription factor data for PI, PII, PIII, PIV.
11. Calculate the intercellular location (on protein level) of the proteins encoded by the genes from PI, PII, PIII, PIV using PSORT (Nakai & Horton, 1999).
12. Extract data for PI, PII, PIII, PIV from the GeneOntology (GeneOntology, 2006).

Just to be mentioned here is a problem of compatibility, grounding on a m-to-n relation between coding regions on a gene (RNA-level) and the proteins (protein-level). This results in problems like the resolving of the Gene IDs when obtaining the PSORT(protein-level) data. See section 7.6 for further details.
In the following sections the choice of the data sets and the challenges in integrating them are explained in detail.

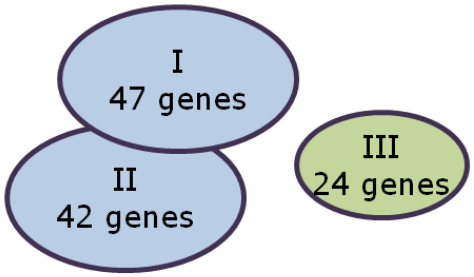

Figure 16: The chosen pathways in a schematic Venn diagram: the apoptosis pathway (divided into two parts) and an excerpt from the MAPK pathway (KEGG, 2006).

7.2 Selection and Integration of Omics Data - Evaluation Approach

7.2.1 Experimental Settings

To evaluate the accuracy of the model when propagating our system, the result is compared to a data set whose context is already known in advance. At first

- genes from three pathways from the KEGG (KEGG, 2006) are chosen where two of the pathways interact and one of them stands isolated
- then this set is completed with randomly chosen genes.

This scenario - the composition of the pathways in form of Venn diagrams - is demonstrated in figure 16 . After every update propagation it is possible to calculate a distance between all pairs of objects based on the data stored in their (dynamic) data structures. This distance measure can be seen as an edge between each pair of genes. The full system represents a **complete graph** (stored in a matrix) like in figure 17. Here we have just ten nodes, but for our example we have for 316 vertices and 49770 edges for an undirected complete graph, because of $n_{edges} = (n_{vert}^2 - n_{vert})/2$. When a certain threshold for including or excluding an edge is defined, the resulting graph might be not complete anymore. Edges with a strength lower than this so-called *'cutoff'* are excluded. The *'cutoff'* defines which edges are seen as bias, and which ones show an accepted interaction between two objects (figure 17). This graphs are then analyzed by the graph analysis software (Platzer *et al.*, 2006).

> **Theory 1** *After the update propagation of the model is complete, an analysis of the resulting subgraphs should identify three stronger connected networks, which represent the genes from the originally chosen pathways.*

Of course it is not easy to proove, if the quality of the pathway reconstruction is improving. Several questions arise

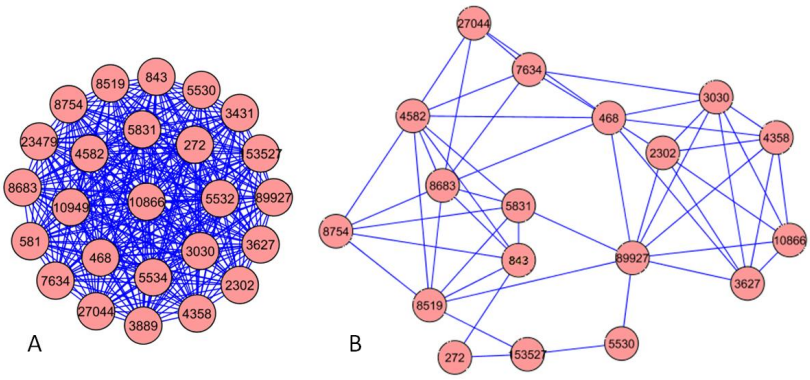

Figure 17: A shows a complete graph and B a graph after a certain cutoff is adapted.

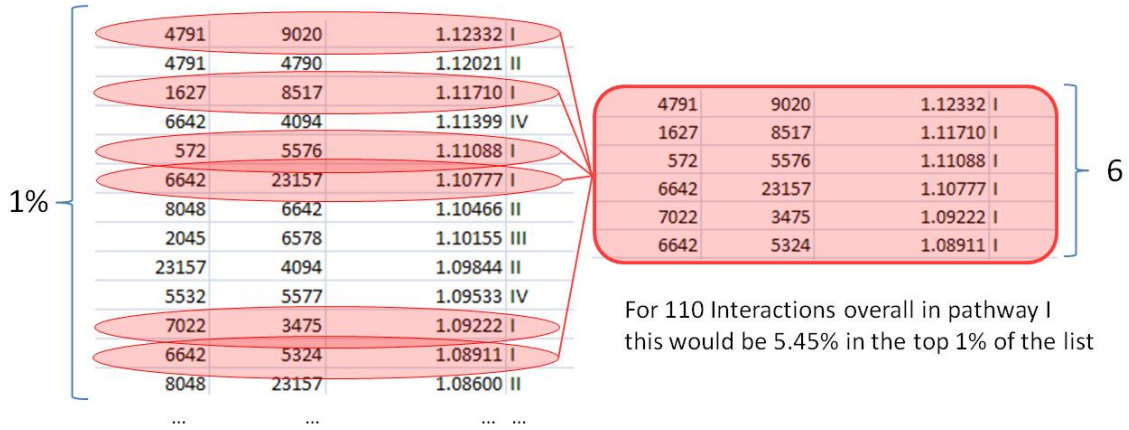

Figure 18: The sorted interaction list and how the pathway percentage is calculated. This calculation is done for each percentage of the *interaction list* and then the results are plotted on the y-axis over an x-axis from 1-100%.

1. how can the improved quality in identifying the pathway networks be measured and documented?
2. how can the evolution of the propagations be monitored and interpreted?
3. how can an exit condition for the optimal system state be found, after the n^{th} iteration of the system?

Each update propagation is followed by an evaluation and in a first approach the problems mentioned above are addressed in the following way:

- **ad 1**
  A list of ALL interactions of the matrix is created. Then this list is sorted by decreasing strength of the edges (figure 18). We call this list the *interaction list* and due to the symmetry of matrixes generated from undirected graphs, the number of entries can be reduced from $n =$

i_a to $n = i_a^2 - i_a/2$ entries, where i_a denotes the number of all (non-redundant) interactions. In beforehand all interactions of genes within each pathway were extracted from KEGG, which resulted in a list of gene interactions, each list only holding the genes (and interactions) from its corresponding pathway in the KEGG. This leads to four so called *reference lists* PI, PII, PIII, PIV (PIV if the randomly chosen interactions are also taken into account and treated as one extra "pathway").
The *interaction list* is then divided into 100 intervals and processed from the top interval. A percentage ratio is calculated after how many intervals all occurrences of each *reference list* has been processed (figure 18). In further context this value is referred to as the *completeness ratio*

$$c_{r_x} = i_{p_x}/i_a$$

where c_{r_x} is the *completeness ratio* of the considered pathway and $x = \{I, II, III, IV\}$, i_{p_x} denotes the number of interactions processed and i_a denotes again the number of all interactions. This percentage value is formatted into a 2D-plot, where i_{p_x} is plotted on the axis of abscissae and the completeness ratio c_{r_x} on the ordinates. For a working model, the curve results in a concave form, otherwise the curve characteristics are convex.

- **ad 2**
  The systems's evolution is monitored by graph measures which are generated by the graph characteristics calculated by a software from Platzer et al. (Platzer *et al.*, 2006) at selected *cufoffs* after a chosen number of update propagations. To get an understanding of how the graph measurements (Platzer *et al.*, 2006) scale on our graph and which ones can be transferred into an useful context for our problem, the *interaction list* from above is divided into ten intervals and these intervals are processed in a top down manner, subsequently adding the edges of the next interval to the graph and computing its characteristics. After each step the characteristics are represented in a 2D-plot with the graph parameter on the y-axis and the number of intervals processed on the x-axis as given in section 7.2.2. The interactions are also be logged for examination later.

- **ad 3**
  In a first attempt the behaviour of the graph is monitored for 1000 iterations. The basic idea is to find a graph measure (e.g. the entropy of the graph) to formulate an exit condition. Also the *graph energy* - a term introduced to show the current energy state of the graph like in a thermodynamical system - is monitored. An optimal *graph energy* considers the perfect state of the graph as the most likely configuration after the theory of (Metropolis *et al.*, 1953), a Monte Carlo approach with a Metropolis criterion.

Another issue is the choice of the sequence of the update propagations, which comes along with the system dynamics and is therefore described in section 8.

The remainder of this section gives an exact explanation of the difficulties, particularities and pitfalls during the choice and assessment of the data set, as well as it lists the sources of the data sets. How the data was derived, processed and interpreted is also covered.

7.2.2 Analysis Utilizing Graph Characteristics

Platzer et al. (Platzer *et al.*, 2006) established a software characterizing protein interaction networks in tumors. Not all of the originally 22 graph characteristics have been included, because some of them were calculated relatively to OPHID (OPHID, 2006) which is not of interest to us. The remaining graph characteristics are shown in figure 19. The software was changed to return the number of subgraphs, vertices and edges and the 19 remaining graph measures for the biggest subgraph. Every time a graph analyis on the system is conducted it is done with ten *different minimum edge weights* also called *cutoffs*. All edges not exceeding this minimum are marked as not existent for the graph analysis. So for every characteristic, ten results (for all the different *cutoffs*) are achieved for one graph analysis. These characteristics are then drawn into a diagram with the percentage for each *cutoff* for an edge weight between 0 and 1 on the x-axis, and the value of the graph characteristic on the y-axis as given in figure 21.

7.2.3 Selection Process of the Control Group

The first necessity is the selection of the control group. The focus bases upon pathways from the KEGG database (KEGG, 2006).
Following our approach we selected three well understood pathways of actual scientific interest in medical studies, of which two are interacting and one is an isolated pathway, which does not have relations to the first two chosen ones as seen in figure 22. The two interacting pathways are:

- the apoptosis pathway
- the TRAIL - caspase cascade pathway

and the isolated one is the

- the MAP kinase signaling pathway.

All of these steps together include 113 participating genes for which a maximum of corresponding -omics data has to be collected. Some of the data is generated in-house (in that case only a proper selection and/or conversion to fit the import interface has to be done), while other data was more difficult to retrieve and/or had to be calculated.

These 113 overall genes from PI, PII and PIII now have to be examined with respect to overlaps. This ratios are given in the following table:

Name	Description	
	size measures	
Closeness Centrality	$CC_i = \frac{1}{\sum_j d(i,j)}$	*d(i,j)* is the length of the shortest path between vertex *i* and *j*. The sum of CC_i over all vertices gives the total closeness centrality of a given sub-graph.
Graph Diameter	$GD = \frac{\max(d(i,j))}{N}$	*d(i,j)* is the length of the shortest path between vertex *i* and *j*. *GD* is computed for all pairs *(i,j)*, the graph diameter is reflected by the longest path identified.
Index of Aggregation	$IoA = \frac{A}{B}$	*A* is the total number of vertices in the sub-graph, and *B* is the total number of all given vertices of the graph.
	distribution measures	
Assortative Mixing Coefficient	$r = \frac{4^* < k_1 * k_2 > - < k_1 + k_2 >}{2^* < k_1^2 + k_2^2 > - < k_1 + k_2 >^2}$	k_1 and k_2 are the counts of edges of two vertices connected by a given edge. This measure reflects the edge-edge distribution over all edges of a graph.
Entropy of the distribution of edges	$H = -\sum_k p(k) \ln p(k)$	*k* is the count of edges of one vertex, and *p(k)* is the ratio of vertices which have *k* edges.
	relevance measures	
Betweenness	$B = \frac{\sum_{i \in V} \sum_{j,k} \frac{\sigma(j,i,k)}{\sigma(j,k)}}{N}$	σ *(j,i,k)* is the total number of shortest connections between vertex *j* and *k*, where each shortest connection has to pass vertex *i*, and σ *(j,k)* is the total number of shortest connections between *j* and *k*. We computed σ *(j,i,k)* and σ *(j,k)* for the entire OPHID graph, but then only used vertices also present in the sub-graph generated on the basis of a given gene expression data set.
Betweenness of all selected vertices	As for Betweenness, but all selected vertices are considered.	
Stress Centrality	$StC = \sum_{i \in V} \sum_{j,k} \sigma(j,i,k)$	σ *(j,i,k)* is the total number of shortest connections between vertex *j* and *k*, where each shortest connection has to pass vertex *i*.
	density measures	
Connectivity	$C = \frac{A}{B}$	*A* is the total number of edges realized in a given graph, and *B* is the maximum number of edges possible.
Clustering Coefficient	$CLUST_i = \frac{A}{B}$	*A* is the total number of edges between the direct neighbors of vertex *i*, and *B* is the maximum number of possible edges between the direct neighbors of vertex *i*. The sum of $CLUST_i$ over all vertices gives the total Clustering Coefficient of a given sub-graph.
Number of edges divided by the number of vertices	$NeNv = \frac{A}{B}$	*A* is the total number of edges in a given graph, and *B* is the number of selected vertices in a given graph.
Community	$Comm = \frac{A}{B}$	*A* is the total number of edges, where both connected vertices are in the given sub-graph, and *B* is the total number of edges, where one connected vertex is in the sub-graph and the other vertex is outside.

Figure 19: Part one of the adapted graph characteristics from Platzer et al. (Platzer *et al.*, 2006).

Entropy	$H(G) = \sum_{v \in V, i(v)>=2} (i(v)-1) * \log(\frac{\mid E \mid - \mid V \mid +1}{i(v)-1})$ |*E*| gives the total number of edges, |*V*| gives the total number of vertices; $i(v)$ gives the number of edges of vertex *v*.	
Graph Centrality	$GC_i = \frac{1}{\max(d(i,j))}$	max(*d(i,j)*) is the length of the shortest path between vertex *i* and *j* for a given vertex *i*.
Sum of the Wiener Number	$W_i = \frac{1}{2} * \sum_{i,j} d(i,j)$	*d(i,j)* is the length of the shortest path between vertex *i* and *j*. We computed a sum of the Wiener Number for each vertex.
cycles		
Eigenvalues	$EV = \sum_j \mid ER_j \mid ^2$	ER_j is the real part of the j-th Eigenvalue for the adjacency matrix of the given sub-graph.
Subgraph Centrality	$SC = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{\infty} \frac{(A^k)ii}{k!}$	*A* is the adjacency matrix. We computed *SC* for *k* [1,99].
Cyclic Coefficient	$\theta(i) = \frac{2}{k_i * (k_i - 1)} * \sum_{j,k} \frac{1}{S_i(j,k)}$ $\theta = 1/N * \theta(i)$	S_i is the smallest possible cycle of the vertex *i* and two of its neighbor vertices *k*. The total Cyclic Coefficient is then given for all vertices *N* as θ.

Figure 20: Part two of the adapted graph characteristics from Platzer et al. (Platzer *et al.*, 2006).

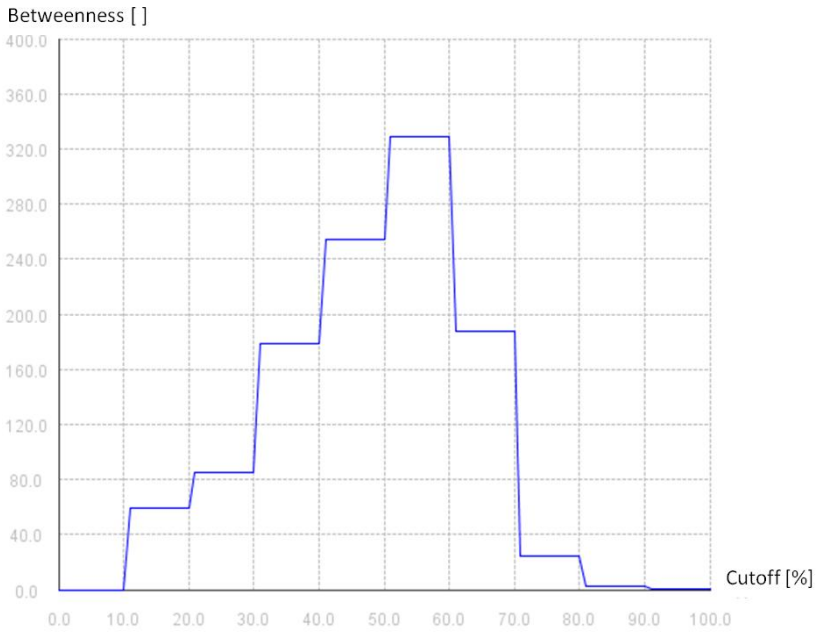

Figure 21: The betweenness graph characteristic in 10%-steps (*cutoffs* between an edge weight of 0 and 1).

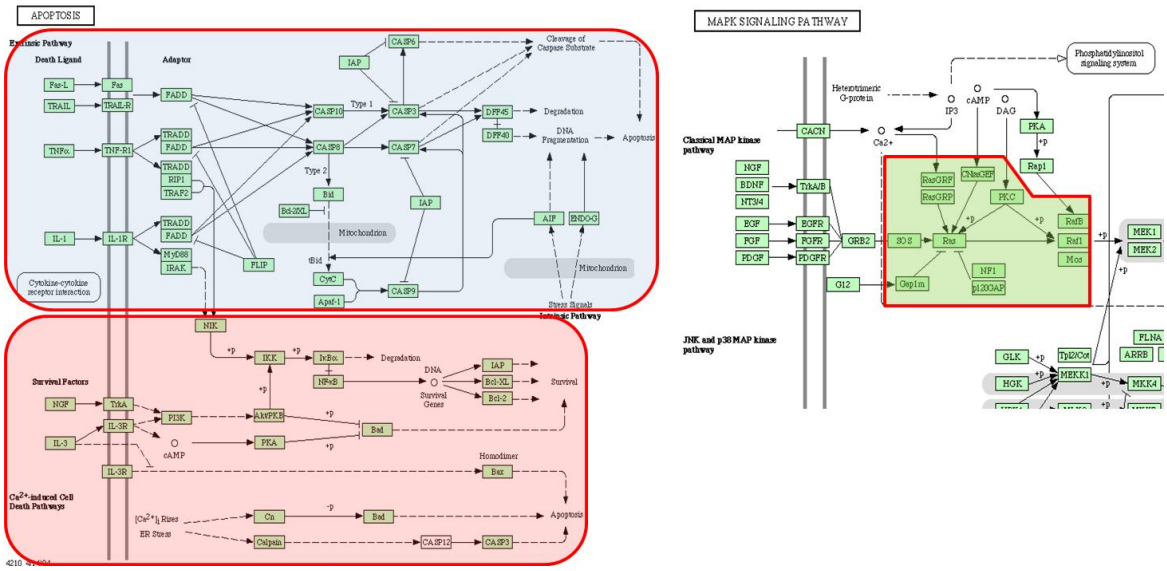

Figure 22: The chosen pathways: One is a split up apoptosis pathway and the other one is an excerpt from the MAPK pathway (KEGG, 2006).

Pathway	No of Genes	Unique Genes
PI	42	41
PII	47	47
PIII	24	24
	$\sum$=113	$\sum$=112

The pathways - we call the sets *PI, PII and PIII* - have their corresponding cardinalities 42, 47 and 24. In set PI a gene occurs twice (cardinality decreases from 42 to 41) which would result in 112 genes. Due to the fact that the union is build of all three sets $PI \cap PII \cap PIII$, again genes are overlapping which results in 106 unique genes after creating the union of the sets PI, PII and PIII. This chosen set of genes represents approximately 30% of the overall selection. The other 70% are chosen randomly (pathway PIV). The only important constraint is that the random set has to be filled with gene expression data at the same rate as the genes from the sets PI, PII and PIII. This is, because of the gene expression's data samples (microarrays) from clinical experience represent all human genes. This means that only a percentage of the chosen genes from PI, PII, PIII matches to genes on the microarray, and as a consequence the objects (representing the genes) are only partially filled with gene expression data. Based on our 106 genes, 72 genes are covered by data, whereas 34 are not, which represents approximately about 68% coverage. This ratio has to be the same in both, the selected and the randomly chosen data.

7.2.4 Randomized Data

The randomized data was chosen by reducing the set of all genes in the microarray by the chosen ones from PI, PI, PIII and then extracting 210 randomly chosen genes. The coverage of the microarray data with respect to the selected random set PIV was adjusted to the coverage of the gene expression data for PI, PII and PIII which was about 68%. In absolute numbers 142 genes are filled with gene expression data and 68 are blank.
Now - after the selection of the genes - the procedure is continued by filling the objects (which represent genes) with data.

7.3 Pathway Data

7.3.1 KEGG

Actually the first step - instead of getting the pathway identifiers to the corresponding genes - was to identify the underlying genes from the pathways. This was already done in-house by downloading the pathway database files from the KEGG, and extracting those genes, where one of the chosen pathway identifiers occurs.
Because the **apoptosis pathway** and the **TRAIL - caspasis cascade pathway** are combined in one single pathway in the KEGG database (KEGG, 2006), their identifiers were split up into two new identifiers representing two different pathways. Instead of the original pathway (notated by the identifier hsa04210) two new pathways PI and PII were created. The **MAP kinase pathway** was named PIII instead of hsa04010, and the randomly chosen genes were identified by the tag PIV. This gives us the possibility to clearly identify our chosen pathways and distinguish them from one another. This choice of naming had to be applied to the extracted interaction data from KEGG and to the *reference lists* mentioned above.

7.4 Gene Expression Data

7.4.1 Source Data

With respect to gene expression the underlying data was retrieved from a set of raw gene expression data sets from (J.B. Welsh, 2001) based on Affymetrix array technology (Affymetrix, 2000). We started with an uncomfortable mapping from AffymetrixIDs to NCBI GeneIDs, because the Affymetrix-technology provides its own nomenclature schemes.
Another fact is the appearance of genes multiple times in multiple measurement rows on the microarray. In that cases we normalized this test rows to one test row by taking the mean value of the combined measurements.
After solving this issues the data was forwarded to a statistical analysis to examine its frequency distributions of its Pearson coefficient of correlation.

7.4.2 Statistical Analysis

The first part of this analysis tries to figure out how many of the chosen gene expression measurement sequences from PI, PII, PIII show correlation. Therefore boxblots, respectively t-tests for identifying significance, have been made for each group in each measurement row (this means two boxplots per row). We found out, that only a small part of the genes show a significant correlation (about 30%).

The second part of this analysis aims to find out, if the gene expression gives evidence about a stronger correlation of each pair of genes whose place of residence is in the same pathway, and if there is a weaker correlation between each pair of the genes located in different pathways. Therefore seven different scenarios have been studied:

The correlation within pathways (figure 23)

- PI and PI
- PII and PII
- PIII and PIII
- PIV and PIV

The correlation between pathways (figure 24)

- PI and PII,PIII
- PII and PI,PIII
- PIII and PI,PII

In this case the unions of the combined sets had to be recreated to eliminate multiple genes.

After that, the same analysis was applied to the random set. Then a frequency distribution was used to compare the correlation of coefficients. They all showed almost the same distributions.

We conclude that gene expression alone does not expose a closely related behaviour within pathways and an isolated behaviour between them. In fact, the results from the statistical analysis show a nearly equal frequency distribution. This means correlations of objects in the system afterwards can not solely arise from gene expression.

After this analysis the records from the pathways PI, PII, PIII and PIV were extraced from (J.B. Welsh, 2001) and converted to an EmergeElect importable format.

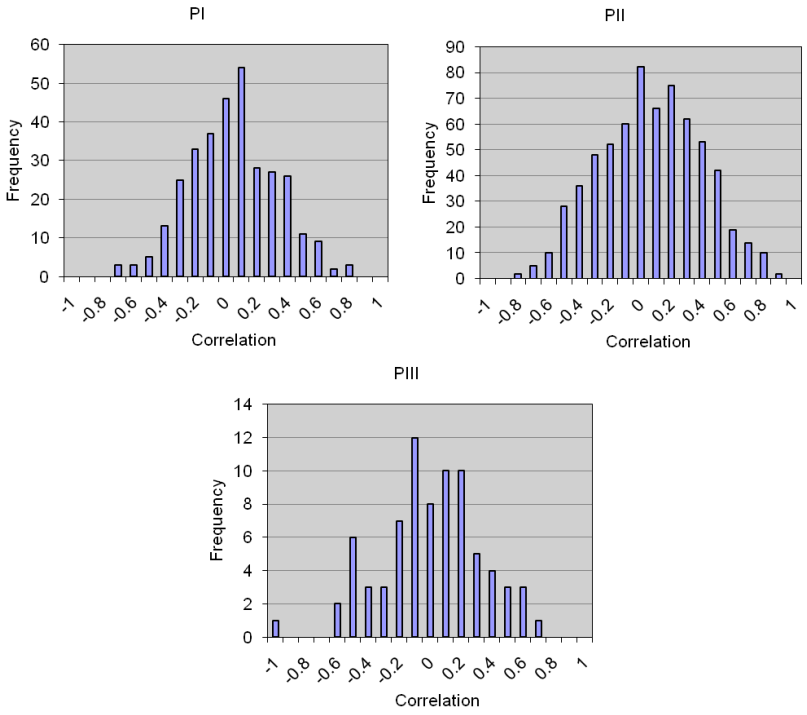

Figure 23: The frequency on the y-axis gives the counts for equal Pearson coefficients of correlation (x-axis) between two genes **within the pathways** PI, PII and PIII.

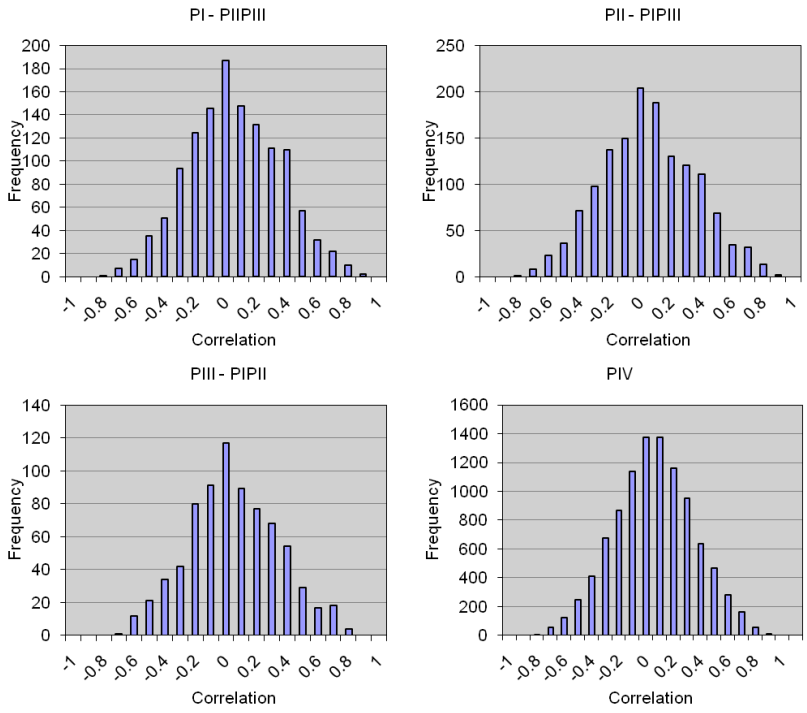

Figure 24: The frequency on the y-axis gives the counts for equal Pearson coefficients of correlation (x-axis) between two genes **between pathways** and **within the the random pathway** PIV.

7.5 Protein Protein Interaction Data

7.5.1 OPHID

Another very good data source is the OPHID interactions, which were included as well. First the OPHID interactions needed to be downloaded, interpreted (Perco *et al.*, 2006), parsed and converted to a proper input format. Then a mapping from proteins to Gene Symbols had to be done. Therefore we linked the longest transcript for a particular gene. Then the resulting file had to be mapped from Gene Symbols to GeneIDs and converted to the EmergeElect importable format.

7.6 Intracellular Location Data

7.6.1 PSORT

For all the genes from the chosen pathways the intracellular location of its corresponding protein is needed. Unfortunately a given gene is not exclusively linked to one particular protein, more likely many proteins derive from one gene, so that a way had to be identified to uniformly choose one of the protein sequences for every gene automatically. The NCBI (NCBI, 2006) is one of the institutions holding all these links from genes to proteins. As a restriction for the conversion from GeneID to refSeq (NCBI, 2006) it was stated, that in an ambiguous case the longest transcript was chosen. These protein sequences were saved in FASTA format and imported into an in-house database. A tool named PSORT was fed by this information, the output was converted back to GeneIDs and saved as an importable format for the EmergeElect framework.

7.7 Transcription Factor Data

7.7.1 Company Internal Data Sources

The transcription factor data was obtained from an in-house development process, based on the method of (Kielbasa *et al.*, 2005) and the data from JASPAR (Sandelin *et al.*, 2004) and TRANSFAC (Matys *et al.*, 2003). The resulting data sets (holding identifiers of transcription factors influencing expression data of particular genes) were transformed to our standardized EmergeElect input format. Each line in the input format contains a GeneID and the corresponding identifiers of the transcription factors.

7.8 Gene Ontology Data

7.8.1 www.geneontology.org

Gene ontology data was downloaded from www.geneontology.org (GeneOntology, 2006) and processed into a special format, where each gene is linked to the GO terms in which it is located. Then

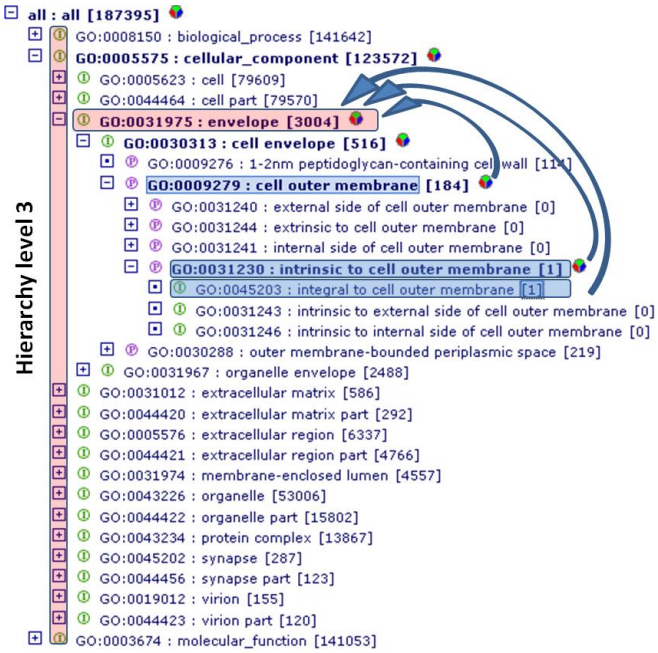

Figure 25: This is the GO hierarchy of (GeneOntology, 2006). All occurrences of genes in ontologies lower than three hierarchies have been mapped back in a pruning process to hierarchy three GO Terms as indicated by the arrows.

the GO terms were all mapped back to hierarchy three in the GO tree as represented by figure 25. This format is also compatible to an EmergeElect importable format.

8 Evaluation of the Model Dynamics

8.1 A First Test on the Correctness of Implementation with Biochemical Data

Before biochemical data was imported the implementation has been tested with a randomly generated data set. Now for the first time biochemical data have been imported. This was primarily a test of the import engine. On the one hand it was important to see if the biochemical data contains formatting anomalies and on the other hand this was done to figure out which kind of information has to be logged, and which programming libraries to take for creating diagrams. Then a simple dynamics rule was tested to verify the propagation. Figure 26 shows a snapshot after three iteration steps of the dynamics of the system. The value on the z-axis represents the sum of all incident edges to an object and the dots are the objects themselves. After each iteration the objects' values have been altered and thus also their relative position on the z-axis. The dynamics in that case strengthened the edges, so all objects increased in their value for the z-axis.

8.2 A Straight Assessment of the Import Data Set

This step is similar to obtaining data from STRING (von Mering *et al.*, 2005). Nevertheless STRING also owns an additional scoring scheme for generating additional information. This step applied all the assessment functions on different kinds of import data and verified this with a test on overlap with the KEGG pathways. The following scenarios were tested

- each dataset on its own
- all pair combinations of datasets
- all five datasets at a time

It is not necessary to show all assessments of the datasets. The data's behavior, when assessing the single datasets, were found to be very selective and varied strongly for the different pathways (like seen in figure 27 A,B). For example in figure 27 A only the GO terms are assessed and result in a very good overlap for pathway PI, while in B (assesses only the data from transcription factors) PI gets the minimal overlap. Obviously, the pair combinations counterbalanced the overlaps of the different data sets. Previously stronger pathways get weaker and the weak ones stronger when assessing all data sets at a time. Figure 27 C shows the assessment of GO and OPHID data together and they complete each other very well thus enhancing the overlap of PI. But when looking at D (OPHID and PSORT data) just the strong interactions from OPHID are rated good and afterwards we can observe a strong decrease of PI. At last the assessment took place with all of the datasets considered as given in figure 28, and the curves became even more balanced. As a future goal the system result has to enhance interactions resulting from synergic data and knock out inconsistent interactions so that this curve becomes maximally convex.

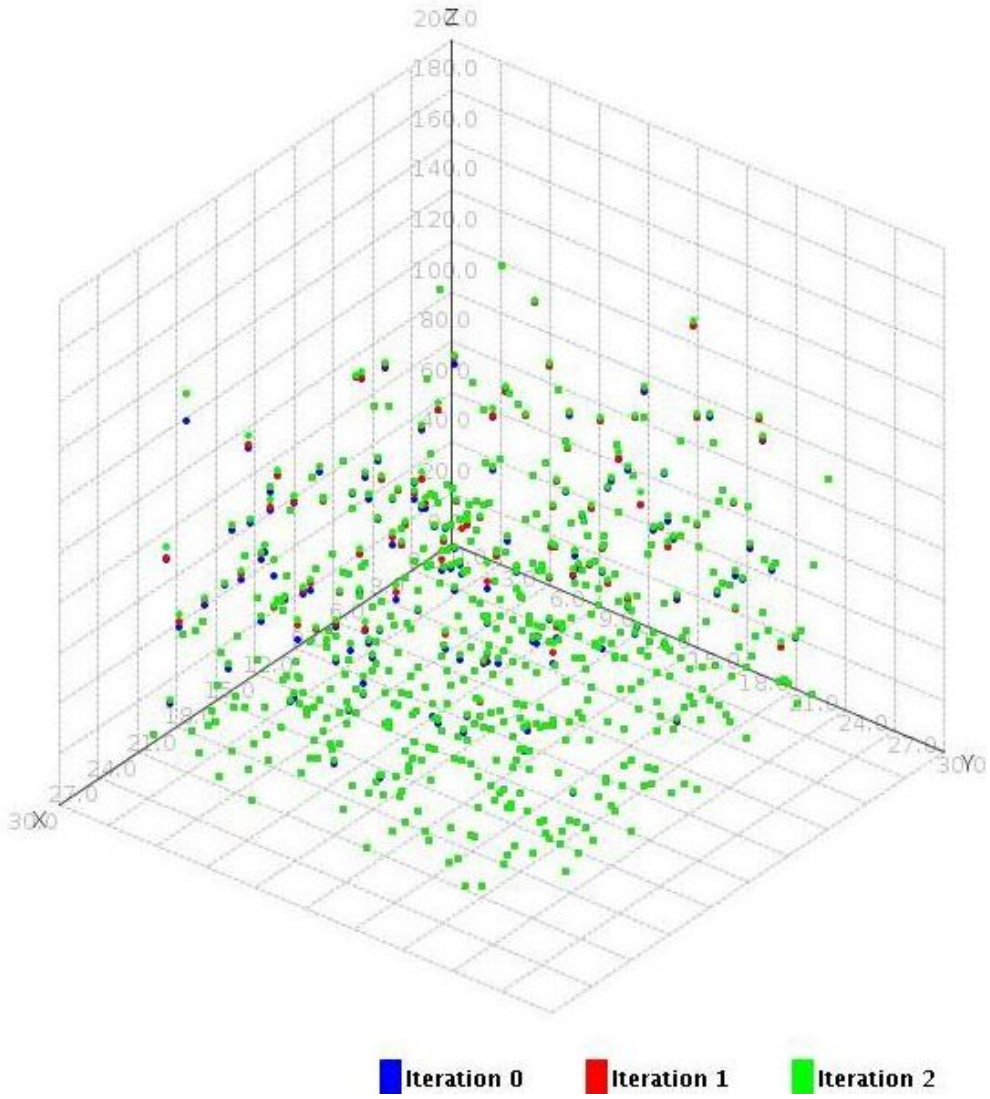

Figure 26: A snapshot of the results of in a first system test with biochemical data.

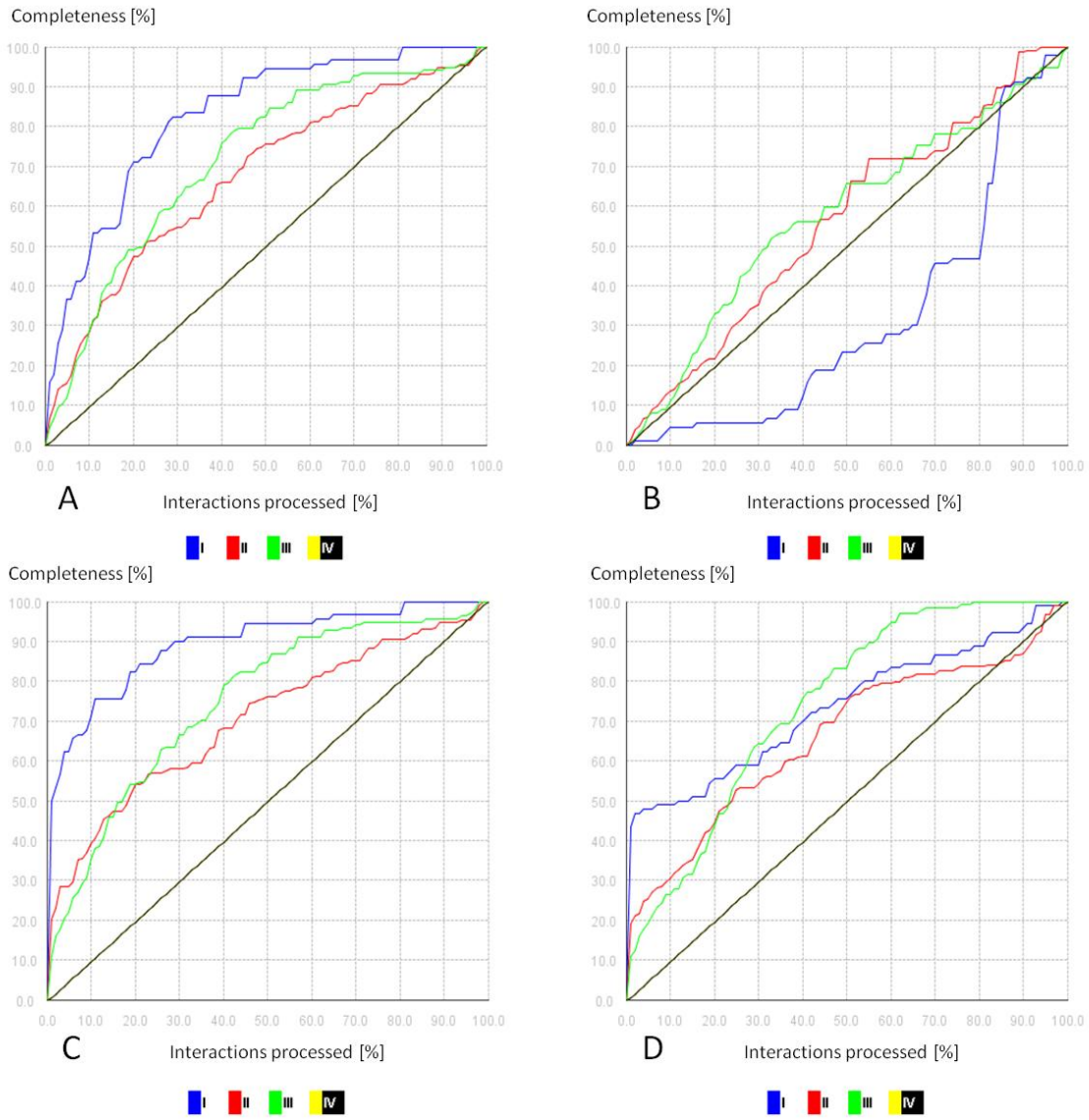

Figure 27: Independent assessment of the single data sources. A is the result of including only the GO terms, B of the transcription factors, C of GO and OPHID together, and D of OPHID and PSORT. The 'Completeness' on the y-axis denotes the overlap of processed interactions of the *interaction list* with the *reference list* holding the KEGG interactions and 'Interactions processed' refers to the percentage of interactions already processed from the *interaction list* as described in section 7.2.1.

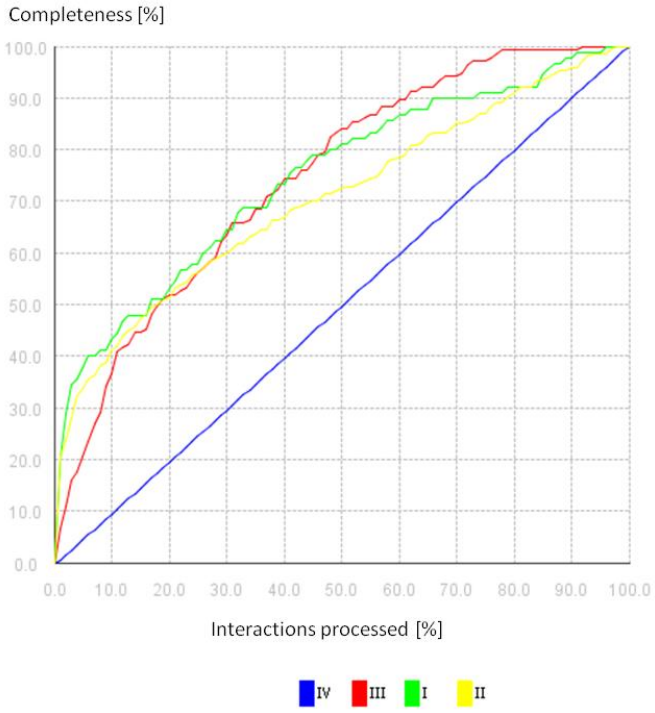

Figure 28: Assessment of all datasources together before propagation of the system. The 'Completeness' on the y-axis denotes the overlap of processed interactions of the *interaction list* with the *reference list* holding the KEGG interactions and 'Interactions processed' refers to the percentage of interactions already processed from the *interaction list* as described in section 7.2.1.

8.3 Results

The following section presents the results achieved in the evaluation. The first part gives a detailed description on the characteristics each testrun performed. Subsequently the results and their interpretation can be found. Testruns with no particular findings are only summarized and not described in detail. Only the major outcomes of testruns are outlined in detail.

8.3.1 Characteristics

For each testrun the environment, the settings, and several system characteristics were recorded to track the behavior of the system. Details on presented testruns are given in general

- **Verbal Description** A verbal description of this testrun.
- **Environment and settings**
  - *Assessment function* f_{asses}
    The assessment function is responsible for assessment of the edge weights between objects.
  - *Update-sequence function* $f_{neighbor}$
    Describes the part of the neighborhood function responsible for the update order of the objects.
  - *Update-mixing function* f_{mix}
    Describes which and how the data entries are updated in the objects.
  - *Number of iterations* n_{it}
    If $f_{next}(S_{n+1}) = f_{neighbor}(S_n) \circ f_{mix}(S_n)$ describes the update step to the next overall system state (S_{n+1} and S_n are the system states at iterations $n+1$ and $\circ$ is the composition) then $0 < n \leq n_{it}$.
  - *Analysis interval* $n_{analysis}$
    If $f_{analysis}(S) = f_{graph}(S) \circ f_{kegg}(S) \circ f_{out}(S)$ describes the analysis step, the analysis interval $n_{analysis}$ describes the interval in which a system analysis step takes place, where $f_{graph}(S)$ is the graph analysis, $f_{kegg}(S)$ the KEGG quality analysis and $f_{out}(S)$ an output function for parameters gained in each graph analysis step.
- **Iteration specific system characteristics in comparison**
  For each $f_{analysis}(S)$ extra data is generated.
  - *Graph characteristics for given offsets*
    The number of subgraphs remaining, if all edges under a certain offset are deleted. The offset starts at 0.0 and raises in steps of 0.1 to a maximum of 1.0 like presented in section 7.2.2. The interpretation is summarized for more analysis steps. Furthermore, the

number of interactions and vertices, and the number of vertices of the largest subgraph is documented.

- *General interpretation of subgraph behavior for all given offsets*
  An interpretation of the behavior over all subgraphs gained from each of the offset steps.
- *Graph measurement characteristics*
  The characteristics of the parameters calculated by f_{graph} in form of general explanations or in form of a diagram for prototypical situations.
- *Cytoscape graph interpretation*
  The interpretation of the subgraphs by visualization in cytoscape.
- *KEGG overlap*
  The interpretation of the overlap of the system with respect to the reconstruction of the KEGG pathways.

- **Overall system characteristics**
  These data and interpretations are only once per run subject to discussion.
  - *Graph energy*
    The graph energy is the sum of all interaction weights of each pathway, divided by the maximum weight of all possible interactions in this pathway (calculated for every iteration) assembled in a plot. This was done for all pathways PI, PII, PIII and for the full set of interactions respectively the full graph.
  - *Conclusion*
    A conclusion of the behavior of the system in this configuration, and the documentation of hypotheses on aspects explaining this behavior.

8.3.2 DM 01 01

- **Verbal Description**
  This effort was the first complete testrun of the system in the following environment.
- **Environment and settings**
  - f_{assess} : Assessment function 1 - Division by one
  - $f_{neighbor}$: Highest Edge Choice 1 - Reconsider RF
  - f_{mix} : Intersect from All - Empty Set Intersection
  - n_{it} : 1000
  - $n_{analysis}$: 100
- **Iteration 100**

– *Graph characteristics for given offsets*

Step	Cutoff	No. of interactions	Graph size	Largest subgraph	No. of subgraphs
1	0.0	49770	316	316	1
2	0.34804069528203774	39815	314	314	1
3	0.6960813905640755	35584	312	312	1
4	1.044122085846113	20209	287	287	1
5	1.392162781128151	13231	286	286	1
6	1.7402034764101888	5016	225	221	3
7	2.088244171692226	879	157	119	5
8	2.436284866974264	147	84	64	10
9	2.784325562256302	31	33	14	6
10	3.1323662575383397	10	14	5	4

– *General interpretation of subgraph behavior for all given offsets*
The graph slowly breaking up into subgraphs. The largest subgraph resembles the total graph up to high cutoff values. The dynamics - reflected as changing objects - stopped after a few iterations.

– *Graph Measures*
The betweenness characteristics confirms the subgraph behavior. But as long as the required dynamic behavior is not achieved, it does not make sense searching for characteristics in graph measures.

– *Cytoscape graph interpretation*
Overall the graph shows one big subgraph for nearly all cutoffs, clustering in the graph can not be observed. After 200 iterations the system is stable, as all updates are propagated (figure 29). Graphs after iteration 200 are not shown as the system remains stable.

– *KEGG overlap*
KEGG becomes initially worse but increases again (figure 30).

- **Overall system characteristics**

  – *Graph energy*
  The graph energy decreases and reaches stability fast. After iteration 100 the graph is nearly in its final state, and at iteration 200 the graph energy is stable and only a single interaction changes in the update interval 100 - 200. Because of the missing system

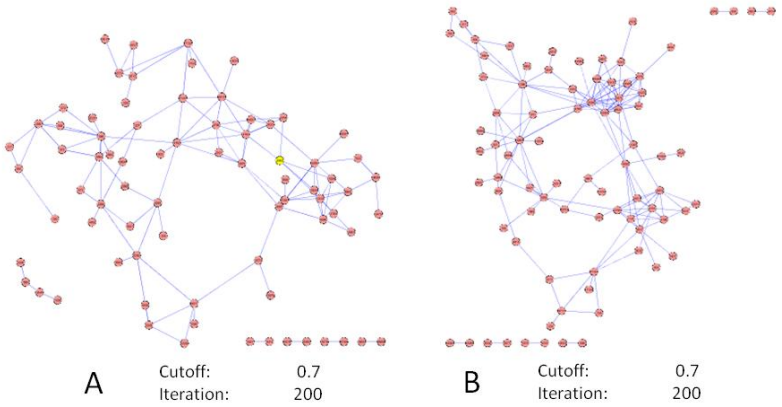

Figure 29: Graph at a cutoff of 0.7 before system propagation at iteration 0 can be seen in A. B shows the graph after 200 iterations.

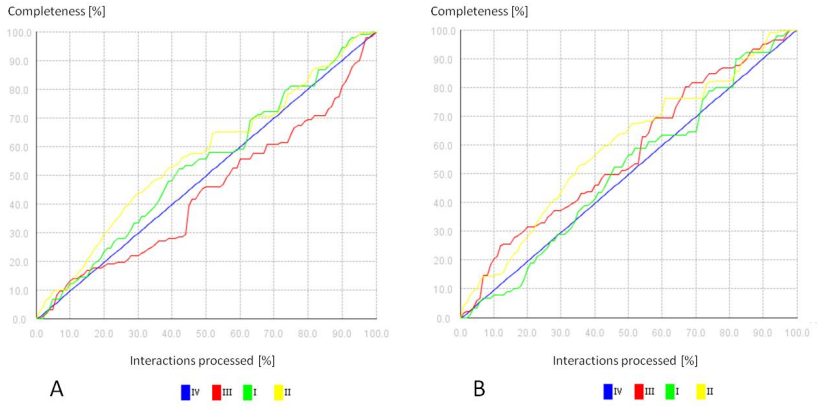

Figure 30: A shows the KEGG results after 100 iterations and B after 200 until 1000 iterations. Surprisingly we notice an increase again (after a decrease in A). The 'Completeness' on the y-axis denotes the overlap of processed interactions of the *interaction list* with the *reference list* holding the KEGG interactions and 'Interactions processed' refers to the percentage of interactions already processed from the *interaction list* as described in section 7.2.1.

dynamics, the focus in planning the next steps lies on splitting the graph into subgraphs by examining different neighborhood functions.

– *Conclusion*
This setup is highly dynamical during the first iterations, but then converges very fast (the influence of the mixing functions on the system might be too stringent). A major shortcoming is the subgraph behavior, and that there is no further clustering into subgraphs in the course of iterations. Because of that, we first focus on altering the dynamic behaviour (and neglect improving the KEGG overlap in the first place).

When making this run with the mixing function 'Intersect from Highest - Empty Set Intersection', a reasonable KEGG overlap was achieved, but the system's dynamics was missing. A new neighborhood function was introduced from DM 02 01 and DM 03 01 and the assessment function also changed to consider not initialized data entries in the objects. The next paragraph leads through some attempts with bad benchmarks, but with valuable insight into the systems dynamics. The experiences earned in this testruns build the base for the following section, where we could improve systems dynamics. The settings for DM 02 01 are:

- f_{assess} : Assessment function 4 - Division by Evaluated
- $f_{neighbor}$: Incident Rotation Edges - Get Highest then Random - Reconsider RF
- f_{mix} : Intersect from All - Empty Set Intersection
- n_{it} : 1000
- $n_{analysis}$: 100

and for DM 03 01:

- f_{assess} : Assessment function 4 - Division by Evaluated
- $f_{neighbor}$: Incident Rotation Edges - Check Random for All Unsorted - Reconsider RF
- f_{mix} : Intersect from All - Add to OPHID Everytime
- n_{it} : 1000
- $n_{analysis}$: 100

The behavior did not really improve with respect to a more fragmented or clustered graph. Also the KEGG overlap remained marginal. After this approaches a slight alteration was done again to the neighborhood function, resulting in DM 04 01, however, still no changes to the dynamics could be observed. Further studies were done in two more runs (DM 04 01 and DM 04 03) under different conditions, but the dynamics of the system just changed slightly. Then 'No Reconsideration' was

introduced and suddenly we noticed a change in the dynamics and a partitioning into subgraphs and clusters. The alteration was first done to DM 0X XX -series, and we could observe a stronger fragmentation and a higher KEGG overlap, but the clusters were not equally distributed. There was one large cluster and the others were very small.

8.3.3 DM 01 03 and DM 01 04

- **Verbal Description**
  This effort was the first testrun with 'No Reconsideration' approaches. Two runs with the same neighborhood function, but a slightly different assessment function are compared.
- **Environment and settings for DM 01 03**
  - f_{assess} : Assessment function 3 - Division by Average Filling
  - $f_{neighbor}$: Highest Edge Choice 2 - No Reconsideration
  - f_{mix} : Intersect from All
  - n_{it} : 1000
  - $n_{analysis}$: 100
- **Environment and settings for DM 01 04**
  - f_{assess} : Assessment function 3 - Division by All
  - $f_{neighbor}$: Highest Edge Choice 2 - No Reconsideration
  - f_{mix} : Intersect from All
  - n_{it} : 1000
  - $n_{analysis}$: 100
- **Iteration 1000 for DM 01 03 and DM 01 04**
  - *Graph characteristics for given offsets for DM 01 03*

Step	Cutoff	No. of interactions	Graph size	Largest subgraph	No. of subgraphs
1	0.0	49770	316	316	1
2	0.09999940628718453	39498	315	315	1
3	0.19999881257436905	28343	314	314	1
4	0.2999982188615536	18366	311	311	1
5	0.3999976251487381	10073	308	306	2
6	0.4999970314359226	4494	262	240	11
7	0.5999964377231072	1197	228	149	26
8	0.6999958440102917	183	191	47	64
9	0.7999952502974762	45	84	5	40
10	0.8999946565846607	6	12	2	6

– *Graph characteristics for given offsets for DM 01 04*

Step	Cutoff	No. of interactions	Graph size	Largest subgraph	No. of subgraphs
1	0.0	49770	316	316	1
2	0.08	40378	315	315	1
3	0.16	29164	314	314	1
4	0.24	19059	308	308	1
5	0.32	10681	307	305	2
6	0.4	5106	305	269	17
7	0.48	1550	237	156	27
8	0.56	250	194	65	49
9	0.64	58	101	8	47
10	0.72	13	26	2	13

– *General interpretation of subgraph behavior for all given offsets*
  The number of subgraphs did rise fast in both testruns. More subgraphs were found in DM 01 03, although the additional subgraphs found were small, and most of them contained just one edge. Overall the two results look similar. The lower amount of subgraphs in DM 01 04 can be explained by the different assessment function. The division by the average filling seems to introduce a higher variance to the interactions.
– *Graph Measures*

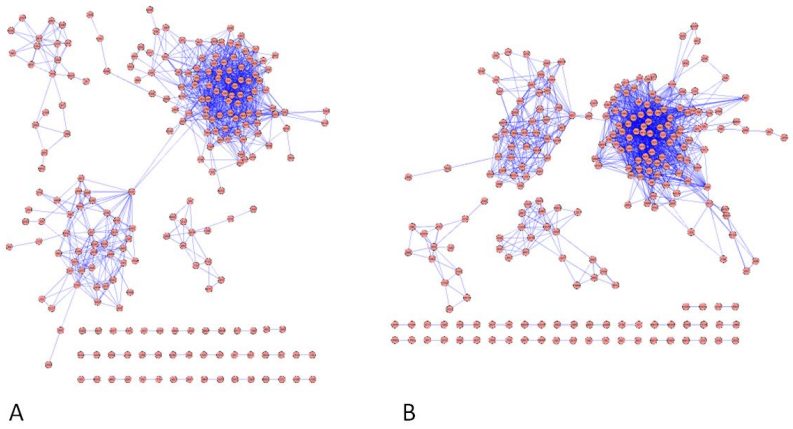

Figure 31: Cytoscape graph visualization for a cutoff of 0.7 for DM 01 03 in A and for DM 01 04 in B.

The system dynamics stopped after 100 iterations. There were no differences found in graph measures for subsequent steps.

- *Cytoscape graph interpretation*
  The graph shows several subgraphs. Most of them have only one interaction. After all the two testruns are similar regarding to their structure as shown in figure 31.
- *KEGG overlap*
  The KEGG verification is also similar, but improved already when compared to the first testruns, as given in figure 32.

- **Overall system characteristics**
  - *Graph energy*
    The graph energy is decreasing fast and becomes stable after 100 iterations. Obviously, system dynamics is still minor (figure 33).
  - *Conclusion*
    In this case we can observe a similar behavior of two different assessment functions. Nevertheless, at this point we started to favor the more distributed approach of the incident edge addition neighborhood functions.

Nevertheless, the change of the neighborhood function to our second basic approach led to more promising results. In the following result section the system converges to a state of equally distribution of clusters with high density and the KEGG overlap could be improved again.

8.3.4 DM 04 04 and DM 04 07

- **Verbal Description**
  This effort brought interesting clustered results, the 'No Reconsideration' approach was re-

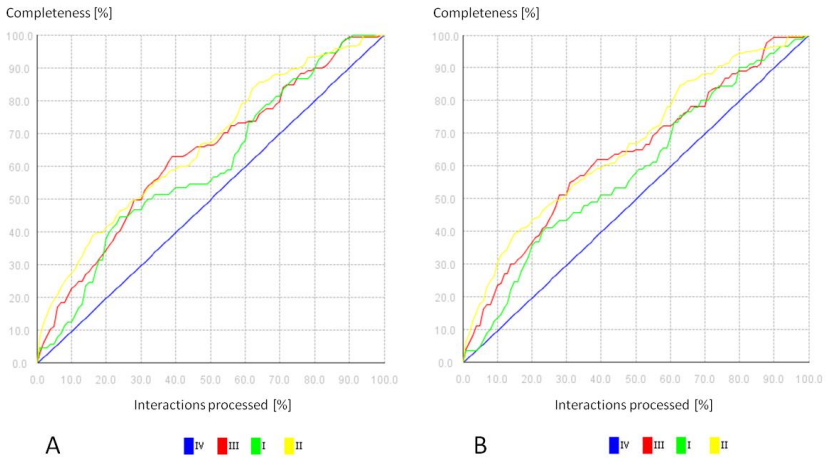

Figure 32: KEGG verification for DM 01 03 in A and DM 01 04 in B. The 'Completeness' on the y-axis denotes the overlap of processed interactions of the *interaction list* with the *reference list* holding the KEGG interactions and 'Interactions processed' refers to the percentage of interactions already processed from the *interaction list* as described in section 7.2.1.

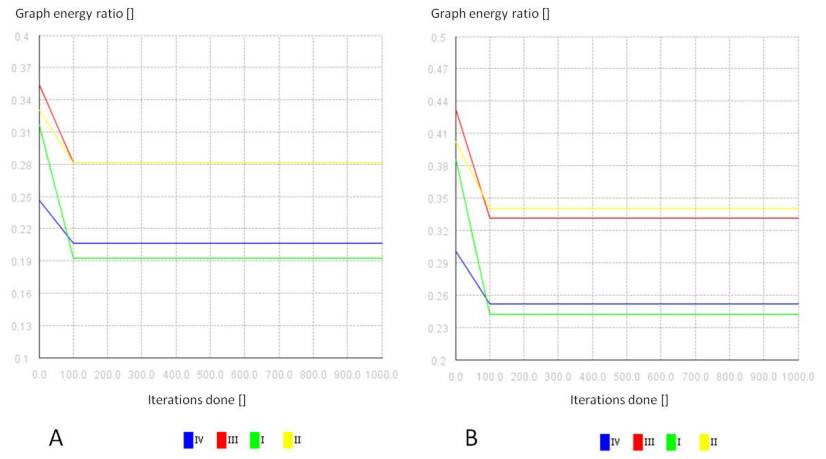

Figure 33: Graph energy for DM 01 03 in A and DM 01 04 in B plotted versus the iteration steps.

sponsible for fragmenting the graph into dense clusters. Two very similar testruns were set up and are now discussed. The settings for both of them are outlined in the next two paragraphs:

- **Environment and settings for DM 04 04**
  - f_{assess} : Assessment function 3 - Division by Average Filling
  - $f_{neighbor}$: Incident Rotation Edges - Check Random for All Sorted - No Reconsideration
  - f_{mix} : Intersect from All
  - n_{it} : 1000
  - $n_{analysis}$: 100
- **Environment and settings for DM 04 07**
  - f_{assess} : Assessment function 3 - Division by All
  - $f_{neighbor}$: Incident Rotation Edges - Check Random for All Sorted - No Reconsideration
  - f_{mix} : Intersect from All
  - n_{it} : 1000
  - $n_{analysis}$: 100
- **Iteration 1000 for DM 04 04 and DM 04 07**
  - *Graph characteristics for given offsets for DM 04 04*

Step	Cutoff	No. of interactions	Graph size	Largest subgraph	No. of subgraphs
1	0.0	49770	316	316	1
2	0.0975609756097561	29884	314	314	1
3	0.1951219512195122	22888	313	313	1
4	0.29268292682926833	11591	308	308	1
5	0.3902439024390244	6346	305	294	5
6	0.4878048780487805	3022	297	205	17
7	0.5853658536585367	1279	244	86	21
8	0.6829268292682927	1096	177	21	20
9	0.7804878048780488	271	108	16	13
10	0.8780487804878049	19	23	5	8

– *Graph characteristics for given offsets for DM 04 07*

Step	Cutoff	No. of interactions	Graph size	Largest subgraph	No. of subgraphs
1	0.0	49770	316	316	1
2	0.08124814717166705	32415	315	315	1
3	0.1624962943433341	24574	314	314	1
4	0.24374444151500113	14263	312	309	2
5	0.3249925886866682	7626	308	301	4
6	0.40624073585833526	2955	261	230	10
7	0.48748888303000226	1114	238	132	24
8	0.5687370302016693	817	181	64	25
9	0.6499851773733364	249	105	24	16
10	0.7312333245450034	16	14	5	4

– *General interpretation of subgraph behavior for all given offsets*
The fragmentation of the graph into subgraphs proceeds fast. Clusters are built during the iterations, but unlike in the next testrun DM 05 01, there is no amplification effect in the emergence of the clusters.

– *Graph Measures*
The *assortative mixing coefficient, stress centrality* and the *betweenness* exhibited a drastic change from iteration 100 to 1000. The edge-edge distribution (*assortative mixing coefficient*) is initially high for a cutoff of 90%. Later on this measure exhibits peaks for cutoffs of 10%, 70% and 90%. The behavior of the *stress centrality* for cutoffs higher than 60% was similar at 100 and 1000 iterations. But what dramatically decreased are the results for cutoffs below 60% for the 1000^{th} iteration. This finding indicates a stronger separation of the edge weights in the lower regions, or also a stronger division into subgraphs (this is confirmed by the betweenness measure). Slight differences could be observed for the *entropy of the distribution of edges, clustering coefficient, graph centrality, index of aggregation, cyclic coefficient, Wiener numbers* and *connectivity*. These measures did not show high differences for 100 and 1000 iteration neither for DM 04 04 nor for DM 04 07.

– *Cytoscape graph interpretation*
The graph showed a variety of subgraphs for both testruns (figure 34). For a cutoff of 80%, clusters can be identified. Cutoffs at 90% showed still an equal distribution of subgraphs. DM 04 04 seems to exhibit a stronger separation around a cutoff of 80%. DM 04 07 distorted genes from the apoptosis pathway dramatically and the clusters were

not clearly separated as in DM 04 04. The yellow nodes represent genes taken from the apoptosis pathway PI and PII.

- *KEGG overlap*
  The KEGG graph showed no significant changes. Until the $100th$ iteration all pathways become more biased, and afterwards the verification underlay only minor changes as seen in figure 35. Anyway, the pathway overlap improved compared to DM 01 03 and DM 01 04.

- **Overall system characteristics**
  - *Graph energy*
    The systems energy became more dynamic compared to the former runs (figure 36). Propagations in the first 100 iterations influenced the system massively. From iteration 100 to 1000 minor changes were observed.
  - *Conclusion*
    The systems dynamics did not reach a stable state as fast as in the former runs. The first iteration steps showed formidable impact on the system, but dynamics go on until iteration 1000 and beyond. The pathway overlap did rise compared to the former runs. In DM 04 04 the MAPK pathway had a lower overlap, but pathway PI can be reconstructed well. In DM 04 07 the reconstruction of MAPK was better, but the pathway PI performed worse.

As we can see, the major impact given on the systems dynamics results from the neighborhood function. For further examining this finding we expanded the function by implementing the Metropolis criterion. We did not get further change to the overall dynamics, but with an additional change of the mixing function the KEGG overlap could be further improved.

8.3.5 DM 05 01 and DM 05 04

- **Verbal Description**
  Both testruns included the Metropolis criterion explained in section 4.3.7. These approaches differed actually only in the mixing function used. While the first approach only intersected the data entries, the second approach had an adding function for OPHID entries included. The dynamics continued to change for a higher number of update steps in these runs, so the number of iterations was increased.

- **Environment and settings for DM 05 01**
  - f_{assess} : Assessment function 3 - Division by Average Filling
  - $f_{neighbor}$: Incident Rotation Edges - Check Random for All Sorted with Metropolis - No Reconsideration

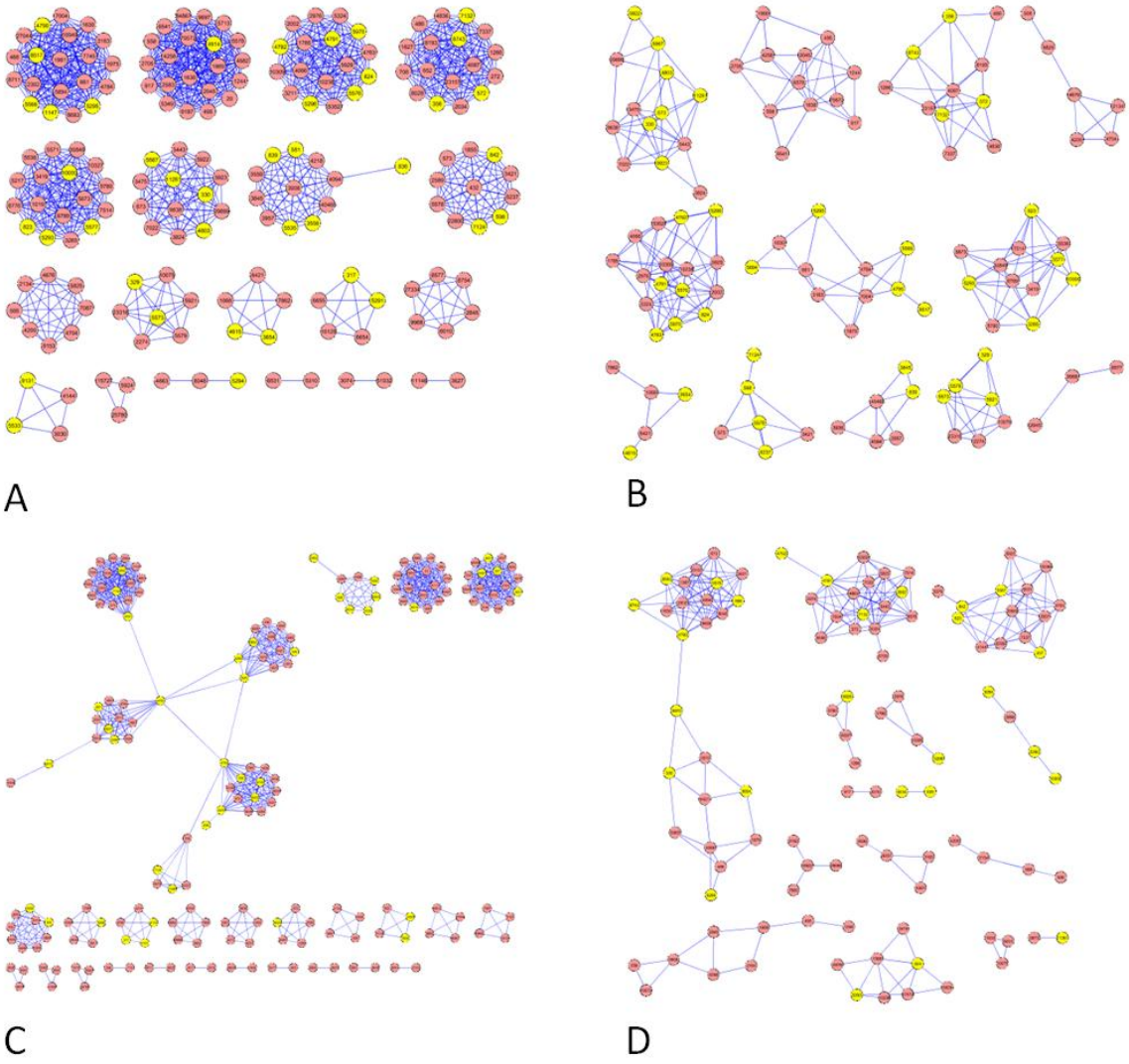

Figure 34: Cytoscape graph visualization in iteration 1000 for a cutoff of 80% for DM 04 04 in A and DM 04 07 in C. For a cutoff of 90% for DM 04 04 see B and for DM 0 07 see D.

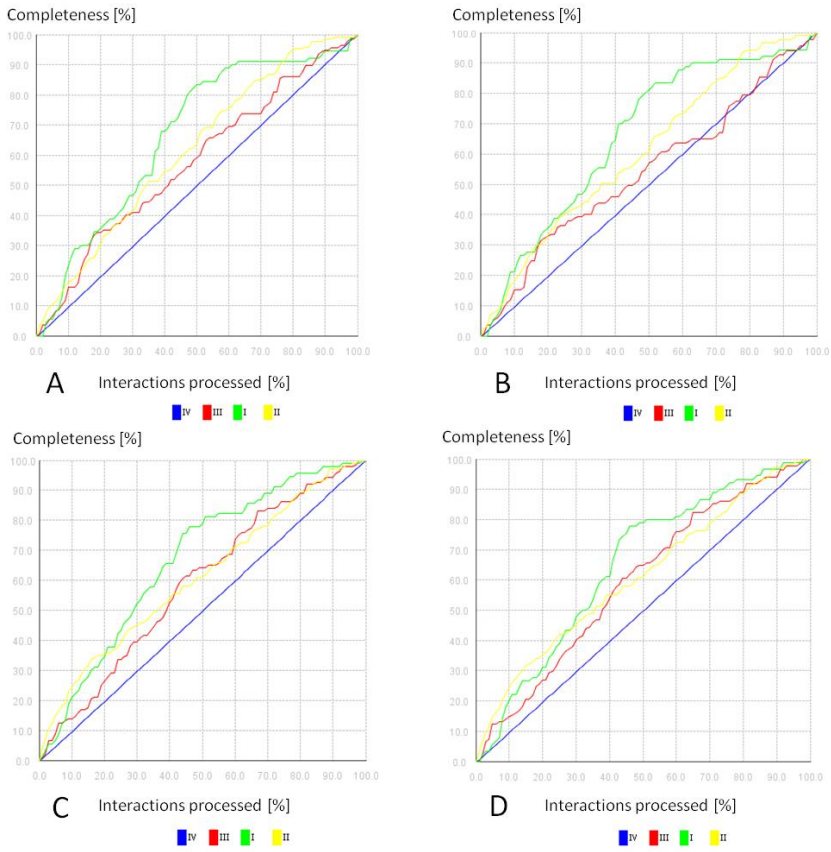

Figure 35: KEGG verification for DM 04 04 for 100 (A) and 1000 (B) iterations and the same for DM 04 07: 100 (C) 1000 (D). The 'Completeness' on the y-axis denotes the overlap of processed interactions of the *interaction list* with the *reference list* holding the KEGG interactions and 'Interactions processed' refers to the percentage of interactions already processed from the *interaction list* as described in section 7.2.1.

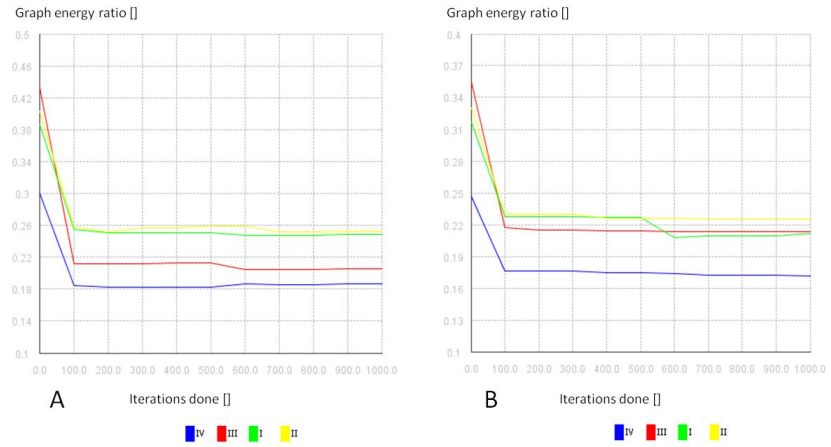

Figure 36: Graph energy for DM 04 04 in A and DM 04 07 in B plotted versus the iteration steps.

  - f_{mix} : Intersect from All
  - n_{it} : 2000
  - $n_{analysis}$: 100

- **Environment and settings for DM 05 04**
  - f_{assess} : Assessment function 3 - Division by Average Filling
  - $f_{neighbor}$: Incident Rotation Edges - Check Random for All Sorted with Metropolis - No Reconsideration
  - f_{mix} : Intersect from All Except OPHID - Add To OPHID Restricted
  - n_{it} : 2000
  - $n_{analysis}$: 100

- **Iteration 100 and 2000 for DM 05 01 and DM 05 04**
  - *Graph characteristics in iteration 100 for given offsets for DM 05 01*

Step	Cutoff	No. of interactions	Graph size	Largest subgraph	No. of subgraphs
1	0.0	49770	316	316	1
2	0.11310285291900402	30945	315	315	1
3	0.22620570583800803	25760	314	314	1
4	0.33930855875701205	10630	308	305	2
5	0.45241141167601606	4193	301	273	9
6	0.5655142645950201	1442	260	207	11
7	0.6786171175140241	873	200	114	16
8	0.7917199704330281	217	116	22	17
9	0.9048228233520321	24	25	7	7
10	1.017925676271036	7	8	5	2

  - *Graph characteristics in iteration 100 for given offsets for DM 05 04*

Step	Cutoff	No. of interactions	Graph size	Largest subgraph	No. of subgraphs
1	0.0	49770	316	316	1
2	0.10248152709710659	32861	315	315	1
3	0.20496305419421318	23908	314	314	1
4	0.3074445812913198	13934	311	311	1
5	0.40992610838842636	5817	307	290	6
6	0.5124076354855329	2325	261	203	10
7	0.6148891625826396	1052	227	102	21
8	0.7173706896797462	753	183	38	22
9	0.8198522167768527	114	87	13	17
10	0.9223337438739593	2	4	2	2

– *Graph characteristics in iteration 2000 for given offsets for DM 05 01*

Step	Cutoff	No. of interactions	Graph size	Largest subgraph	No. of subgraphs
1	0.0	49770	316	316	1
2	0.11310285291900402	31045	315	315	1
3	0.22620570583800803	26638	314	314	1
4	0.33930855875701205	10056	308	305	2
5	0.45241141167601606	4527	301	269	11
6	0.5655142645950201	1645	260	199	12
7	0.6786171175140241	1063	196	108	16
8	0.7917199704330281	262	111	36	15
9	0.9048228233520321	24	26	6	8
10	1.017925676271036	6	7	4	2

– *Graph characteristics in iteration 2000 for given offsets for DM 05 04*

Step	Cutoff	No. of interactions	Graph size	Largest subgraph	No. of subgraphs
1	0.0	49770	316	316	1
2	0.10248152709710659	30746	314	314	1
3	0.20496305419421318	23533	313	313	1
4	0.3074445812913198	11937	311	311	1
5	0.40992610838842636	4734	307	290	5
6	0.5124076354855329	1941	263	119	14
7	0.6148891625826396	1190	223	79	21
8	0.7173706896797462	1005	176	46	20
9	0.8198522167768527	124	82	18	13
10	0.9223337438739593	2	4	2	2

– *General interpretation of subgraph behavior for all given offsets*
The creation of subgraphs proceeded fast in both runs. Run DM 05 01 did split into a higher number subgraphs than DM 05 04. This finding might have been caused by the mixing function of DM 05 04, which adds strong OPHID entries connecting already strongly clustered subgraps by single interactions (observable in figure 37). Focusing on cutoffs between 70 and 90%, interesting findings could be observed. Generally we can say, that with proceeding propagation number of nodes decreases, while the number of iterations (only for high cutoffs) increases. This fact indicates a separation of edges and significant clustering as confirmed by graph visualization later on. Another interesting behavior is that DM 05 04 holds 116 nodes and contains 40 nodes from the apoptosis pathway for a cutoff of 80% after the 100^{th} iteration . After the 2000^{th} iteration the graph has only 111 nodes, but contains two nodes more from the apoptosis network with an overall of 42 nodes. The number of interactions was also increasing, as usual in these runs.

– *Graph Measures*
For DM 05 01: The *assortative mixing coefficient* and the *betweenness* exhibited a significant change from iteration 100 to 2000. For iteration 2000 the edge-edge distribution (*assortative mixing coefficient*) is high (≈ 45) compared to iteration 100 (≈ 20) for a cutoff of 70% (for a cutoff of 60% it is ≈ 8 for 100 iterations and ≈ 17 for 2000 iterations). This indicates that the edges distribution in the graph becomes more unequal. Slight differences could be observed for the *entropy of the distribution of edges, stress centrality, Eigen values, clustering coefficient, graph centrality, index of aggregation, graph diameter, the Wiener numbers and connectivity*.
For DM 05 04: The *entropy of the distribution of edges, stress centrality* and the *be-*

tweenness exhibited a significant change from iteration 100 to 2000. The *betweeness* is much lower for a cutoff of 60% at iteration 2000 (≈ 270) instead of ≈ 300 at iteration 100. This is exactly where the graph begins to split into subgraphs. The stress centrality shows a noticeable change for a cutoff of 50% ($1.2x10^7$ at 100 iterations but only $0.5x10^7$ at iteration 2000). This might imply that the graph breaks into subgraphs earlier, or that there are more edges connecting clusters. Slight differences could be observed for the *clustering coefficient, graph diameter, graph centrality, assortative mixing coefficient, index of aggregation, graph diameter, Wiener numbers* and *connectivity*.

- *Cytoscape graph interpretation*
  As given in figure 37 run DM 05 01 shows a very interesting, self-amplifying behavior. For a cutoff of 80% after iteration 100 the graph contains a few subgraphs holding overall 316 interactions, and after 2000 iterations there were 1005 interactions in strongly clustered subgraphs which emerged from the system. The amplifying process started at about iteration 500, as indicated by the graph energy (figure 40). In figure 38 there is no such amplifying emerging effect, but as already mentioned above, the objects from the apoptosis pathway found in the graph increased during the iterations at the same cutoff.
- *KEGG overlap*
  The KEGG behavior was also interesting (figure 39). For DM 05 01 pathway PIII performed bad, but still, the overlap improved at the end of the update steps again. A very interesting behavior has been observable for DM 05 04: while pathway PIII performed well at the start and decreased during the update steps, the overlap of pathway PI seemed to improve. PII was stable after 100 iterations and did not change anymore.

- **Overall system characteristics**
  - *Graph energy*
    As shown in figure 40, the system showed dynamics throughout the whole update cycle. Nevertheless, the major impact occurred to be during the first 100 iterations. An interesting observation was a spike occurring in graph energy of pathway PIII in DM 05 04. The graph energy suddenly rose, before decreasing even stronger afterwards.
  - *Conclusion*
    These last testruns were the most interesting ones with respect to our initially defined goals. These testruns showed dynamic behavior throughout the entire simulation and the best KEGG performance. Furthermore the system exhibited a not explicitly implemented property emerging from the data (the heavily clustered subgraphs). As one additional aspect, we showed that - even if a pathway first showed weak representation - the system's change increase the overlap for the pathway again. A local optimum does not necessarily have to be the global one.

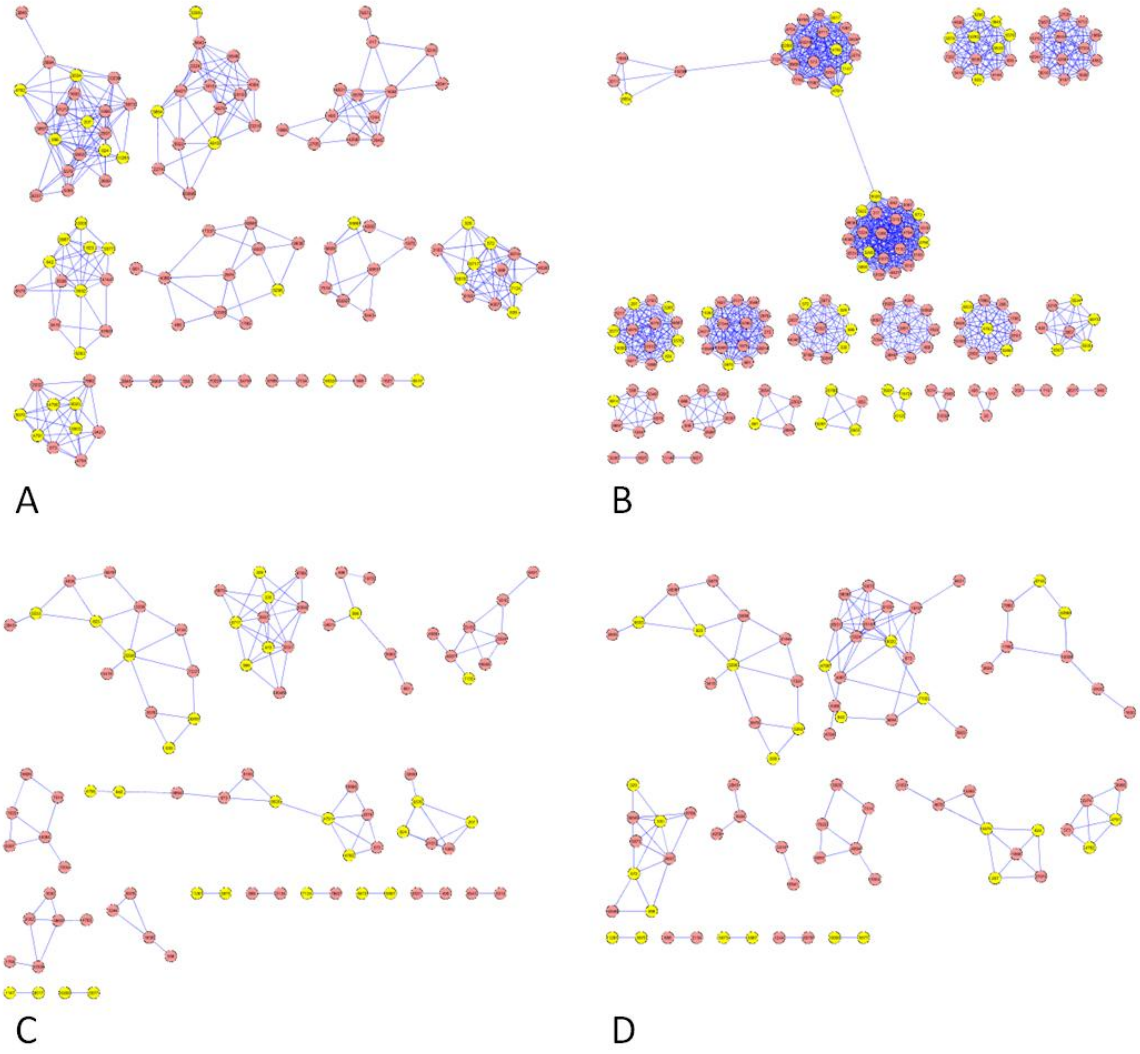

Figure 37: Cytoscape graph visualization for a cutoff of 80% for DM 05 01 after 100 (A) and 2000 (B) iterations. The second run shows DM 05 01 at a cutoff of 90% for 100 (C) and 2000 (D) iterations.

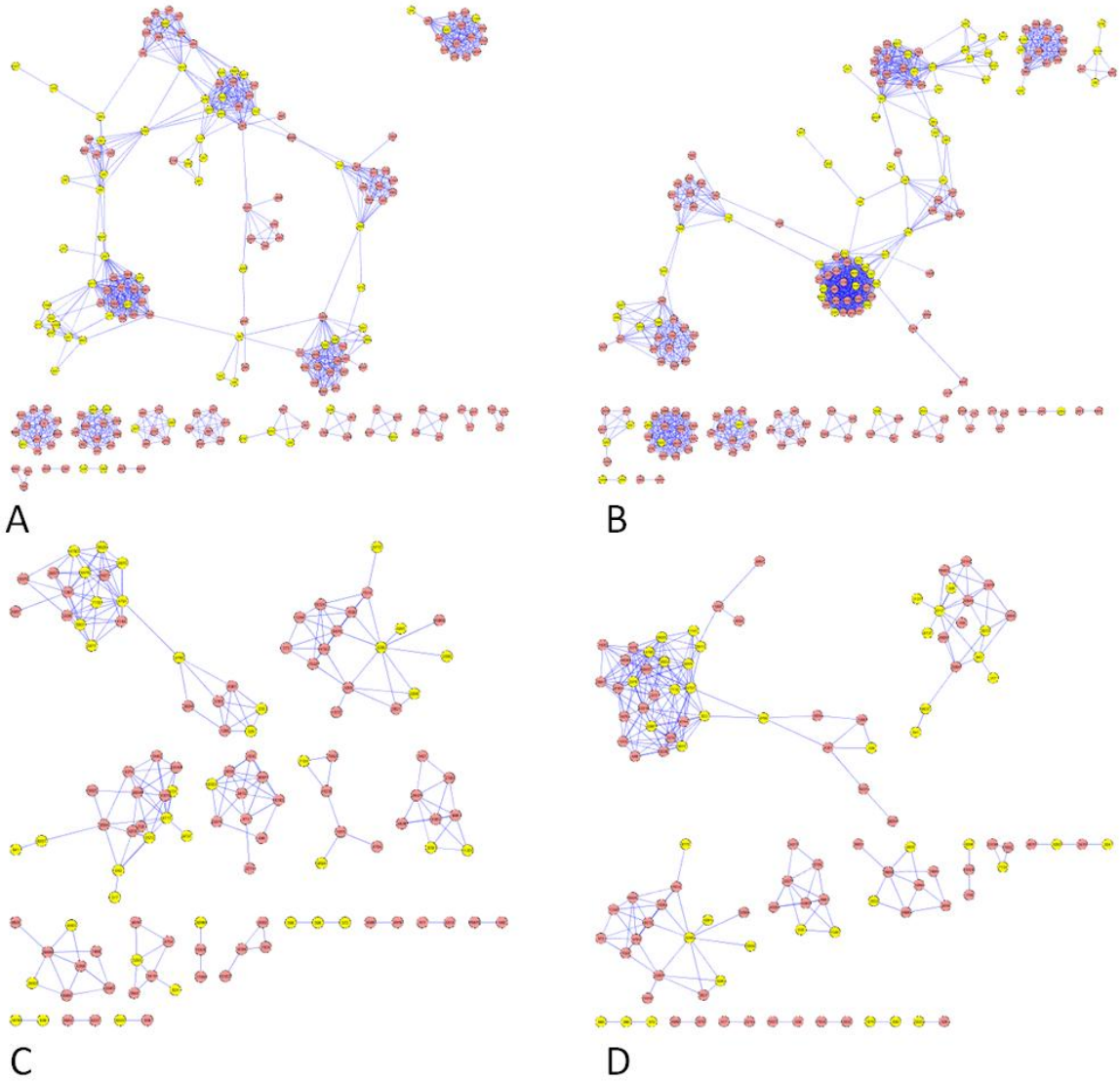

Figure 38: Cytoscape graph visualization for a cutoff of 70% for DM 05 04 after 100 (A) and 2000 (B) iterations. The second run shows DM 05 04 at a cutoff of 80% for 100 (C) and 2000 (D) iterations.

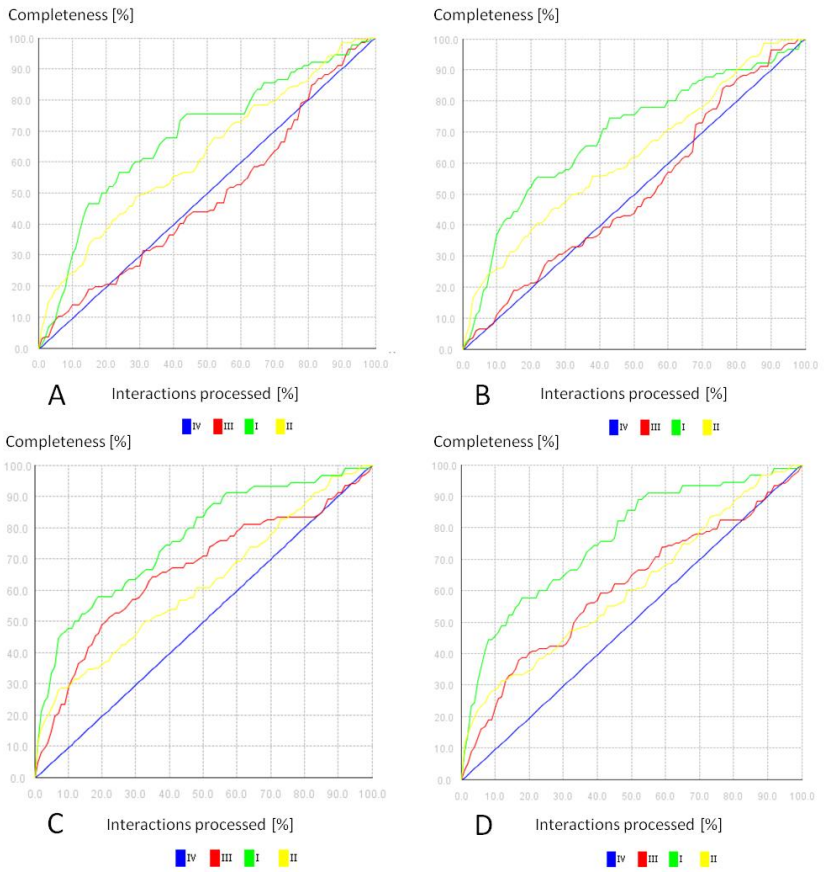

Figure 39: KEGG verification for DM 05 01 for 100 (A) and 1000 (B) iterations and the same for DM 05 04: 100 (C) 1000 (D). The 'Completeness' on the y-axis denotes the overlap of processed interactions of the *interaction list* with the *reference list* holding the KEGG interactions and 'Interactions processed' refers to the percentage of interactions already processed from the *interaction list* as described in section 7.2.1.

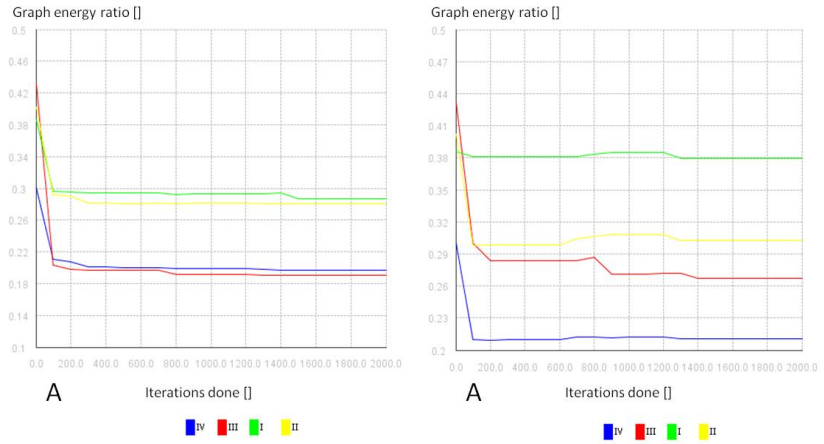

Figure 40: Graph energy for DM 05 01 in A and DM 05 04 in B plotted versus the iteration steps.

Overall we can say, that a distributed way of a system update resulted in a considerably more promising behavior, and that the pathway reconstruction increased in its overlap constantly. What is left for further research now is an examination of neighborhood functions including more than two participating objects. Another task is the mixing function: a neighborhood declaring more than two objects being neighbors would not just change the update **sequence**. This change would even alter the possibilities of the mixing functions. Mixing with the information of more than two objects might result in much better KEGG benchmarks.

9 Conclusion

This thesis introduces a novel simulation approach aimed at revealing functional context on the basis of heterogenous biological data sources. The general situation is given as follows: Data generation on intra-cellular events was boosted significantly following the introduction of novel high throughput screening methods - briefly called 'omics' technologies. These technologies focus on distinct intracellular levels, as transcriptomics on determining whole cell RNA concentration profiles and proteomics on deciphering overall protein abundance. Dedicated analysis routines have been established for analyzing transcriptomics, co-regulation, protein interaction etc., but approaches allowing joint analysis of all data sets have been limited.

Systems Biology is aimed at bridging this gap by combining these multi-level data in a unified framework. This approach is mandatory for interlinking data representing different functional hierarchies towards deriving context, where context is defined as deciphering functional dependencies - which itself is the overall goal of these endeavors: Understanding organizational principles, functional context, and key molecular players for a given cellular state. In the application context this means e.g. biomarker discovery, i.e. usually proteins which are directly associated to a certain disease which in turn can be used for developing novel diagnostics and therapy approaches.

Based on this given background this thesis was aimed at testing the mathematical concept of hyperstructures and emergent properties for its use in Systems Biology. Hyperstructures denote a formal organizational principle, where objects solely due to local interaction organize in hierarchical levels stabilized by both, up- and downward causalities. A consequence of such hierarchical organization is the emergence of properties (e.g. functionalities) which are only observable in the organizational context of the hyperstructure, but not on the level of individual objects. Intracellular functionalities may exactly be seen as such structures: Single objects (e.g. proteins) interact locally, thereby generating various feedback loops - which by itself drive a dynamics with emergent properties seen as phenotypic observables as cell proliferation or apoptosis.
If intracellular dynamics is characterized by dynamical hierarchies and emergent properties the formal concept as such should be suited for representing the system via embedding data generated by 'omics' technologies.

We first designed a representation of intracellular objects, including omics data from differential gene expression, transcription factor co-regulation, protein interaction, functional categories, and intracellular location. After encoding this representation as well as functionalities for data import we optimized object interaction functions for representing similarities. The concept here is that objects with, from a biological perspective, similar or synergistic properties will be in closer functional context. In our setup similarity is encoded as differential weights of edges between nodes

(objects) when representing all objects as undirected graph.
Next, different object neighborhoods for calculating edge weights were analyzed, and the consequence of changing neighborhoods with respect to computing weights was monitored. Both issues, computing a similarity matrix under varying similarity functions on the basis of included omics data, as well as using different neighborhood constraints were then tested in varying update functionalities: Within an update step we aimed at reaching 'consensus' between pairs of objects, where consensus was defined as biologically plausible interactions: Pairs with high correlation in differential gene expression might show joint transcription factor binding sites; Pairs embedded in the same functional context might be found in the same intracellular compartment.
Our update functions were designed to evaluate following such biological rationale, and then actively changing the data entries within objects for aligning consensus. By doing so we aimed at a) reducing the rate of false positive entries as given by the raw omics data, and b) replacing missing values based on biological plausibility. If successful this procedure should reach a final state of maximum consensus between all objects - which in turn should be the state representing functional dependencies as represented by high weight edges on the interaction graph level.

We used three, experimentally well described functional pathways for testing our system: Two representing parts of the apoptosis cascade, and one holding a part of the MAPK signaling pathway. Goal of the system modeling is now a de novo reconstruction of these functional pathways solely based on first level interactions - i.e. pair-wise object interactions as encoded in our similarity matrix.
We systematically explored different core data sets, update functionalities, and neighborhood functions, followed by analyzing resulting interaction graphs by using traditional measures from graph theory, as well as by checking the degree of overlap between stable sub-graphs and the given pathways from apoptosis and MAPK.

The optimal variant of our modeling framework finally reached significant representation of the apoptosis pathway - providing the Proof of Concept for our approach.
We feel confident that expanding this concept for covering the entire human proteome, complemented by further expansion of omics sources included in our object representation, has the potential of contributing to Systems Biology towards increasing our knowledge on intracellular events in both, basic as well as applied research.

A Tools Used

A.0.6 Programming of the System

The system was realized in Sun Java 1.5 J2SE. For execution at least a JRE 1.5 compliant Java Runtime is needed.

A.0.7 Diagrams

The 2D- and 3D-plots have been created with support of the JMathTools-library.

http://jmathtools.sourceforge.net/index.php

A.0.8 Visualization of the Graph Networks

At the beginning the visual analysis of the graph networks was done in ProteoLens. Later on all visualization has been done in Cytoscape (Cytoscape, 2006; Proteolens, 2006).

Cytoscape 2.4.0.

http://www.cytoscape.org

A.0.9 Boxplots and Statistical Analysis

Statistical analysis was performed in R.

http://www.r-project.org

References

A. Valencia, F. P. (2002). *Current Opinion in Structural Biology,* **12**, 368–373.

Affymetrix (2000).

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). *Journal of Molecular Biology,* **215** (3), 403–410.

Ambesi-Impiombato, A. & di Bernardo, D. (2006). *Current Bioinformatics,* **1**, 3–13.

Anthony, R. (2004). Proceedings of the International Conference on Autonomic Computing (ICAC04).

Apic, G., Ignjatovic, T., Boyer, S., & Russell, R. (2005). *FEBS Letters,* **579** (8), 1872–1877.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., & Sherlock, G. (2000). *Nature Genetetics,* **25** (1), 25–29.

B. Mayer, S. R. (1998). In: *ALIFE: Proceedings of the sixth international conference on Artificial life* pp. 123–139, Cambridge, MA, USA: MIT Press.

Baas, N. (1992). *Artificial Life,* **3**, 515–537.

Baas, N. & Emmeche, C. (1997). *Intellectica,* **2** (25), 67–83.

Bar-Yam, Y. (1997). *Dynamics of Complex Systems (Studies in Nonlinearity)*. Addison-Wesley.

Bayon, R. & Lygeros, N. (2006). *Hyperstructures and Automorphism Groups*. UniversitÃ© Lyon I.

Bilotta, E., Gross, D., Smith, T., Lenaerts, T., Bullock, S., Lund, H., Bird, J., Watson, R., Pantano, P., Pagliarini, L., Abbass, H., Standish, R., & Bedau, M., eds (2002). *ALifeVIII: Workshop Proceedings* Sidney, Australia.

Bina, J., Nano, F., & Hancock, R. (1997). *FEMS Microbiology Letters,* **148** (1), 63–68.

BIOCarta (2006). http://www.biocarta.com/.

BIOGrid (2006). http://www.thebiogrid.org/.

Biron, D., Brun, C., Lefevre, T., Lebarbenchon, C., Loxdale, H., Chevenet, F., Brizard, J., & Thomas, F. (2006). *Proteomics,* **6**, 5577–96.

BOND (a). http://bond.unleashedinformatics.com/.

Brent, R. (2004). *Nature Biotechnology,* **22** (10), 1211–4.

Brown, K. R. & Jurisica, I. (2005). *Bioinformatics,* **21** (9), 2076–2082.

Butcher, E., Berg, E., & Kunkel, E. (2004). *Nature Biotechnology,* **22** (10), 1253–1259.

Chen, H. (2005). *Prediction of Protein Structures and Protein-Protein Interactions: A Bioinformatics Approach*. PhD thesis Drexel University, Philadelphia.

Cytoscape (2006). http://www.cytoscape.org/.

da Silva, J. M., Lemke, N., Mombach, J., de Souza, J. C., Sinigaglia, M., & Vieira, R. (2006). *Genetics and Molecular Research,* **5** (1), 182–191.

De Wolf, T., Jaco, L., Holvoet, T., & Steegmans, E. (2002). In: *LNCS 2463, Ant Algorithms* Lecture Notes in Computer Science pp. 290–291,. URL = http://iridia.ulb.ac.be/˜ants/ants2002/.

D.L. Gumucio, D.A. Shelton, W. B. J. S. & Goodman, M. (1993). *Proceedings of the Natural Academy of Sciences,* **90**, 6018–6022.

Dorin, A. & McCormack, J. (2002). In: *Proceedings of Alife 8*, (et al, S., ed) pp. 423–428, MIT Press.

Dubowsky, S., Plante, J., & Boston, P. (2006). In: *Security and Rescue Robotics* , Gaithersburg, MD, USA:.

E-Cell (2006). http://www.e-cell.org/software/e-cell-system.

EBI (2006). http://www.ebi.ac.uk/.

EMBL (2006). http://www.embl.org/.

Fields, S. & Song, O. (1989). *Nature,* **340** (6230), 245–246.

GeneOntology (2006). www.geneontology.org.

Gianchandani, E., Papin, J., Price, N., Joyce, A. R., & Palsson, B. (2006). *PLoS Computational Biology,* **2** (8), e101.

Goldstein, J. (1999). *Emergence,* **1** (1), 49–72.

Gordon, D. (2002/2003). *Ant-based Pathfinding - Artificial Intelligence with Philosophy*. University of Leeds, School of Computing final year undergraduate projects edition.

Gross, D. & McMullin, B. (2002). *Artificial Life,* **7** (4), 355–365.

Hartmann, N. (1940). *Der Aufbau der realen Welt*. de Gruyter.

Ho, J. S., Mortimer, J., Arenillas, D., Brumm, J., Walsh, C., Kennedy, B., & Wasserman, W. (2005). *Nucleic Acids Research,* **33** (10), 3154–3164.

HPRD (2006). http://www.hprd.org/.

Hucka, M., Finney, A., Bornstein, B., Keating, S., Shapiro, B., Matthews, J., Kovitz, B., Schilstra, M., Funahashi, A., Doyle, J., & Kitano, H. (2004). *IEE Systems Biology,* **1** (1), 41–53.

Jacobi, M. (2005). *Artificial Life,* **11** (4), 493–512.

J.B. Welsh, P.P. Zarrinkar, L. S. S. K. C. B. B. M. D. L. R. B. G. H. (2001). *Proc. Natl. Acad. Sci. USA,* **98** (3), 1176–1181.

J.K. Joung, E. R. & Pabo, C. (2000). *Proceedings of the National Academy of Sciences,* **97** (13), 7382–7387.

Joyce, A. & Palsson, B. (2006). *Nature Molecular Cell Biology,* **7**, 198–210.

K. Nishikawa, T. O. (1982). *Journal of Biochemistry (Tokyo).* **91**, 1821–1824.

KEGG (2006). http://www.genome.jp/kegg/.

Kenri, T., Seto, S., Horino, A., Sasaki, Y., Sasaki, T., , & Miyata, M. (2004). *Journal of Bacteriology,* **186** (20), 6944–6955.

Kielbasa, S., Gonze, D., & Herzel, H. (2005). *BMC Bioinformatics,* **6**, 237.

Knippers, R. (2001). *Molekulare Genetik*. Georg Thieme Verlag.

Korber, B., Labute, M., & Yusim, K. (2006). *PLoS Computational Biology,* **2** (6), e71.

Kumar, R., Xie, Y., & Das, A. (2000). *Molecular Microbiology,* **36** (3), 608–617.

LabVantage (2006). http://www.labvantage.com/.

Lenaerts, T., Chu, D., & Watson, R. (2005). *Artificial Life,* **11** (4), 403–405.

Lenaerts, T. & Gross, D. (2002). In: *ALifeVIII: Workshop Proceedings*, (Bilotta, E., Gross, D., Smith, T., Lenaerts, T., Bullock, S., Lund, H., Bird, J., Watson, R., Pantano, P., Pagliarini, L., Abbass, H., Standish, R., & Bedau, M., eds) pp. 45–55,.

Li, H. (2006). *Efficient Discovery of Binding Motif Pairs from Protein-Protein Interactions*. PhD thesis National University of Singapore.

Liang, P. & Pardee, A. (1992). *Science,* **14** (257), 967–971.

Lipshutz, R. J., Fodor, S. P., Gingeras, T. R., & Lockhart, D. J. (1999). *Nature Genetics,* **21** (1), 20–24.

Lygeros, N. & Vougiouklis, T., eds (2005). *Structure Elements of Hyperstructures*. Spanidis Press.

Marcotte, E., Pellegrini, M., Thompson, M., Yeates, T., & Eisenberg, D. (1999). *Nature,* **402**, 83–86.

Matys, V., Fricke, E., Geffers, R., Gössling, E., Haubrock, M., Hehl, R., Hornischer, K., Karas, D., Kel, A. E., Kel-Margoulis, O. V., Kloos, D.-U., Land, S., Lewicki-Potapov, B., Michael, H., Münch, R., Reuter, I., Rotert, S., Saxel, H., Scheer, M., Thiele, S., & Wingender, E. (2003). *Nucleic Acids Research,* **31** (1), 374–378.

McGinnis, S. & Madden, T. (2004). *Nucleic Acids Research,* **32**, W20–W25.

McGregor, S. & Fernando, C. (2005). *Artificial Life,* **11** (4), 459–472.

Mesarovic, M., Sreenath, S., & Keene, J. (2004). *IEE Systems Biology,* **1** (1), 19–27.

Metropolis, N., Rosenbluth, A., M.N.Rosenbluth, Teller, A., & Teller, E. (1953). *The Journal of Chemical Physics,* **21** (6), 1087–1092.

Microarrays (2006). http://www.ebi.ac.uk/microarray/.

MIPS (2006). http://mips.gsf.de/.

Nakai, K. & Horton, P. (1999). *Trends Biochemical Science,* **24** (1), 34–35.

NCBI (2006). http://www.ncbi.nlm.nih.gov/.

Ng, A., Bursteinas, B., Gao, Q., Mollison, E., & Zvelebil, M. (2006). *Nucleic Acids Research,* **34**, 527–534.

Nikolsky, Y., Nikolskaya, T., & Bugrim, A. (2005). *Drug Discovery Today,* **10** (9), 653–62.

Nitschke, G. (2005). *Artificial Life,* **11** (3), 367–396.

OMG (2006). http://www.omg.org/.

Ono, N. & Ikegami, T. (1999). *Model of self-replicating cell capable of self-maintenance.* University of Tokyo, Institute of Physics, Graduate School of Sciences unpublished edition.

OPHID (2006). http://ophid.utoronto.ca/ophid/.

PANTHER (2006). http://www.pantherdb.org/.

PDB (2006). http://www.rcsb.org/pdb/home/home.do.

Perco, P., Rapberger, R., Siehs, C., Lukas, A., Oberbauer, R., Mayer, G., & Mayer, B. (2006). *Electrophoresis,* **27** (13), 2659–2675.

Phizicky, E. M. & Fields, S. (1995). *Microbiological Reviews,* **59** (1), 94–123.

Platzer, A., Perco, P., Lukas, A., & Mayer, B. (submiitted 2006). Characterization of protein interaction networks in tumors. BMC Bioinformatics.

Popescu, M., Keller, J., & Mitchell, J. (2006). *IEEE/ACM Transactions on Computational Biology and Bioinformatics,* **3** (3), 263–274.

Porollo, A. (2006). http://info.cchmc.org/presentations/porollo_21feb03.ppt.

Prokopenko, M., Wang, P., Valencia, P., Price, D., Foreman, M., & Farmer, A. (2005). *Artificial Life,* **11** (4), 407–426.

Proteolens (2006). http://bio.informatics.iupui.edu/proteolens/index.stm.

Qiu, Z. (2004). *Hyperstructure-Based Search Methods for the World Wide Web*. PhD thesis Technische Universität Darmstadt.

Rapberger, R. (2007). *A Systems Biology Approach towards Identification of Desease Associated Proteins*. PhD thesis University of Vienna.

Rasmussen, S., Baas, N., Barret, C., & Olesen, M. (1996). *A Note on Simulation and Dynamical Hierarchies*. Santa Fe Internal Working Paper.

Rasmussen, S., Baas, N. A., Mayer, B., & Nilsson, M. (2002). *Artif. Life,* **7** (4), 367–373.

Regenmortel, M. V. (2006). *Journal of Molecular Recognition,* **19** (3), 183–187.

Roos, D. (2001). *Science,* **291** (5507), 1260–1261.

Rosenberg, A. (1985). *The Structure of Biological Science*. Cambridge University Press.

Rowe, J., Vose, M., & Wright, A. (2005). *Artificial Life,* **11** (4), 473–492.

Sandelin, A., Alkema, W., Engström, P., Wasserman, W., & Lenhard, B. (2004). *Nucleic Acids Research,* **32**, 91–94.

Shannon, C. (1948). *The Bell System Technical Journal,* **27**, 379–423,623–656.

Simon, H. (1973). In: *Hierarchy Theory - The Challenge of Complex Systems*, (Pattee, H., ed) pp. 1–27. Goerge Braziller New York.

Simon, H. (1996). *The sciences of the artificial (3rd ed.)*. Cambridge, MA, USA: MIT Press.

Tanenbaum, A. & van Steen, M. (2003). *Distributed Systems: Principles and Paradigms*. Prentice Hall.

Taylor, R. C., Treatman, A. C., & Blevins, M. (2006). *Bioinformatics,* **22** (21), 2706–2708.

van Criekinge, W. & Beyaert, R. (1999). *Biological Procedures Online,* **2** (1), 1–38.

Vass, M., Shaffer, C., Ramakrishnan, N., Watson, L., & Tyson, J. (2006). *IEEE/ACM Transactions on Computational Biology and Bioinformatics,* **3** (2), 155–164.

von Mering, C., Jensen, L., Snel, B., Hooper, S., Krupp, M., Foglierini, M., Jouffre, N., Huynen, M., & Bork, P. (2005). *Nucleic Acids Research,* **33**, 433–437.

Watson, R. & Pollack, J. (2005). *Artificial Life,* **11** (4), 445–457.

Wolf, T. D. & Holvoet, T. (2005a). In: *Engineering Self-Organising Systems Methodologies and Applications*, (Springer, ed) volume 3464 of *Lecture Notes in Computer Science* pp. 1–15,.

Wolf, T. D. & Holvoet, T. (2005b). *Self-Organization and Autonomic Informatics (I),* **135**, 18–34.

Wolf, T. D. & Holvoet, T. (2006). Proceedings of the Joint Smart Grid Technologies (SGT) and Engineering Emergence for Autonomic Systems (EEAS) Workshop.

Wolf, T. D., Samaey, G., Holvoet, T., & Roose, D. (2005). Proceedings of the Second International Conference on Autonomic Computing (ICAC05).

Yalamanchili, N., Zak, D., Ogunnaike, B., Schwaber, J., Kriete, A., & Kholodenko, B. (2006). *IEE Proceedings - Systems Biology,* **153** (4), 236–246.

You, L. (2004). *Cell Biochemistry and Biophysics,* **40** (2), 167–184.