Lecture Notes in Bioinformatics 3380

Edited by S. Istrail, P. Pevzner, and M. Waterman

Subseries of Lecture Notes in Computer Science

Corrado Priami (Ed.)

Transactions on Computational Systems Biology I

Springer

Editor-in-Chief

Corrado Priami
Università di Trento
Dipartimento di Informatica e Telecomunicazioni
Via Sommarive, 14, 38050 Povo (TN), Italy
E-mail: priami@dit.unitn.it

Library of Congress Control Number: 2005922555

CR Subject Classification (1998): J.3, H.2.8, F.1

ISSN 0302-9743
ISBN 3-540-25422-6 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11410232 06/3142 5 4 3 2 1 0

Preface

This is the first issue of a new journal of the LNCS journal subline. The aim of the journal is to encourage inter- and multidisciplinary research in the fields of computer science and life sciences. The recent paradigmatic shift in biology towards a system view of biological phenomena requires a corresponding paradigmatic shift in the techniques from computer science that can face the new challenges. Classical tools usually used in bioinformatics are no longer up to date and new ideas are needed.

The convergence of sciences and technologies we are experiencing these days is changing the classical terms of reference for research activities. In fact clear distinctions between disciplines no longer exist because advances in one field permit advances in others and vice versa, thus establishing a positive feedback loop between sciences. The potential impact of the convergence of sciences and technologies is so huge that we must consider how to control and correctly drive our future activities.

International and national funding agencies are looking at interdisciplinary research as a key issue for the coming years, especially in the intersection of life sciences and information technology. To speed up this process, we surely need to establish relationships between researchers of different communities and to define a common language that will allow them to exchange ideas and results. Furthermore, expectations of different communities can be merged only by running activities like common projects and experiences.

The Transactions on Computational Systems Biology could be a good forum to help life scientists and computer scientists to discuss together their common goals.

This first issue is made up of contributions by members of the Editorial Board to provide a smooth start-up of the journal. The first paper, by Gómez et al., surveys the new methods needed for acquiring data suitable to enable simulation of simple cellular systems. Then Shenhav et al. discuss how system biology can be of help also in studying very early organisms in the time evolution scale. The third contribution is by Roux-Rouquié and Soto and shows how useful is the notion of model and metamodel in the systemic approach to the study of biological systems. Feytmans et al. investigate the relationships between the complexity of a genome and the functional complexity arising from it. The next paper moves inside computer science. Priami and Quaglia show how calculi for describing concurrent systems can be used to model biological systems as well. The last invited contribution is by Uhrmacher et al. and copes with the problem of multilevel and multiscale simulation. Finally, Zobeley et al., in the regular paper of this issue, present a new complexity reduction method which is time-dependent and suited not only for steady states, but for all possible dynamics of a biochemical system.

Trento, January 10, 2005 Corrado Priami

LNCS Transactions on Computational Systems Biology – Editorial Board

Table of Contents

Accessible Protein Interaction Data for Network Modeling. Structure of the Information and Available Repositories

Manuel Gómez[1], Ramón Alonso-Allende[2], Florencio Pazos[3],
Osvaldo Graña[4], David Juan[4], and Alfonso Valencia[4]

[1] Centro de Astrobiología (CSIC/INTA),
Instituto Nacional de Técnica Aeroespacial,
Ctra de Torrejón a Ajalvir, km 4,
28850 Torrejón de Ardoz, Madrid, Spain
[2] Bioalma, Ronda de Poniente, 4 - 2nd floor, Unit C-D,
28760 Tres Cantos, Madrid, Spain
[3] Structural Bioinformatics Group,
Biochemistry Building,
Department of Biological Sciences,
Imperial College, London SW7 2AZ, U.K
[4] Protein Design Group,
National Center for Biotechnology (C.N.B. - C.S.I.C.),
Cantoblanco, E-28049 Madrid, Spain
valencia@cnb.uam.es

Abstract. In recent years there has been an incredible explosion of computational studies of molecular biology systems, particularly those related to the analysis of the structure and organization of molecular networks, as the initial steps toward the possible simulation of the behavior of simple cellular systems. Needless to say, this task will not be possible without the availability of a new class of data derived from experimental proteomics. Large-scale application of the yeast two-hybrid system, affinity purification (TAPs-MS), and other methodologies are for the first time providing overviews of complete protein interaction networks. Interestingly a number of computational methods are also contributing substantially to the identification of protein interactions, by comparing genome organization and evolution. Other disciplines, such as structural biology and computational structural biology, are complementing the information on interaction networks by providing detailed molecular descriptions of the corresponding complexes, which will become essential for the direct manipulation of the networks using theoretical or experimental methods. The storage, manipulation and visualization of the huge volumes of information about protein interactions and networks pose similar problems, irrespective of the source of the information: experimental or computational. In this sense, a number of competing systems and emerging standards have appeared in parallel with the publication of the data. In this review, we will provide an overview of the main experimental, high-throughput methods for the study of protein interactions, the parallel developments of computational methods for the prediction of protein interactions based on genome and sequence information, and the development of databases and standards that facilitate the analysis of all this information.

C. Priami et al. (Eds.): Transactions on Computational Systems Biology, LNBI 3380, pp. 1 – 13, 2005.

1 Introduction

Proteins are involved in key cellular processes, including signal transduction, metabolism, cellular architecture and information transfer. To carry out these functions, proteins interact to form complexes of varying nature and stability, from stable interactions of structural proteins to transient contacts modulated by post-translational modifications, as is typical of signaling proteins.

During the last few years, proteomics has produced spectacular advances in the description of these complexes, utilizing high-throughput techniques such as systematic yeast 2-hybrid approaches [1-4], Tandem Affinity purification followed by Mass Spectrometry resolution of the isolated complexes [5], and various combinations of information obtained from peptide libraries [6, 7]. Other techniques, such as chromatin immunoprecipitation (ChIP), have systematically addressed the relationship between transcription factors and their specific DNA binding sites [8, 9]. Nevertheless, establishing the complete structure of the complexes and protein interactions in a living cell, including the modulation of the interactions in different cellular states (temporal) and compartments (spatial), is a formidably complex problem.

Despite its limited size, the public release of the first set of proteomic data has produced an avalanche of theoretical studies on the organization of protein interaction networks, the identification of the basic control and interaction motifs, and the comparison to other non-biological networks [10-18].

At the formal level, the structure of metabolic and protein interaction networks has been fitted to power law distributions similar to those of many other biological and non-biological systems [19, 20]. As in these other systems, the implication is that the protein interaction networks are in a meta-stable situation (or critical state), which makes it impossible to predict the future development of the network and the fate of individual interactions. Considerable effort has also been put into the search for well-defined regions of the interaction network associated with defined biological properties, such as metabolic pathways with distinctive patterns of interactions [15, 21-24].

Here we review the sources of information available for protein interaction data, their organization in databases, and the potential of computational biology methods to complement the experimental information by inferring new interactions. Clearly, the availability of large-scale, well organized interaction data with the proper quality controls is essential for the success of theoretical studies of the properties of the molecular systems.

2 Large-Scale Studies of Protein Complexes: The Proteomes

2.1 Experimental Methods for the Large-Scale Detection of Protein Interactions

Several experimental methods are being applied for the large-scale detection of protein interactions. Some of these involve the implementation of standard techniques to study protein-protein interactions. One of the methods most often used is the yeast two-hybrid system (Y2H) [25, 26], based on the modular properties of the Gal4 protein of the yeast S. cerevisiae, as well as its modifications for application to membrane proteins [27]. A similar approach is based on beta-lactamase activity recovery [28]. Genome-wide studies involving variations of the Y2H protocol have been carried out in yeast, H. pylori, C. elegans and Drosophila [1-5, 29].

Ho et al., applied ultra-sensitive mass spectrometry to identify protein complexes in S. cerevisiae, covering 25% of the yeast proteome [30]. Tandem-affinity purification (TAP) and mass spectrometry was used by Gavin et al. to characterize multi-protein complexes in S. cerevisiae [5]. Yeast protein chips and microarrays have also been used to screen protein-protein interactions and protein-drug interactions [31]. Tong et al. applied a combination of computational prediction of interactions from phage-display ligand consensus sequences with large-scale two-hybrid physical interaction tests, to identify interaction partners of yeast SH3 domains [7].

Large-scale proteomics also implies some limitations, and the introduction of certain artefacts, such as those produced by the presence of promiscuous proteins with an artifactual preference to interact with many other proteins in Y2H assays or the over-representation of small proteins in complex purification strategies [32-36]. As in other high-throughput applications (e.g. DNA arrays), accuracy in the determination of individual properties is sacrificed in order to gain insight into the global properties of the system [37].

2.2 Extrapolating Experimental Information to Build Interaction Networks of Related Species

A number of attempts have been made to extrapolate the information on protein interactions obtained from model systems (S. cerevisae, C. elegans, H. pylori, D. melanogaster) to other genomes. In general, inferences have been made by assuming that orthologous sequences will participate in similar interactions. For example, the experimental interactions determined for H. pylori were extrapolated to E. coli by combining sequence similarity searches with a clustering strategy, based on interaction patterns and interaction domain information [38]. Lappe et al. developed an integration system to combine, compare and analyze interaction data from different sources and different organisms at a single level of abstraction [39]. Matthews et al. proposed a method to search for 'interologs' (potentially conserved interactions) in C. elegans using experimentally identified interacting protein partners of S. cerevisae [40-43].

These studies are very interesting, and certainly correspond to the most-simple assumption of conservation of interactions across different species. Nevertheless, the risk of extrapolating too far is considerable, even more so given that the principle of conservation of interactions across large evolutionary time has yet to be demonstrated and the combinatorial possibilities of protein domains complicates the situation significantly.

An interesting exploration of this problem has been published by Aloy and Russell [44] in which they calculated the degree of conservation of the interaction regions for pairs of proteins with different degrees of similarity. The conclusion of this study was that similar interaction sites can be predicted for proteins with sequence similarities as low as 30-40 %, even if the noise of the system is considerable. It is important to bear in mind that this study only implies that proteins that do interact tend to do so using similar regions, and not that similar proteins will necessarily interact (see below for a discussion of the problem of predicting interaction specificity).

3 Computational Methods for the Prediction of Interaction Partners

A number of computational methods have recently appeared that use sequence information to predict physical or functional interactions between proteins. Five of them are described in Box 1 [45, 46], although others are likely to appear.

The possibility of using sequence and evolutionary information to identify potential interaction partners brings additional opportunities to enrich the collection of interactions available for modeling studies. However, a definitive evaluation of these methods is still incomplete since the collections of experimental data on interacting proteins that can be used as controls have their own limitations (see the section on experimental methods above) and the overlap between the sets of predicted or experimental interactions is currently limited. Nevertheless, taking all these limitations into account, the increasing availability of genomic sequence information and the improvement of the methods still makes it likely that computational methods for predicting protein-protein interactions could achieve coverage and accuracies similar to those of the high-throughput experimental methods [47, 48].

Not surprisingly, interaction networks predicted by the various experimental and computational methods that are based on similar principles tend to have similar organizations [17].

3.1 Methods Based on Domain Composition

An alternative to the prediction of functional relationships between protein interactions is the study of the statistical association between proteins that share domains. The assumption in this case is that proteins that share a given domain will be functionally related by virtue of having this domain. Given the large number of multidomain proteins found in eukaryotes, it is easy to see that such a network will be highly complex and extremely dense. One approach attempts to elucidate which domains participate more often in protein interactions by considering the pairs of interacting yeast proteins recorded in the MIPS, MYGD and DIP databases, and the sequence domains included in the InterPro Database [49]. Another approach considers proteins as collections of conserved domains, where each domain is responsible for a specific interaction with another and a Markov chain Monte Carlo approach is used for the prediction of posterior probabilities of interaction between sets of proteins [50, 51].

3.2 Hybrid Methods Based on Sequence and Structure. Extrapolating from Interaction Partners to Interacting Regions

In order to manipulate molecular systems, by simulation or employing experimental methods, it is important to have information available not only about the general interaction networks, but also the details of the specific interaction at a molecular level. For example, the experimental manipulation of a signaling pathway with point mutations requires specific knowledge of the amino acid residues involved in the interactions. In other words, it is important to develop methods for the discovery of interacting regions, as a way of channeling the capacity of molecular biology and simulation techniques for the exploration of interaction networks.

Computational methods for the prediction of interaction partners based on genome comparisons (phylogenetic profiles, conservation of gene neighborhood and gene fusion detection; see inset) do not provide information about the molecular details; the predictions remain at the level of functional relationships between sequences. In contrast, the predictions of the other two methods described here (mirror-trees and in-silico-two-hybrid) can be translated at the residue level for particular proteins.

Structural biology is also contributing substantially to the study of protein complexes, and perhaps the most important milestone in this area has been the determination of the structure of the ribosome [52]. Generally speaking, information about the structure of proteins is an essential component of the study of biological systems. From this type of experimental information we have learned about stable and transient protein complexes, about their interaction surfaces, and, to some extent, about the specificity of their interactions. A very interesting new avenue has been recently open by Aloy et al. [53] with the combination of experimental structure, protein models, and biochemical information to build the structure of new complexes whose general shape was solved by systematic electron microscopy studies of protein complexes purified by TAPs-MS.

From a computational point of view, major advances have been in the development of programs for the prediction of the structure of protein complexes (docking programs, [54, 55]), and a number of sequence-related analysis systems for the prediction of potential interaction regions.[56] In the near future, interesting progress is expected in the prediction of interaction regions by combining structural and sequence information.

Beyond the prediction of complex structure for interacting proteins of known structure, we still have to face the problem of distinguishing between potentially interacting proteins, e.g. all the pairs of proteins belonging to two protein families, versus the few protein pairs that are actually interacting. The specificity of those interactions is essential for the function of cellular systems in which members of the same protein family, using the same basic architecture, are able to trigger different signaling pathways. It is conceivable that a combination of protein modeling techniques and sequence information analysis will contribute to the search for the molecular basis of protein-protein recognition specificity. A few methods have been developed to this end. These methods make use of residue pair potentials obtained from interacting surfaces of known complexes. The information is then used to assess the extent to which the homologues of two interacting proteins of known structure will interact [57, 58]. Lu et al. have extended their protein structure prediction method to the prediction of the stability of protein complexes (Multiprospector). In this case, all combinations of protein sequences are tested for their compatibility in the framework of known protein complexes. The rationale is that proteins that will naturally form complexes will be more stable when associated with their partners than in isolation [59, 60]. The application of this method to complete genomes shows an impressive capacity for predicting potential interactions and an accuracy similar to other prediction methods [61]. Our group has studied the problem of molecular specificity in various systems in which computational predictions have allowed us to manipulate the molecular basis of specific recognition in protein interactions [62-66]. However, in some cases accurate prediction of interactions is not possible due to the complexity of the conformational changes in the interaction surfaces.

4 Organization of the Information on Interactions in Specific Databases

In recent years, high-throughput methods have made molecular biology a data-intensive discipline. These data have to be stored in a structured way for data retrieval and analysis. A number of protein interaction results have been stored in this manner and made accessible via web services (see Table 1). All of these projects are still in an initial phase, which explains the current lack of differentiating characteristics that in the long run will determine their utility and survival in competition with other initiatives.

The Human Proteome Organization (HUPO) has launched an effort to establish standards for interaction databases that would be acceptable for all existing projects. These standards contain the minimum sufficient information to describe interactions, with the intention of facilitating information exchange between interaction databases. The consortium behind these initiatives has already designed the basic layer (XML) for the exchange, and a technical vocabulary for the description of the many experimental and theoretical techniques that produce data on protein interactions. Similar initiatives are taking place in related areas such as metabolic pathway databases[67]. The main databases of this kind have been running for years EMP [68, 69], WIT [70], KEGG [71], EcoCyc [72], and new ones are still appearing (aMAZE) [73, 74]. They are designed for storing information on enzymes, biochemical reactions and small molecules, and in some cases, the corresponding kinetic parameters. There are initiatives to create compatible standards between metabolic databases (see for example BioPAX-http://www.biopax.org/), which in the future may include protein interaction databases.

Alongside the data standardization structure, other projects have focused on a solution to another major database problem: data distribution. Many institutes and labs have relevant scientific information that is accessible through static web interfaces that are rarely visited. New technologies are now arising that are able to make all these data accessible through a single interface that can retrieve the information from its main source. An example of this technology is the PLANET project (see http://eu-plant-genome.net), where different data repositories are being made accessible through a single interface thanks to BioMoby technology [75].

The internet has offered a fast channel for information interchange. This has been particularly the case for the development of computational biology. Massive data exchange operations have made data reliability a major concern. Error propagation has proved to be a concern in areas with database annotation, making the link between annotation and the underlying experimental information an important issue. This need has increased the efforts in text mining research to recover the links between protein interaction databases and the corresponding sentences in the literature. During the last few years the technology in this area has developed rapidly [76-79]. Nevertheless, key

Table 1. Main databases on protein-protein interactions

Database	Site and Description
DIP [80-82]	Stores experimentally determined interactions between proteins. Currently, it includes 18,488 interactions for 7134 proteins in 104 organisms. http://dip.doe-mbi.ucla.edu/
MINT [98]	Designed to store functional interactions between biological molecules (proteins, RNA, DNA). It is now focusing on experimentally-verified direct and indirect protein-protein interactions. http://cbm.bio.uniroma2.it/mint/
BIND [99]	Contains full descriptions of interactions, molecular complexes and pathways http://www.bind.ca/
MIPS [100]	Large collection of diverse types of interactions. Commonly used as equivalent to 'hand-curated' sets of interactions. http://www.mips.biochem.mpg.de/
PathCalling Yeast Interaction Database [1]	Identifies protein-protein interactions on a genome-wide scale for functional assignment and drug target discovery http://portal.curagen.com/extpc/com.curagen.portal.servlet.Yeast
The GRID [101]	A database of genetic and physical interactions that contains interaction data from several sources, including MIPS and BIND http://biodata.mshri.on.ca/grid/servlet/Index
IntAct [67]	The project (funded by a European Commission grant, TEMBLOR) aims to represent and annotate protein-protein interactions, and to develop a public database of experimentally identified and predicted interactions. The database structure is designed to incorporate experimentally determined and predicted interactions, with special care in tracing the origin of the information. The interactions will be directly linked to original sentences in the literature describing them, for which text mining technology will be used. http//www.ebi.ac.uk/intact
STRING [46]	STRING is a database of known and predicted protein-protein interactions. http://string.embl.de/newstring_cgi/show_input_page.pl
HPID [42]	The human protein interaction database. Contains human protein interactions inferred by homology searches against experimental interaction data. http://www.hpid.org/
Prolinks [102]	A database of protein functional linkages derived from coevolution. Contains functional links predicted by several methods. http://169.232.137.207/cgi-dev/functionator/pronav
Predictome [103]	A database of putative functional links between proteins. Contains functional links establish by a variety of techniques, both experimental and computational. http://predictome.bu.edu/

problems remain in the field, such as the identification of protein and gene names. For example, in 2001 it was possible to link only 30% of the DIP database entries to the literature [80-82]. Only 20% of the missing links were explained by inaccuracies in the text mining system; the remaining 80% were produced because the protein names used in the database were not found in any of the available Medline entries, or because there was no information about the interactions in the literature. In the current status of the technology, the number of synonyms has grown, as well as the number of technical possibilities for detecting interactions[79]. Thus, this technology is maturing fast and may soon be able to facilitate the tasks of annotating databases, and to keep direct pointers between the interactions and the literature. (Very recently a collaborative effort has been launched to assess technologies in this area, see http://www.pdg.cnb.uam.es/BioLink).

5 Concluding Remarks

Genomic sequencing, proteome characterization and structural genomics projects are providing a wealth of information about genes and proteins. Proteomics now offers the possibility of entering a new dimension of understanding, directly related to the organization of the basic components in protein networks and complexes. The experimental and computational approaches published in the last five years have provided the first wide ranging view of the properties, organization, evolution and complexity of protein interaction networks. Computational Biology is contributing to this collective effort with, firstly, new methods to identify protein interaction partners on a large scale, and secondly with new approaches able to provide detailed descriptions, and associated predictions, of protein interaction sites.

It is important to bear in mind that the characterization of protein interaction networks is only one initial step towards the understanding of cellular systems; a step for which high-throughput proteomics, bioinformatics and computational biology are inherently associated with the success of Computational Systems Biology.

Acknowledgements

This study was funded by the EC project TEMBLOR (EU grant QLRT-2001-00015). MJG is recipient of an I3P contract from the Spanish Research Council (CSIC).

Boxes

Box 1. Computational Methods for the Prediction of Interaction Partners

Phylogenetic profiles. This method is based on the identification of genes that have the same pattern of presence/absence in a number of genomes. A group of genes with the same phylogenetic profile is assumed to encode proteins that are functionally related (for example, they may be part of the same metabolic pathway) and that may or may not interact physically. The drawback of the method is that it can only be applied to complete genomes [83, 84].

Conservation of gene neighborhood. Especially in prokaryotes, the neighborhood of a gene has a tendency to be conserved, both in terms of identity and order of the genes. This is partly related to the fact that genes in prokaryotes are often organized in operons. Operons contain genes that need to be expressed in a coordinated fashion, usually because they are involved in related functions. The observed relationship between chromosome proximity and function [85] has been exploited to predict gene interactions, both in the physical and in the functional sense [86, 87].

Gene fusion. Two proteins, or protein domains, encoded by different genes are assumed to interact physically, or at least functionally, if in some species they are coded by a single gene, presumably originating from a gene fusion event [88, 89]. It has been shown that fusion events are particularly common in metabolic proteins [90].

Mirror trees. Interacting proteins are expected to co-evolve. Therefore, the corresponding phylogenetic trees should be more similar than those of non-interacting proteins. The first qualitative assessments of this concept were performed with the pairs composed of the insulin and their receptors [91], and dockerins and cohexins [92]. Later, linear correlation between the distance matrices used to construct the trees was proposed to measure tree similarity [93] and the approach was applied to large data sets [94]. Recently, a method based on this concept has been developed for predicting interaction specificity [95].

In silico two-hybrid. The co-evolution of interacting proteins can be studied by analysis of mutations in one of the partners that compensate mutations in the other. The detection of correlated mutations has been used to predict the tendency of pairs of residues to be in physical proximity [96]. This method has been applied to large data sets of proteins and domains [97].

References

1. Uetz, P., et al., A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. Nature, 2000. 403(6770): p. 623-7.
2. Giot, L., et al., A protein interaction map of Drosophila melanogaster. Science, 2003. 302(5651): p. 1727-36.
3. Rain, J.C., et al., The protein-protein interaction map of Helicobacter pylori. Nature, 2001. 409(6817): p. 211-5.
4. Li, S., et al., A map of the interactome network of the metazoan C. elegans. Science, 2004. 303(5657): p. 540-3.
5. Gavin, A.C., et al., Functional organization of the yeast proteome by systematic analysis of protein complexes. Nature, 2002. 415(6868): p. 141-7.
6. Landgraf, C., et al., Protein interaction networks by proteome Peptide scanning. PLoS Biol, 2004. 2(1): p. E14.
7. Tong, A.H., et al., A combined experimental and computational strategy to define protein interaction networks for peptide recognition modules. Science, 2002. 295(5553): p. 321-4.
8. Ren, B., et al., Genome-wide location and function of DNA binding proteins. Science, 2000. 290(5500): p. 2306-9.
9. Lee, T.I., et al., Transcriptional regulatory networks in Saccharomyces cerevisiae. Science, 2002. 298(5594): p. 799-804.
10. Milo, R., et al., Network motifs: simple building blocks of complex networks. Science, 2002. 298(5594): p. 824-7.

11. Fraser, H.B., et al., Evolutionary rate in the protein interaction network. Science, 2002. 296(5568): p. 750-2.
12. Jeong, H., et al., The large-scale organization of metabolic networks. Nature, 2000. 407(6804): p. 651-4.
13. Jeong, H., et al., Lethality and centrality in protein networks. Nature, 2001. 411(6833): p. 41-2.
14. Maslov, S. and K. Sneppen, Specificity and stability in topology of protein networks. Science, 2002. 296(5569): p. 910-3.
15. Ravasz, E., et al., Hierarchical organization of modularity in metabolic networks. Science, 2002. 297(5586): p. 1551-5.
16. Rives, A.W. and T. Galitski, Modular organization of cellular networks. Proc Natl Acad Sci U S A, 2003. 100(3): p. 1128-33.
17. Hoffmann, R. and A. Valencia, Protein interaction: same network, different hubs. Trends Genet, 2003. 19(12): p. 681-3.
18. Milo, R., et al., Superfamilies of evolved and designed networks. Science, 2004. 303(5663): p. 1538-42.
19. Amaral, L.A., et al., Classes of small-world networks. Proc Natl Acad Sci U S A, 2000. 97(21): p. 11149-52.
20. Barabasi, A.L. and R. Albert, Emergence of scaling in random networks. Science, 1999. 286(5439): p. 509-12.
21. Snel, B., P. Bork, and M.A. Huynen, The identification of functional modules from the genomic association of genes. Proc Natl Acad Sci U S A, 2002. 99(9): p. 5890-5.
22. von Mering, C., et al., Genome evolution reveals biochemical networks and functional modules. Proc Natl Acad Sci U S A, 2003. 100(26): p. 15428-33.
23. Wuchty, S., Z.N. Oltvai, and A.L. Barabasi, Evolutionary conservation of motif constituents in the yeast protein interaction network. Nat Genet, 2003. 35(2): p. 176-9.
24. Yook, S.H., Z.N. Oltvai, and A.L. Barabasi, Functional and topological characterization of protein interaction networks. Proteomics, 2004. 4(4): p. 928-42.
25. Fields, S. and O. Song, A novel genetic system to detect protein-protein interactions. Nature, 1989. 340(6230): p. 245-6.
26. Phizicky, E., et al., Protein analysis on a proteomic scale. Nature, 2003. 422(6928): p. 208-15.
27. Stagljar, I. and S. Fields, Analysis of membrane protein interactions using yeast-based technologies. Trends Biochem Sci, 2002. 27(11): p. 559-63.
28. Wehrman, T., et al., Protein-protein interactions monitored in mammalian cells via complementation of beta -lactamase enzyme fragments. Proc Natl Acad Sci U S A, 2002. 99(6): p. 3469-74.
29. Ito, T., et al., Toward a protein-protein interaction map of the budding yeast: A comprehensive system to examine two-hybrid interactions in all possible combinations between the yeast proteins. Proc Natl Acad Sci U S A, 2000. 97(3): p. 1143-7.
30. Ho, Y., et al., Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry. Nature, 2002. 415(6868): p. 180-3.
31. Zhu, H., et al., Global analysis of protein activities using proteome chips. Science, 2001. 293(5537): p. 2101-5.
32. Aloy, P. and R.B. Russell, Potential artefacts in protein-interaction networks. FEBS Lett, 2002. 530(1-3): p. 253-4.
33. Lakey, J.H. and E.M. Raggett, Measuring protein-protein interactions. Curr Opin Struct Biol, 1998. 8(1): p. 119-23.
34. Legrain, P., J. Wojcik, and J.M. Gauthier, Protein--protein interaction maps: a lead towards cellular functions. Trends Genet, 2001. 17(6): p. 346-52.
35. Bader, G.D. and C.W. Hogue, Analyzing yeast protein-protein interaction data obtained from different sources. Nat Biotechnol, 2002. 20(10): p. 991-7.

36. von Mering, C., et al., Comparative assessment of large-scale data sets of protein-protein interactions. Nature, 2002. 417(6887): p. 399-403.
37. Grunenfelder, B. and E.A. Winzeler, Treasures and traps in genome-wide data sets: case examples from yeast. Nat Rev Genet, 2002. 3(9): p. 653-61.
38. Wojcik, J. and V. Schachter, Protein-protein interaction map inference using interacting domain profile pairs. Bioinformatics, 2001. 17 Suppl 1: p. S296-305.
39. Lappe, M., et al., Generating protein interaction maps from incomplete data: application to fold assignment. Bioinformatics, 2001. 17 Suppl 1: p. S149-56.
40. Matthews, L.R., et al., Identification of potential interaction networks using sequence-based searches for conserved protein-protein interactions or "interologs". Genome Res, 2001. 11(12): p. 2120-6.
41. Goffard, N., et al., IPPRED: server for proteins interactions inference. Bioinformatics, 2003. 19(7): p. 903-4.
42. Han, K., et al., HPID: the human protein interaction database{paragraph}. Bioinformatics, 2004.
43. de la Torre, V., et al., iPPI: a web server for protein-protein interactions inference. submited.
44. Aloy, P., et al., The relationship between sequence and interaction divergence in proteins. J Mol Biol, 2003. 332(5): p. 989-98.
45. Valencia, A. and F. Pazos, Computational methods for the prediction of protein interactions. Curr Opin Struct Biol, 2002. 12(3): p. 368-73.
46. von Mering, C., et al., STRING: a database of predicted functional associations between proteins. Nucleic Acids Res, 2003. 31(1): p. 258-61.
47. Huynen, M.A., et al., Function prediction and protein networks. Curr Opin Cell Biol, 2003. 15(2): p. 191-8.
48. Deng, M., F. Sun, and T. Chen, Assessment of the reliability of protein-protein interactions and protein function prediction. Pac Symp Biocomput, 2003: p. 140-51.
49. Sprinzak, E. and H. Margalit, Correlated sequence-signatures as markers of protein-protein interaction. J Mol Biol, 2001. 311(4): p. 681-92.
50. Gomez, S.M., S.H. Lo, and A. Rzhetsky, Probabilistic prediction of unknown metabolic and signal-transduction networks. Genetics, 2001. 159(3): p. 1291-8.
51. Gomez, S.M. and A. Rzhetsky, Towards the prediction of complete protein--protein interaction networks. Pac Symp Biocomput, 2002: p. 413-24.
52. Yusupov, M.M., et al., Crystal structure of the ribosome at 5.5 A resolution. Science, 2001. 292(5518): p. 883-96.
53. Aloy, P., et al., Structure-based assembly of protein complexes in yeast. Science, 2004. 303(5666): p. 2026-9.
54. Smith, G.R. and M.J. Sternberg, Prediction of protein-protein interactions by docking methods. Curr Opin Struct Biol, 2002. 12(1): p. 28-35.
55. Camacho, C.J. and S. Vajda, Protein-protein association kinetics and protein docking. Curr Opin Struct Biol, 2002. 12(1): p. 36-40.
56. Garcia-Ranea, J.A. and A. Valencia, Distribution and functional diversification of the ras superfamily in Saccharomyces cerevisiae. FEBS Lett, 1998. 434(3): p. 219-25.
57. Aloy, P. and R.B. Russell, Interrogating protein interaction networks through structural biology. Proc Natl Acad Sci U S A, 2002. 99(9): p. 5896-901.
58. Aloy, P. and R.B. Russell, InterPreTS: protein interaction prediction through tertiary structure. Bioinformatics, 2003. 19(1): p. 161-2.
59. Lu, H., L. Lu, and J. Skolnick, Development of unified statistical potentials describing protein-protein interactions. Biophys J, 2003. 84(3): p. 1895-901.
60. Lu, L., H. Lu, and J. Skolnick, MULTIPROSPECTOR: an algorithm for the prediction of protein-protein interactions by multimeric threading. Proteins, 2002. 49(3): p. 350-64.

61. Lu, L., et al., Multimeric threading-based prediction of protein-protein interactions on a genomic scale: application to the Saccharomyces cerevisiae proteome. Genome Res, 2003. 13(6A): p. 1146-54.
62. Bauer, B., et al., Effector recognition by the small GTP-binding proteins Ras and Ral. J Biol Chem, 1999. 274(25): p. 17763-70.
63. Hernanz-Falcon, P., et al., Identification of amino acid residues crucial for chemokine receptor dimerization. Nat Immunol, 2004. 5(2): p. 216-23.
64. Morillas, M., et al., Identification of conserved amino acid residues in rat liver carnitine palmitoyltransferase I critical for malonyl-CoA inhibition. Mutation of methionine 593 abolishes malonyl-CoA inhibition. J Biol Chem, 2003. 278(11): p. 9058-63.
65. Azuma, Y., et al., Model of the ran-RCC1 interaction using biochemical and docking experiments. J Mol Biol, 1999. 289(4): p. 1119-30.
66. Renault, L., et al., Structural basis for guanine nucleotide exchange on Ran by the regulator of chromosome condensation (RCC1). Cell, 2001. 105(2): p. 245-55.
67. Hermjakob, H., et al., IntAct: an open source molecular interaction database. Nucleic Acids Res, 2004. 32 Database issue: p. D452-5.
68. Selkov, E., Jr., et al., MPW: the Metabolic Pathways Database. Nucleic Acids Res, 1998. 26(1): p. 43-5.
69. Selkov, E., et al., The metabolic pathway collection from EMP: the enzymes and metabolic pathways database. Nucleic Acids Res, 1996. 24(1): p. 26-8.
70. Overbeek, R., et al., WIT: integrated system for high-throughput genome sequence analysis and metabolic reconstruction. Nucleic Acids Res, 2000. 28(1): p. 123-5.
71. Kanehisa, M., et al., The KEGG databases at GenomeNet. Nucleic Acids Res, 2002. 30(1): p. 42-6.
72. Karp, P.D., et al., The EcoCyc Database. Nucleic Acids Res, 2002. 30(1): p. 56-8.
73. van Helden, J., et al., Representing and analysing molecular and cellular function using the computer. Biol Chem, 2000. 381(9-10): p. 921-35.
74. Joshi-Tope, G., et al., The Genome Knowledgebase: A Resource for Biologists and Bioinformaticists. CSHL Symposium 2003, 2003.
75. Wilkinson, M.D. and M. Links, BioMOBY: an open source biological web services proposal. Brief Bioinform, 2002. 3(4): p. 331-41.
76. Andrade, M.A., et al., Classification of protein families and detection of the determinant residues with an improved self-organizing map. Biol Cybern, 1997. 76(6): p. 441-50.
77. Blaschke, C., et al., Automatic extraction of biological information from scientific text: protein-protein interactions. Proc Int Conf Intell Syst Mol Biol, 1999: p. 60-7.
78. Friedman, C., et al., GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. Bioinformatics, 2001. 17 Suppl 1: p. S74-82.
79. Blaschke, C., L. Hirschman, and A. Valencia, Information extraction in molecular biology. Brief Bioinform, 2002. 3(2): p. 154-65.
80. Xenarios, I., et al., DIP: the database of interacting proteins. Nucleic Acids Res, 2000. 28(1): p. 289-91.
81. Xenarios, I., et al., DIP: The Database of Interacting Proteins: 2001 update. Nucleic Acids Res, 2001. 29(1): p. 239-41.
82. Xenarios, I., et al., DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions. Nucleic Acids Res, 2002. 30(1): p. 303-5.
83. Gaasterland, T. and M.A. Ragan, Microbial genescapes: phyletic and functional patterns of ORF distribution among prokaryotes. Microb Comp Genomics, 1998. 3(4): p. 199-217.
84. Pellegrini, M., et al., Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. Proc Natl Acad Sci U S A, 1999. 96(8): p. 4285-8.
85. Tamames, J., et al., Conserved clusters of functionally related genes in two bacterial genomes. J Mol Evol, 1997. 44(1): p. 66-73.

86. Dandekar, T., et al., Conservation of gene order: a fingerprint of proteins that physically interact. Trends Biochem Sci, 1998. 23(9): p. 324-8.
87. Overbeek, R., et al., Use of contiguity on the chromosome to predict functional coupling. In Silico Biol, 1999. 1(2): p. 93-108.
88. Enright, A.J., et al., Protein interaction maps for complete genomes based on gene fusion events. Nature, 1999. 402(6757): p. 86-90.
89. Marcotte, E.M., et al., Detecting protein function and protein-protein interactions from genome sequences. Science, 1999. 285(5428): p. 751-3.
90. Tsoka, S. and C.A. Ouzounis, Prediction of protein interactions: metabolic enzymes are frequently involved in gene fusion. Nat Genet, 2000. 26(2): p. 141-2.
91. Fryxell, K.J., The coevolution of gene family trees. Trends Genet, 1996. 12(9): p. 364-9.
92. Pages, S., et al., Species-specificity of the cohesin-dockerin interaction between Clostridium thermocellum and Clostridium cellulolyticum: prediction of specificity determinants of the dockerin domain. Proteins, 1997. 29(4): p. 517-27.
93. Goh, C.S., et al., Co-evolution of proteins with their interaction partners. J Mol Biol, 2000. 299(2): p. 283-93.
94. Pazos, F. and A. Valencia, Similarity of phylogenetic trees as indicator of protein-protein interaction. Protein Eng, 2001. 14(9): p. 609-14.
95. Ramani, A.K. and E.M. Marcotte, Exploiting the co-evolution of interacting proteins to discover interaction specificity. J Mol Biol, 2003. 327(1): p. 273-84.
96. Pazos, F., et al., Correlated mutations contain information about protein-protein interaction. J Mol Biol, 1997. 271(4): p. 511-23.
97. Pazos, F. and A. Valencia, In silico two-hybrid system for the selection of physically interacting protein pairs. Proteins, 2002. 47(2): p. 219-27.
98. Zanzoni, A., et al., MINT: a Molecular INTeraction database. FEBS Lett, 2002. 513(1): p. 135-40.
99. Bader, G.D., D. Betel, and C.W. Hogue, BIND: the Biomolecular Interaction Network Database. Nucleic Acids Res, 2003. 31(1): p. 248-50.
100. Mewes, H.W., et al., MIPS: a database for genomes and protein sequences. Nucleic Acids Res, 2002. 30(1): p. 31-4.
101. Breitkreutz, B.J., C. Stark, and M. Tyers, The GRID: the General Repository for Interaction Datasets. Genome Biol, 2003. 4(3): p. R23.
102. Bowers, P.M., et al., Prolinks: a database of protein functional linkages derived from coevolution. Genome Biol, 2004. 5(5): p. R35.
103. Mellor, J.C., et al., Predictome: a database of putative functional links between proteins. Nucleic Acids Res, 2002. 30(1): p. 306-9.

Early Systems Biology and Prebiotic Networks

Barak Shenhav*, Ariel Solomon*, Doron Lancet, and Ran Kafri

Department of Molecular Genetics and the Crown Human Genome Center,
the Weizmann Institute of Science, Rehovot 76100, Israel
{barak.shenhav, ari-el.solomon, ron.kafri,
doron.lancet}@weizmann.ac.il

Abstract. Systems Biology constitutes tools and approaches aimed at deciphering complex biological entities. It is assumed that such complexity arose gradually, beginning from a few relatively simple molecules at life's inception, and culminating with the emergence of composite multicellular organisms billions of years later. The main point of the present paper is that very early in the evolution of life, molecular ensembles with high complexity may have arisen, which are best described and analyzed by the tools of Systems Biology. We show that modeled prebiotic mutually catalytic pathways have network attributes similar to those of present-day living cells. This includes network motifs and robustness attributes. We point out that early networks are weighted (graded), but that using a cutoff formalism one may probe their degree distribution and show that it approximate that of a random network. A question is then posed regarding the potential evolutionary mechanisms that may have led to the emergence of scale-free networks in modern cells.

1 Prebiotic Molecular Networks

Most researchers admit that somewhere along the line towards the appearance of self-reproducing protocells, a web of interacting molecules must have been at work. Yet, investigators are divided on a crucial question related to the first reproducing entities. One set of scenarios claims that life began with a single molecular replicator, e.g. an RNA-like biopolymer [1-3]. It is further assumed that complex molecular networks came much later, and were genetically instructed by the replicating polymers. The second set of scenarios asserts that early replicating entities must have constituted complex molecular networks right from the outset. The latter view claims that the emergence of single molecules, whose inner works allowed them to instruct the synthesis of their own copies, are extremely unlikely under prebiotic conditions. It is claimed that the spontaneous accretion of specific mixtures or assemblies of simple organic molecules, capable of self-reproduction, is more probable. Furthermore, it is suggested that the capacity of such molecular assemblies to undergo a replication or reproduction-like process is a direct consequence of certain network properties parallel to those that allow present-day cells to divide and beget progeny. If the "network-first" scenario is right, then a better understanding of network properties within contemporary living cells should be a crucial tool for understanding prebiotic evolution.

* These authors contributed equally to this article.

C. Priami et al. (Eds.): Transactions on Computational Systems Biology, LNBI 3380, pp. 14 – 27, 2005.

The early Systems Biology view has a promise for merging the two seemingly conflicting scenarios for prebiotic evolution. This has been recently presented by Luisi [4] as follows: “Another new wind in our field comes, in my opinion, from the development of system biology – biology seen in terms of system theory, namely the whole biological system studied in its entire complexity: proteomics, genomics, networks and non linear systems, and so on. This has brought about a revival of theoretical and experimental studies on chemical complexity, like self-organization, emergent properties, autocatalysis – concepts that were already with us, that however have acquired nowadays a new importance.”

2 Binary and Weighted Networks

Until 1960, regular networks, such as lattices, were the typical structured mathematical entities studied in the realm of graph theory. One such regular network, a hyper-cube, underlies the dynamics described by the classical quasi-species model for early evolution [5]. Later, Paul Erdös importantly introduced random networks, and studied their mathematical properties. In this type of networks the nodes are connected in a haphazard fashion to every other, each edge having a probability p to appear. Such network inspired Kauffmann’s mutual catalysis model for the origin of life (Fig. 1B) [6].

Despite their seeming generality, it was more recently shown that random networks do not correspond properly to those that often appear in biological systems. In particular, random networks have a binomial or Poisson distribution of node degrees, typified by a most probable value, while many biological networks are scale-free, and their degree distribution follows a power law (Fig.1 and 4) [7-13]. The scale-free nature of present-day protein networks (to address one example) stems from the fact that a few proteins, belonging to certain families, are capable of interacting with a large number of partners [14].

Considerable Systems Biology research has been performed in recent years on the properties of biological networks [12, 15-18]. Much of this effort pertains to unweighted - binary networks, in which a specific node is either connected or not connected to another. Classical network attributes such as degree distribution, mean path length and clustering coefficient, are based on counting these binary connections. However, in many instances, not only restricted to biology, every two nodes in a network are connected in a graded or weighted fashion. This happens when a continuous measure such as affinity or catalytic rate governs the interactions among nodes, as is the case in the Graded Autocatalysis Replication Domain (GARD) model for early evolution, described below. It is possible to explore ways in which to convert weighted networks into binary ones, so as to afford analyses with existing network tools.

3 Specificity Attributes Within Networks

In present-day cellular networks, connectivity is rather sparse. Thus, a protein interaction network may have many thousands of nodes, but each node is connected to only a few or at most few dozen others (Fig. 1C, D). This state of affairs could be due to the long evolutionary process, in which proteins have emerged as highly specific and selective recognition devices. However, it would be surmised that in the early stages of

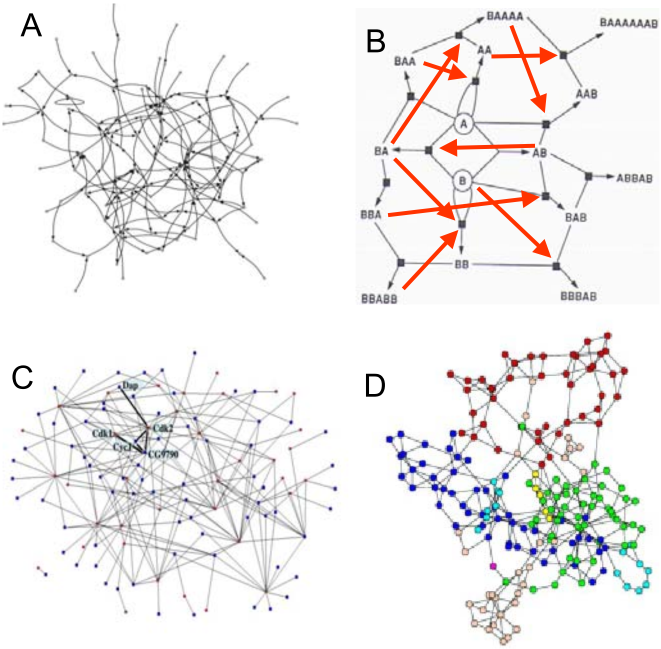


Fig. 1. Prebiotic and contemporary biological networks

A. The canonic composome (cf. figure 3) of a Graded Autocatalytic Replication Domain (GARD) system (cf. figure 2) with a molecular repertoire, N_G=1000. The total molecular count in the system, N, is taken to be 800. Only molecular species with concentration $\frac{n_i}{N} > \frac{1}{800}$ are shown. An edge is shown if the catalysis exerted on joining of molecular species A_i by molecular species A_j (β_{ij}) enhances the reaction by at least 100 folds.

B. Schematic illustration of an autocatalytic set as proposed by Kauffman (modified from [6]). Strings composed of A's and B's represent molecular species (oligomers) with different length and sequence. Black lines joined by a black square represent a ligation (concatenation) reaction. Wide arrows indicate catalysis on such ligation reaction by a member of the set. According to the model, if one assumes a fixed probability, *p*, for each molecular species to catalyze each reaction and if the number of molecular species is sufficiently large, then an autocatalytic set will emerge. In such set, the formation of every molecular species, except for the basic building blocks (A and B) will be catalyzed by at least one member of the set. Kaufmann pre-biotic network is binary by definition, as no gradation of catalytic potencies is assumed. Its degree distribution is clearly binomial, as every edge has the same probability p to appear.

C. Protein interaction map (PIM) generated by using ~100 known or suspected cell cycle regulators, including Cdk1 and Cdk2, in high throughput screens to detect possible interactions with ~13,000 Drosophila proteins [36]. The typical hubs that typify contemporary biological network may be seen.

D. Metabolic network of *E. coli,* where each node corresponds to a metabolite, and edges represent biochemical reactions. Figure is from [37].

molecular evolution, recognition was much more promiscuous. In particular, in prebiotic scenarios as proposed [6, 19-23] biopolymers such as folded proteins and RNAs may have not yet emerged, and smaller, simpler organic molecules may have played pivotal roles in information storage and catalysis. Under such circumstances, molecular species may have had a much larger number of interacting partners, and the corresponding network would have very high average network degree values. Thus, early networks (Fig. 1A,B), whose properties we have attempted to capture in the GARD model, are markedly different from their more modern counterparts. In such early systems, it is likely that practically all interacting pairs would have affinities in the range of what would presently be considered non-specific binding. What nowadays constitute the background noise may have been the only existing interactions at the inception of life.

4 GARD Dynamics and Composomes

We have, in the last decade, explored a defined formalism for describing and simulating the behavior of early systems with mutual catalytic interactions among simple molecules under prebiotic conditions [24-27]. Accordingly, a formula was proposed that defined the probability for a particular value of catalytic potency for a randomly selected molecular pair. This formalism, which is based on a Receptor Affinity Distribution model [28-30], resulted in the definition of a matrix, β, specifying a non-zero interaction for every pair of molecules. A reasonable way to regard such matrix is that it represents a fully connected weighted network of interactions (Fig. 3A).

The Graded autocatalysis Replication Domain (GARD) model depicts the kinetic behavior of such networks. Along the time scale, a dynamic process unfolds, whereby inside a molecular assembly of a finite size, certain molecular species prevail and others are selected against. This results in the emergence of different "composome", quasi-stationary states of the system, with different biased compositions. Fig. 3A shows a network that characterizes a complete β matrix. Despite the small number of components (N_G=300), this network is rather densely connected, because it arises from a system in which nominally every component is connected to every other (visualized with an edge cutoff of β=100).

The GARD model provides a detailed dynamic description of time-dependent changes in the concentrations of different molecular species within an assembly. In terms of concrete chemistry, it is assumed that the molecules within a GARD assembly are amphiphiles (lipid-like), held together within a micelle-like structure by hydrophobic forces. A GARD assembly, similar to a lipid bilayer, is a fluid structure, which can exchange molecules with the environment. Furthermore, rapid diffusion of molecules within it facilitates their mutual rate-enhancing interactions. Computer simulations of GARD equations allow one to view a time series of concentrations or compositions – a GARD trace. At each time point, the count of different molecular species is recorded, resulting in a trace of samples (see Figs. 2, 6B).

GARD usually considers a collection of N_G different molecules, i.e. a molecular repertoire of size N_G. In GARD, every component of an assembly may catalyze the entry/exit or formation/breakdown of every other component. Thus, at every time point, GARD by definition constitutes a mutually catalytic network. The effectiveness of such network, that is the capacity to sustain homeostasis (Fig. 2), varies, and depends on the exact composition, and on the web of interactions that prevails among the components. Composomes are specific compositions that last over many growth-

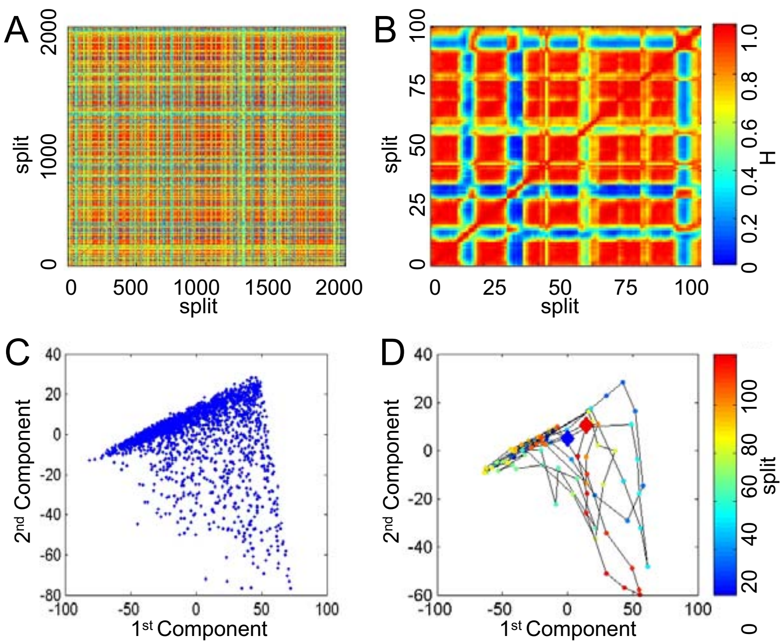


Fig. 2. The dynamics of a GARD assembly [25]. The composition of a molecular assembly is represented in the GARD model by a vector, **n,** with components, n_i, for every one of the N_G different molecular species in the system. Each n_i indicates the molecular counts of molecular species A_i in the assembly. A crucial assumption in the model is that every reaction would be catalyzed, to some extent, by each of the molecules in the assembly. GARD presumes that molecular assemblies undergo occasional fissions that yield smaller assemblies. This is modeled by having N_0 molecules randomly removed from the assembly once its size ($N=\sum n_i$) exceeds a threshold of $2N_0$. A GARD trace is a series of compositional vectors as a function times

A. GARD "carpet" showing an autocorrelation matrix of a trace containing 2,000 splits. The similarity between two compositions, $\mathbf{n_1}$ and $\mathbf{n_2}$, is measured by the scalar product:

$$H(\mathbf{n_1},\mathbf{n_2})=\frac{\mathbf{n_1}\cdot\mathbf{n_2}}{|\mathbf{n_1}|\cdot|\mathbf{n_2}|}$$

with a color scale (right of B) that has red for H = 1.0 (high similarity) and blue for H = 0 (indicates no similarity).

B. A partial 'carpet' for the arbitrarily sampled splits 1,700-1,800 in the trace shown in A. Red squares indicate composomes [25]. Off-diagonal red squares indicate that composition of composomes tends to be repeated.

C. The projection of all 2000 NG-dimensional compositions in the trace (shown in A) displayed in a two dimensional plane defined by the first and second components in a Principle Components Analysis (PCA). The samples form a triangle in the plane, which is not occupied uniformly. The heavily occupied edge corresponds to samples in the prominent composome and the opposing vertex to a lesser frequent composome.

D. The projection of the samples illustrated in B to the plane defined in C. Each composition is colored according to its time of appearance (color bar on right). The blue diamond corresponds to the first sample in the sub-trace and the red diamond to the last. Consecutive samples in time are connected by a line. The analysis show examples of paths taken from the dominating composome to the other one and back, e.g. samples 88-95 (red).

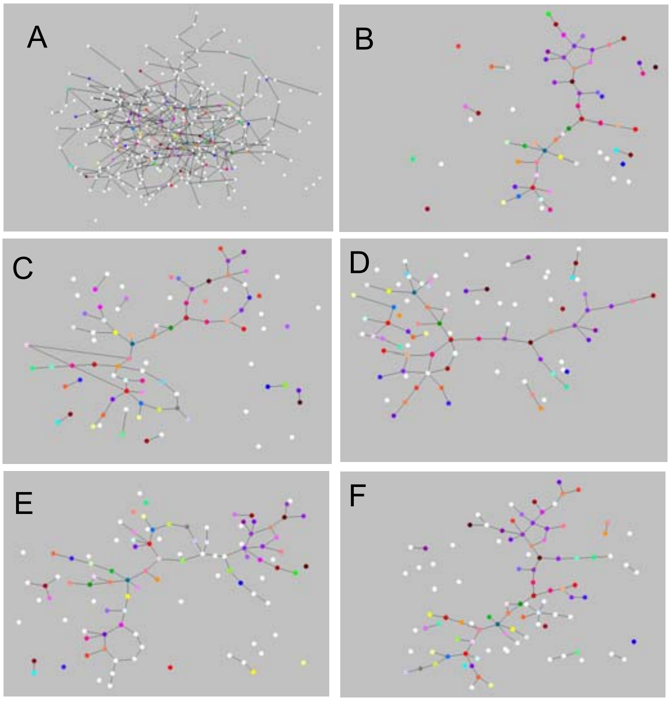


Fig. 3. Networks in the Graded Autocatalytic Replication Domain (GARD) model. An example of networks as found in a GARD system with N_G=300, where each node corresponds to a different molecular species (monomer) and edges are catalytic potencies. A cutoff was used whereby edges are shown only for β values that exceed a threshold of 100. White nodes indicate molecular species which appear only once in the composomes shown in B-F, whereas colored nodes are shared by at least two of these composomes. Colors are assigned arbitrarily but each color uniquely represents a particular molecular species, so as to allow visual inspection of similarities among composomes

A. The thresholded network corresponding to the entire rate enhancement β matrix.

B. The canonic composome. This is the composition which a GARD system assumes in the case of large assemblies ($N_0 >> N_G$). The canonic composome is approximated by the main eigenvector of the β matrix, i.e. the eigenvector with the highest eigenvalue, whose elements are all real [38]. The only molecular species shown are those whose concentration would reflect at least one molecule for an assembly size of 120.

C-F. Networks that correspond to four dynamic composomes, computed by numeric simulation of the GARD stochastic differential equation. These are as observed in a trace of 2,000 splits using the same size limit. The similarity of the composomes to each other (measured by H, cf. figure 2):

	B	C	D	E	F
Canonic (B)	1.0				
Composome1 (C)	0.6	1.0			
Composome2 (D)	0.7	0.6	1.0		
Composome3 (E)	0.8	0.7	0.4	1.0	
Composome4 (F)	0.9	0.6	0.6	0.7	1.0

Some networks modules (subset of connected nodes) are clearly shared by several composomes, e.g. the pentameric cycle in B, E and F.

split cycles due to efficient mutual catalysis that underlies homeostatic growth. These resemble fixed points of a dynamic system (Fig. 2).

A specific GARD system, defined by a particular β matrix, may have numerous composomes, and each of these defines a weighted network of catalytic interactions (Fig. 3). It is, however, legitimate to investigate such network and their properties through conversion to a binary network, based on a judiciously selected cutoff. This analysis may be equally applied to individual composomes (Fig. 3C-F), or to the calculated canonic composome (Fig. 3B). The different composomes may bear different degree of mutual similarities, as manifested in sharing of molecular species and in the values of mutual similarity measure H (Fig. 3, legend).

In the basic GARD formalism, the only chemical reactions being modeled are catalyzed exchange reactions – joining and leaving of molecules. The resulting dynamics involves compositional transitions that may be considered by some as resulting from mutations. When viewed within a limited time frame, the dynamics of this simple GARD model manifests graded transitions between different molecular networks, resembling an evolutionary process.

5 How Did Scale-Free Networks Arise

An important attribute of GARD is its parsimony, as it involves very few preassumptions, and stems directly from chemical kinetics of small molecules. As mentioned above, the resulting networks are graded or weighted, and therefore cannot be readily analyzed by the standard tools of degree distribution analysis.

Yet, with a threshold-based procedure it is possible to see that the degree distribution of a GARD network roughly obeys a binomial distribution (Fig. 4). This is to be expected, as these networks are derived from a randomly disposed matrix of interactions. Importantly, GARD dynamics, leading to composomes select molecular species such that the β values deviate from the original lognormal distribution (Fig. 5), showing an increased preponderance of high β values.

The scale free - power law behavior of biological networks is usually rationalized as being related to the formation of a few hubs with a large number of connections [31]. A parallel potential explanation could be in terms of selective preservation of very richly connected nodes. That the early GARD networks are not scale free is in agreement with the notion that such property is a result of a long evolutionary process. A crucial question is at what stage in evolution these properties arose, and how they are related to what distinguishes very early biological systems from later ones.

As described above, in the basic "joining GARD" model, the concentrations of different molecular species may undergo profound changes, but the basic properties of the molecules may not change. A more open-ended configuration is afforded by later GARD versions that include chemical reactions, in which oligomers are formed by covalent concatenation of monomers. In this extended GARD model the number of molecular species is an exponent of N_G (the size of the monomer repertoire). Preliminary analyses [32] show a more life-like behavior, and it appears that this polymer GARD formalism may also harbor a potential for a graded transition from random networks to scale-free ones. An intuitive rationalization is that a few of the vast number of oligomers that form in such a scenario might interact with or catalyze reactions of a large number of other compounds, hence become network hubs. This is by virtue

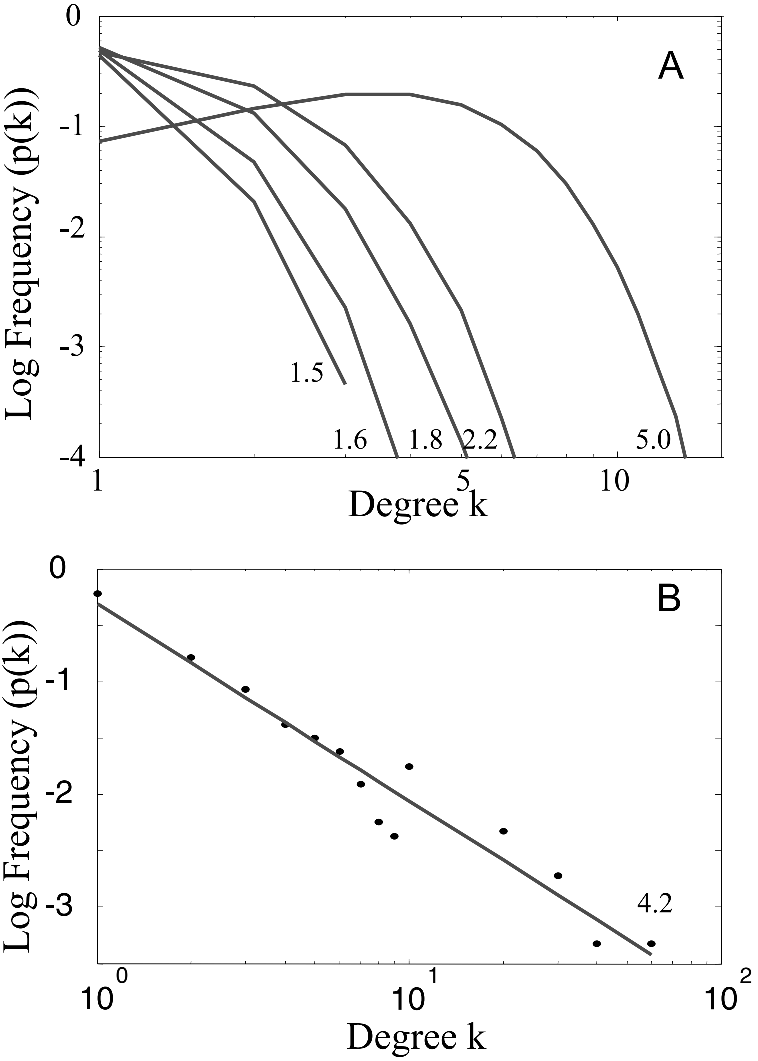


Fig. 4. Comparison of distribution p(k) of degree values k for composomes vs. protein networks, drawn as a double-logarithmic plot. For each curve, the average degree is shown next to its bottom

A. The degree distribution for GARD networks of canonic composomes with different cutoff values on concentration of selected molecular species. The distribution resemble a Binomial distribution with N equals N_E and p equals the probability for a catalysis β_{ij} to exceed the cutoff on β values. The range of degree values is much narrower than for a highly evolved network as shown in B. The distribution was computed for 1,000 canonic composomes with N_G =1,000. Similar distribution was also found for composomes observed in 150 GARD simulations with the same N_G and 1,000 GARD simulations with N_G = 300.

B. Degree distribution for yeast proteins interaction map [8], with 2114 proteins and with a total number of interactions of 4480 (average degree of 4.23). A linear power law relationship is seen, with the best fit equation of $p(k) = 0.494* k^{-1.75}$.

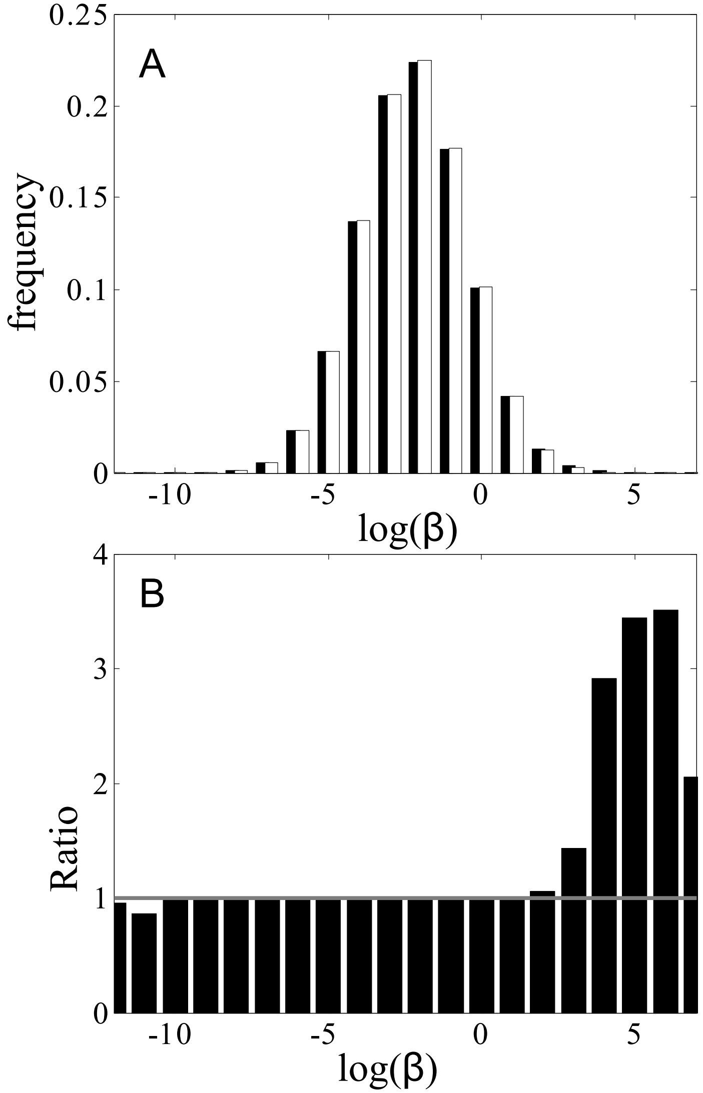


Fig. 5. Comparing the distributions of rate enhancement values (β) for composomes and their original β matrices

A. The distribution of rate enhancement values in the entire β matrix (light) and in the canonic composome (dark). The analysis was performed for 1,000 different β matrices with N_G = 1,000, whose values were randomly selected using a lognormal distribution with μ = -4 and σ = 4 in accordance to previous work [25].

B. The ratio the two distributions of Fig. 5A (canonic composome as the numerator). These two distributions are seen to differ significantly only for values of β larger than 10. In this range there is enrichment in the composome, suggestive of selection that favors species with higher values of rate enhancement. A similar phenomenon was observed also for 1000 dynamic composomes obtained from traces similar to those from which the networks in Fig. 3C-F were derived.

of "molecular adaptors" – sub-strings of the oligomers that may be shared by a large molecular repertoire, possibly in similarity to the small-world phenomenon describe for the repetitive Diels-Alder reactions networks [33].

6 Network Motifs

Molecular networks underlie many different functions in contemporary cells. Analysis of network motifs in biological networks shows that some embody distinct network motifs indicative of information processing [34]. We set out to explore the existence of such motifs in the GARD networks delineated above. In a preliminary analysis for network motifs in the canonic composomes of 1,000 different GARD system with N_G=1,000, we have observed that the feed-forward loop motif and feed backward loop motif tend to be overrepresented (Z score > 2) for about 5% of the composomal networks. To verify that these values are statistically significant, 10 control sets consisting of 1,000 random networks with the same size distribution of the original set, were subjected to the same analysis. Table 1 summarizes the results, showing that the enrichment of the feed-forward loop motif is not higher than expected by chance. On the other hand, feed-backward loop enrichment is statistically significant. This motif, which is a cycle of size 3, supports in a straightforward manner a process of homeostatic growth. It is thus suggested that in some GARD systems this motif may serve the role in the dynamics of the composomal network. Further investigation, including consideration of motifs of larger sizes, is currently underway both for monomer GARD and Polymer GARD.

Table 1. GARD network motifs

Motif	GARD	Control
None	952	976±4
Feed forward loop (A→B, B→C, A→C)	20	18.5±4.0
Feed backward loop (A→B, B→C, C→A)	**27**	4.5±1.9
Both	1	0.4±0.5

7 Sensitivity to Mutations in GARD Networks

Cells maintain a homeostatic composition despite variations to their external milieu. They are also often robust towards internal variations such as genomic mutations that may be as severe as gene deletions [35] In fact, in yeast, only about 20% of the genes were shown to be essential producing a non-viable phenotype upon deletion and about 40% of the genes hardly show any growth defect upon deletion. It was further shown [8] that genes encoding proteins which are highly connected in the interaction network produced more deleterious phenotypes. We asked whether GARD composomes and the networks that they represent have somewhat similar invariance properties.

We performed an analysis analogous to gene deletion within the GARD model. Repeated GARD simulations were carried out using the same molecular repertoire (β matrix) but in each round a different compound was completely depleted from the

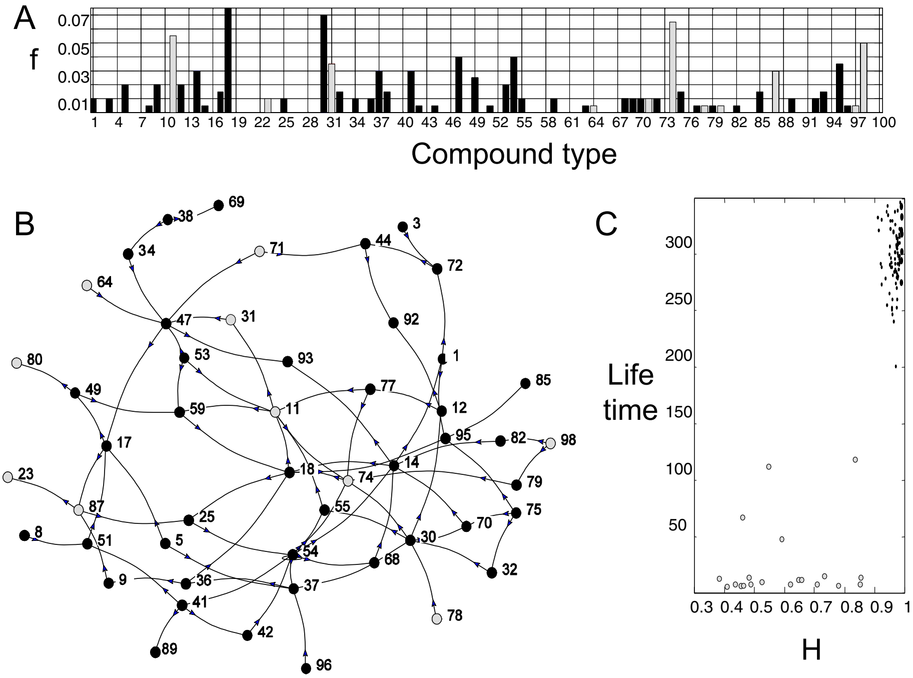


Fig. 6. Robustness of GARD networks to compositional mutations. A GARD simulation with N_G=100 was performed and the composome with the longest life-time selected. This simulation was subsequently repeated 100 times, each time with a different monomer depleted from the composomes external environment, hence also from its internal composition. For every resulting GARD trace the longest-living composome was tested for its cumulative lifetime and for its similarity (using a scalar product H) to the original composome

A. Compositional diagram for undepleted original composome, with compounds that are essential for homeostatic growth (H<0.7) labeled light grey. It is seen that essentiality is not necessarily correlated with monomer concentration.
B. The thresholded composome catalytic interaction network, with nodes colored light grey as in A. Essential monomers are not necessarily those that are highly connected hubs, and includes a significant number of terminal nodes with only one edge.
C. Correlation between the life-time of monomer-depleted composomes and their similarity (H) to the composome with no depletion. A majority of the depletion events appear to have a weak "phenotype (high lifetime and high H), while some are more severely affected. There is a broad correlation between the two quantitative measures for composome effectiveness.

composomes external environment, and thus from the network itself. We found that most compounds have only a small effect on both growth rate and molecular composition of the composome (Fig. 6C). A minority compounds (10-20%) are essential, and have a marked effect on the functional properties of the composomal network, as their removal significantly changes the assembly composition and/or reduces its growth rate. Fig. 6A, B respectively show the composition and the interaction network of a typical composome, highlighting the compounds essential for network stability. Surprisingly, essential compounds were not necessarily those with the highest concentration or highest connectivity within the network, suggesting non-trivial network properties.

8 Conclusion

The GARD model provides elaborate computing tools that help address prebiotic entities via the tools of present day Systems Biology. Since some characteristics of early GARD assemblies are shared with modern biological networks, the analyses described here may also lead to a better understanding of networks in present day life. In parallel, System Biology tools could assist in constructing better models for probing the important question of life's emergence.

References

1. Gilbert, W.: The RNA world. Nature 319 (1986) 618-618
2. Gesteland, R. F., Cech, T. R. ,Atkins, J. F.: The RNA World. 2nd edn. Cold Spring Harbor Laboratory (2000)
3. Joyce, G. F.: The antiquity of RNA-based evolution. Nature 418 (2002) 214-221
4. Luisi, P. L.: Introduction (to COST27 special issue). Origins of Life and Evolution of the Biosphere 34 (2004) 1-2
5. Eigen, M.: Selforganization of matter and the evolution of biological macromolecules. Naturwissenschaften 58 (1971) 465-523
6. Kauffman, S. A.: The origins of order - Self-organization and selection in evolution. Oxford University Press (1993)
7. Jeong, H., Tombor, B., Albert, R., Oltvai, Z. N. ,Barabasi, A. L.: The large-scale organization of metabolic networks. Nature 407 (2000) 651-654
8. Jeong, H., Mason, S. P., Barabasi, A. L., Oltvai, Z. N.: Lethality and centrality in protein networks. Nature 411 (2001) 41-42
9. Wagner, A.: The yeast protein interaction network evolves rapidly and contains few redundant duplicate genes. Mol. Biol. Evol. 18 (2001) 1283-1292
10. Wagner, A. ,Fell, D. A.: The small world inside large metabolic networks. Proc R Soc Lond B Biol Sci 268 (2001) 1803-1810
11. Giot, L., Bader, J. S., Brouwer, C., Chaudhuri, A., Kuang, B., Li, Y., Hao, Y. L., Ooi, C. E., Godwin, B., Vitols, E., Vijayadamodar, G., Pochart, P., Machineni, H., Welsh, M., Kong, Y., Zerhusen, B., Malcolm, R., Varrone, Z., Collis, A., Minto, M., Burgess, S., McDaniel, L., Stimpson, E., Spriggs, F., Williams, J., Neurath, K., Ioime, N., Agee, M., Voss, E., Furtak, K., Renzulli, R., Aanensen, N., Carrolla, S., Bickelhaupt, E., Lazovatsky, Y., DaSilva, A., Zhong, J., Stanyon, C. A., Finley, R. L., Jr., White, K. P., Braverman, M., Jarvie, T., Gold, S., Leach, M., Knight, J., Shimkets, R. A., McKenna, M. P., Chant, J. ,Rothberg, J. M.: A protein interaction map of Drosophila melanogaster. Science 302 (2003) 1727-1736
12. Barabasi, A. L. ,Oltvai, Z. N.: Network biology: understanding the cell's functional organization. Nat Rev Genet 5 (2004) 101-113

13. Li, S., Armstrong, C. M., Bertin, N., Ge, H., Milstein, S., Boxem, M., Vidalain, P. O., Han, J. D., Chesneau, A., Hao, T., Goldberg, D. S., Li, N., Martinez, M., Rual, J. F., Lamesch, P., Xu, L., Tewari, M., Wong, S. L., Zhang, L. V., Berriz, G. F., Jacotot, L., Vaglio, P., Reboul, J., Hirozane-Kishikawa, T., Li, Q., Gabel, H. W., Elewa, A., Baumgartner, B., Rose, D. J., Yu, H., Bosak, S., Sequerra, R., Fraser, A., Mango, S. E., Saxton, W. M., Strome, S., Van Den Heuvel, S., Piano, F., Vandenhaute, J., Sardet, C., Gerstein, M., Doucette-Stamm, L., Gunsalus, K. C., Harper, J. W., Cusick, M. E., Roth, F. P., Hill, D. E. ,Vidal, M.: A map of the interactome network of the metazoan C. elegans. Science 303 (2004) 540-543
14. Kunin, V., Pereira-Leal, J. B. ,Ouzounis, C. A.: Functional evolution of the yeast protein interaction network. Mol Biol Evol 21 (2004) 1171-1176
15. Alm, E. ,Arkin, A. P.: Biological networks. Current Opinion in Structural Biology 13 (2003) 193-202
16. Monk, N. A. M.: Unravelling nature's networks. Biochemical Society Transactions 31 (2003) 1457-1461
17. Newman, M. E. J.: The structure and function of complex networks. Siam Review 45 (2003) 167-256
18. You, L. C.: Toward computational systems biology. Cell Biochemistry and Biophysics 40 (2004) 167-184
19. Wachtershauser, G.: Evolution of the first metabolic cycles. Proc Natl Acad Sci U S A 87 (1990) 200-204
20. Dyson, F. J.: Origins of Life. 2nd edn. Cambridge University Press (1999)
21. Segre, D. ,Lancet, D.: Composing life. Embo Reports 1 (2000) 217-222
22. Segre, D., Ben-Eli, D., Deamer, D. W. ,Lancet, D.: The lipid world. Origins of Life and Evolution of the Biosphere 31 (2001) 119-145
23. Morowitz, H. J.: The Emergence of Everything: How the World Became Complex. Oxford University Press (2002)
24. Segre, D., Lancet, D., Kedem, O., Pilpel, Y.: Graded autocatalysis replication domain (GARD): Kinetic analysis of self-replication in mutually catalytic sets. Origins of Life and Evolution of the Biosphere 28 (1998) 501-514
25. Segre, D., Ben-Eli, D. ,Lancet, D.: Compositional genomes: Prebiotic information transfer in mutually catalytic noncovalent assemblies. Proceedings of the National Academy of Sciences of the United States of America 97 (2000) 4112-4117
26. Shenhav, B., Segre, D. ,Lancet, D.: Mesobiotic emergence: Molecular and ensemble complexity in early evolution. Advances in Complex Systems 6 (2003) 15-35
27. Shenhav, B., Kafri, R. ,Lancet, D.: Graded Artificial Chemistry in Restricted Boundaries. Proceedings of 9th International Conference on the Simulation and Synthesis of Living Systems (ALIFE9), Boston, Massachusetts, USA (2004) 501-506
28. Lancet, D., Sadovsky, E. ,Seidemann, E.: Probability Model for Molecular Recognition in Biological Receptor Repertoires - Significance to the Olfactory System. Proceedings of the National Academy of Sciences of the United States of America 90 (1993) 3715-3719
29. Lancet, D., Kedem, O., Pilpel, Y.: Emergence of Order in Small Autocatalytic Sets Maintained Far from Equilibrium - Application of a Probabilistic Receptor Affinity Distribution (RAD) Model. Berichte Der Bunsen-Gesellschaft-Physical Chemistry Chemical Physics 98 (1994) 1166-1169
30. Rosenwald, S., Kafri, R. ,Lancet, D.: Test of a statistical model for molecular recognition in biological repertoires. Journal of Theoretical Biology 216 (2002) 327-336
31. Han, J. D., Bertin, N., Hao, T., Goldberg, D. S., Berriz, G. F., Zhang, L. V., Dupuy, D., Walhout, A. J., Cusick, M. E., Roth, F. P. ,Vidal, M.: Evidence for dynamically organized modularity in the yeast protein-protein interaction network. Nature 430 (2004) 88-93

32. Shenhav, B., Bar-Even, A., Kafri, R. ,Lancet, D.: Polymer GARD: computer simulation of covalent bond formation in reproducing molecular assemblies. Origins of Life and Evolution of the Biosphere (2005)
33. Benko, G., Flamm, C. ,Stadler, P.F.: Generic properties of chemical networks: Artificial chemistry based on graph rewriting. Advances in Artificial Life, Proceedings 2801 (2003) 10-19
34. Milo, R., Itzkovitz, S., Kashtan, N., Levitt, R., Shen-Orr, S., Ayzenshtat, I., Sheffer, M.,. Alon, U.: Superfamilies of Evolved and Designed Networks. Science 303 (2004) 1538-1542
35. Waddington, C. H.: Canalization of development and the inheritance of acquired characters. Nature 150 (1942) 563-565

Virtualization in Systems Biology: Metamodels and Modeling Languages for Semantic Data Integration

Magali Roux-Rouquié and Michel Soto

UMR 7606 - CNRS - Université Pierre et Marie Curie, LIP6,
8 rue du capitaine Scott, 75 015 Paris, France
{magali.roux, michel.soto}@lip6.fr

Abstract. We examined the process of virtualization to deal with data intensive problems. Since data integration is a first-order priority in systems biology, we started developing a new method to manipulate data models through ordinary metadata transactions, i. e. by preserving the original data format stored in resources. After discussing why metamodels are made for, and the interplay of modeling languages in metamodel design, we presented a systemic metamodel-driven strategy to integrate semantically heterogeneous data.

1 Introduction

The process of virtualization has been defined as the mapping of an abstract data set to a virtual space according to three majors intertwined steps consisting of data selection for representing the problem space, assumptions definition to define the final virtual space, the mapping between the starting space and the final space through a metaphor [1]. Since that, virtualization has been extended to the management of distributed data, the main goal in this approach being to deal with data intensive problems [2].

The major features concern with the process of virtualization are:

- The preservation of data and knowledge in their actual format: the current physical reality and its putative evolution are not impacted at all by the virtualization process. This means that data and knowledge production can go their own way without any necessary change.
- The operationability: virtualization is not just abstraction, it allows to recursively transform the physical reality according to the lessons learned in virtual reality. In these respects, virtualization aims to actualize physical reality and, conversly, any change in physical reality has its counterpart in virtual reality.

Virtualization may be defined more formally by sets E and V corresponding, respectively to physical reality and virtual reality and the two functions M_E and M_V:

$$M_E : E \rightarrow V,\ M_V : V \rightarrow E\ .$$

C. Priami et al. (Eds.): Transactions on Computational Systems Biology, LNBI 3380, pp. 28–43, 2005.

The functions M_E and M_V define the mapping rules between E and V. V is also named metaphor, which refers to a domain of knowledge. The choice of metaphor is driven by both the final goal and the actual possibility to define M_E and M_V.

The metaphor V can be homomorph or heteromorph. A metaphor V is homomorph when the initial E paradigm is used in the V design. Conversely, a metaphor V is heteromorph when a paradigm, different from the initial E one, is used to design V; as an important consequence, different metaphors can be selected to fit with different aims, according to a single physical reality.

Whatever M_E and M_V complexity, virtualization is a suitable process either to face heterogeneity problems or to shift from one domain of knowledge to more convenient representation as illustrated by the following examples.

Information technologies (IT) have been successful in using virtualization to deal with heterogeneity problems, to increase productivity of IT tools and to spread IT products to non IT-skilled users:

- Internet can be considered as the most famous and successful solution to a problem, thanks to virtualization. Because of protocol heterogeneity, existing computer communication architectures were not able to interoperate although they were built for the same purpose: exchanging data between distant computers. This is the process of virtualization that both allowed to integrate existing communication technologies and to preserve the future development; this was a key feature for internet success. To overcome this interoperability problem, a new protocol, named Internet Protocol (IP) and a new computer address format (IP address) were designed upward from the actual computer communication architectures; both IP and IP addresses were virtual in the sense they were not "natively" understood by any computer communication architectures. Mechanisms from IP and IP addresses to actual protocols and actual addresses (named respectively physical protocols and physical addresses) were designed to provide mapping. This was achevied without putting any constraints on future technological developments. Internet is an example of how an homomorph metaphor, the protocol/address paradigm, is of help in the virtualization process.
- Another example concerns the desktop metaphor enabling the use of computer hardware with a limited knowledge of the operating system (OS); in this case the metaphor was heteromorph since the desktop paradigm is very different from the computer hardware one.
- The third example concerns portability, which allows a software to be run on any microprocessor architecture without rewriting it (even partially) but just compiling it with the ad hoc compilator. Nevertheless, the use of compilators can not mask all differences between microprocessor architectures and there are always remaining portability problems. These increase dramatically when the target architecture is not know in advance as it occurs on internet. Virtualization was used to overcome this portability problem in the context of internet. A virtual microprocessor architecture, named virtual machine (or pseudo machine), and new high level programming language, named Java,

was designed: Java compilators traduced Java written softwares in native instructions, named byte code, for the virtual machine and the virtual machine architecture closely mimiked actual microprocessor architectures. In these respects, the virtual machine is an isomorph metaphor with regards to the microprocessor architecture paradigm.

In the field of systems biology, the diversity of biological sources as well as experimental design and methodologies results in heavy heterogeneity, not only at the technological level but also at the semantic level; making data integration a major issue. In parallel, another challenge aims to simulate biological systems to predict their behavior; the ultimate goal being to understand not only their structure but also their dynamic [3].

To approach this problem, virtualization could be of great help. As it does not require any modification in the way of the data are produced, it preserves all accumulated experience and skills. Only mapping rules are concerned with the problem of data heterogeneity and further modifications in experimental approaches. Virtualization can be achieved by developing a metamodel-driven strategy to elicit a model upward from the current knowledge; the availability of such a metamodel for biological systems could be used as a grid for data integration. This needs to have a clear understanding of what a metamodel is made of, how it is designed and what it is doing for.

In these respects, metamodeling is not a final goal but an interface between data from the physical world and models in the virtual world. The mapping between the two worlds is iterative and model transformations are the operational side that misses single abstraction. As matter of fact, metamodeling is the junction that makes it possible to extend the database methodology to simulation thanks to the process of virtualization.

2 Metamodel

Since data integration is a first-order priority in systems biology, metamodel-driven strategies that are the foundation for data integration, should get much attention.

What is a metamodel ?

The methodology to provide a generic metadata abstracting and structuring all models into an integrated metadata repository consists into metamodeling. This means that a metamodel provides all concepts, properties, operations and relations between concepts necessary for designing any kind of models to be contained in it, at some level of abstraction and from some perspective. In these respects, a metamodel makes it possible to map multiple models into a single model by coalescing those elements identified as representing the same concepts.

In a metamodel, the notion of semantics is very important and reflects not only the need to model things in the real world (the signifier or the substance; for example, a molecular structure), but also the meaning that these things have to have for the purpose of the metamodel (the signified, the role in a particular context; for example, a molecular function).

To achieve a metamodel-driven integration, it is necessary to understand the meaning of data in all systems to be integrated: which data have the same meaning, which data are complementary and how they are related. Performing such a semantic analysis yields a metamodel for the types of data to be integrated; In these respects, one metamodel is a models integrator; conversely, the metamodel instances are models.

As a metamodel upward from the model layer, metamodeling deals with the full scope of paradigm translation, enabling the use of one model described into in one formalism to be transformed into a model in another formalism as far as each model is obeying well-formed rules, leading to possible model transformation and coupling.

In discipline-specific metamodeling (DSM), metamodels are the way to explicit the meaning of concepts in such a specific realm and to capture the relevant concepts. Among advantages, this approach allows to organize data without any modification of their structures. In addition, it makes it possible to check for the consistency of the multiple specifications as they do not conflict with one another and must be "in some sense" consistent. Also, it makes it easier to ensure model validation by comparing the computerized model to the model designed by the domain expert, for satisfactory range of accuracy.

From a technical point of view, a metamodel allows all local models and other metadata contained in it to be added, deleted, or modified through ordinary metadata transactions accounting for data and knowledge virtualization, in contrast to a fixed global data model.

In practice, the building of a metamodel will consider four levels:

- the information level 1 or data level, which consists in the basic facts to integrate;
- the data model at level 2, i. e. how the data are organized (for example, the model of a database consists in a special implementation of the metamodel);
- the metamodel (level 3) that describes and organizes concepts with a set of well-formed rules, to integrate all models from level 2;
- the language for metamodeling (metametamodeling, level 4) that use concepts and the relations defined in the metamodel, and may consist in, both, textual and/or graphic notations.

We present in figure 1, an example for a core language metamodel using a class diagram in the object-oriented paradigm: the left part describes classes accounting for the generic description of language elements called descriptor elements; the right part shows classes responsible for instantiation of generic elements and named instance elements. These two blocks are linked by binary associations setting the connection between the generic descriptor elements and the instance elements. They describe which elements of the instance level belong to which element of the generic level.

The Meta Object Facility (MOF) is a well known metamodel maintained by OMG [4]. It allows to create instances which are models, such as the Unified

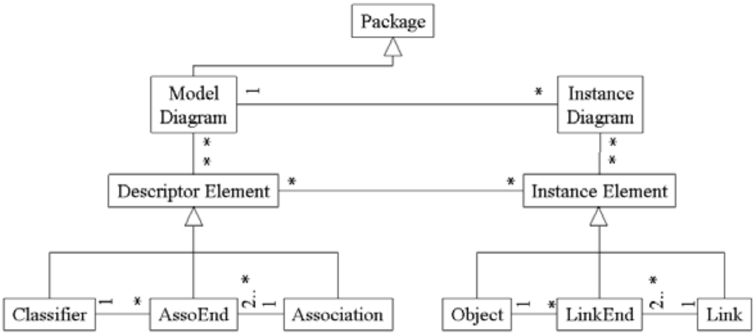


Fig. 1. A core metamodel for generic description and instantiation of language elements

Modeling Language (UML) or the Common Wharehouse Metamodel (CWM). A four levels modelization is used to metamodel UML: the UML model is defined with respect to the MOF, and the MOF is self-contained, i. e. it is used for self-definition. We summarized these four levels of metametamodelization (meta^{2}modelization), which have been defined by OMG:

- the M0 level contains specific information described at level 1 and is data;
- the M1 level represents instances of the UML metamodel;
- the level M2 corresponds to the UML metamodels described with the MOF, the UML metamodel being the language for creating UML models;
- the higher level, M3, corresponds to the MOF which is the language for designing metamodels. For example, the metaclass MOFClass has an instance which is the UMLClass; this recursive nature of the metamodel approach to the definition of the syntax of the UML (see below for details) is elegant.

3 Modeling Languages

In addition to data integration, a metamodel is especially powerful when it is self-contained and does not require auxiliary means or external tools to specify itself; as such, it can be used as a true language to deal with as mentioned with the MOF which, not only allows the design of metamodels, but also allows its own design.

What is a modeling language ?

A modeling language is a language that contains all the elements with which a model can be described. It is a set of symbols and rules used to specify concepts and constructs for any kind of system; they may be textual and/or visual, structural and/or behavioral. Modeling languages are true languages and have syntax (the notation) and semantics (the meaning). Syntactic issues focus purely on the notational aspects of the language and modeling languages have to have a rigid syntax if they have to be further compiled.

In these respects, the structure of a modeling language is the following:

- An abstract syntax defines the different ways symbols may be combined to create well-formed models. Syntax defines the formal relations between the elements of the language; it deals with the form and the structure of the various expression of the language without any reference to their meaning.
- A graphical notation is the concrete syntax, the representation with well-formedness rules. For textual languages, the concrete syntax is a set of characters, the alphabet, characters are grouped into words and arranged into sentences according to precise grammar rules.
- A syntax mapping relates abstract syntax to concrete syntax and back; for example, the syntactic operator ”sum” is mapped to the graphical notation “+”.
- A semantic domain defines the elements that are described by the abstract syntax; semantics considers the meaning of syntactically correct models: what to think, what to feel, what to do for natural language, the computer behavior for programming language.
- A semantic mapping gives the rules that map the syntax to constraints on things in the semantic domain, it gives the ”meaning” of the model according to the syntax. For example, the syntactic graphical operator ”+” in a arithmetic expression is mapped to the addition operator of arithmetic, so that the meaning of the expression 1+2 is to be the number 3, which is the sum of the two numbers.

With the standardization of the UML, the aim was to gather within a unique notation the best features of object-oriented languages. The usage of UML as a modelling language has an important impact and UML descriptions turn out to be abstractions used to capture important properties of the systems to be developed, notably in terms of static structure and dynamic behaviour. In these respects, UML is a true language and as such has syntax and semantics. The UML standard has chosen to use a metamodeling approach based on the very popular class diagram to characterize the abstract syntax of the language; nevertheless, the language is composed of an additional set of notations that may overlap. For example, the state diagram notation can be used to express the same information that could be express in terms of pre/post conditions on operations in class diagram, but there are other aspects of states diagrams that can not. Let us consider some examples of the semantics and syntax of the UML according to the dynamic aspects of the language.

- Statechart: from a semantic point of view, the UML statechart represents the behaviour of entities capable of dynamic behaviour by specifying their responses to the receipt of event instances. Typically, it is used for describing the behaviour of classes. From a syntactic point of view, a statechart is a graph that represents a state machine. States and various other types of vertices (pseudostates) in the state machine are rendered by appropriate state and pseudostate symbols, while transitions are generally rendered by directed arcs that interconnect them. A statechart maps into a StateMachine and a StateMachine is owned by a model element capable of dynamic behaviour.

– State: in UML, a state is a condition during the life of an object or an interaction during which it satisfies some condition, performs some action, or wait for some events. A state may be simple or composite it is used to model an ongoing activity that may be specified as a nested state machine or by a computational expression. A state is shown as a rectangle with rounded corners. A state may be subdivided into multiple compartments separated from each others by a horizontal line. Notably, internal transitions compartment holds a list of internal actions or activities that are performed while the element is in the state. A state symbol maps into a State. A composite state is decomposed into two or more concurrent substates or into mutually exclusive disjoint subtates; and any substate of a composite state can also be a composite state of either type. The notation of a composite state allows showing its internal state machine structure; concurrent states are shown by tiling the graphic region of the state using dashed lines to divide it into substates; in contrast, disjoint states are shown by showing a nested state diagram within the graphic region.
– Event: an event is a noteworthy occurrence; for practical purposes in states diagrams, it is an occurrence that triggers a state transition. Events may be of several kind. For example, the receipt of an explicit "signal" from one object to another results in a signal event instance; it is denoted by the signature of the event as a trigger on a transition. A signal can be declared using the <<signal>> keyword on a class symbol in a class diagram; such keyword is specified as <<stereotype>>.
– Transition: a transition is a relationship between two states indicating that an object in a first state will enter the second state and perform specific actions. It is notated as a solid line originating from the source state and terminated by an arrow on the target state. A transition string and the transition arrow that it labels together, map into a Transition and its attachment. A concurrent transition may have multiple sources states and target states. It represents synchronization and/or a splitting of control into concurrent threads. From the semantic point of view, a concurrent transition is enabled when all the sources states are occupied. After a compound transition fires, all the destination states are occupied. A concurrent transition includes a short heavy bar (a synchronization bar, which can represent synchronization, forking or both). A bar with multiple transition arrows leaving it maps into a fork pseudostate; conversely, a bar with multiple transition arrows entering it maps into a join pseudostate (figure 2).

These limited examples on behaviour specifications clearly point out of the respective parts relying UML syntax and semantics and their mapping. The very intuitive UML notation is even expressive enough to account for a large variety of situations; in these respects, UML customization can be achieved thanks to UML profiles that specify "standard elements" beyond those specified by

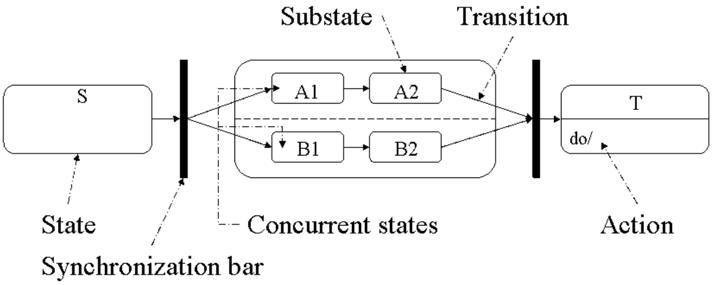


Fig. 2. States and concurrent transitions

the identified subset of the UML meta-model. (OMG Document: ad/99-03-10). Because it gathers the best features of object-oriented language, we think that UML fits all criteria for being developed as a profile in the realm of systems biology. This would benefit the major efforts achieved for the standardization of the language and the fine-tuning to systems biology would be achieve through extension capabilities. Accordingly, a limited number of well-known symbols would be necessary for deciphering the various states of one particular entity, most of the syntax and the semantics being defined by the language itself; in contrast, the dialect in systems biology being refined according to domain-specific ontology, metadata, etc.

The needs for language in Biology made of a limited number of symbol and a simple grammar, have been emphasized recently [5].To support this requirement, it was noticed that more than 75.000 articles were published since 1997 about the apoptosis death-programmed process without giving a clear understanding of it. It was suggested that poor data integration was accounting for such heavy difficulties. To encompass these bottlenecks, an international initiative was launched to set up the Systems Biology Markup language (SBML), a XML-based language, to facilitate data exchange and a Systems Biology Workbench (SBW) was developed for having heterogeneous application components to communicate [6]. A parallel project, BioSpice was using a Model Definition Language that was currently identical to SBML Level 2 [7]. Similarly the CellML project is an XML-based open standard for describing and exchanging models of cellular and subcellular processes [8]. In our hands, theses approaches mostly focus on technological integration and deal with data exchange showing variations between formatting whereas the semantic approaches was dealing with a limited number of topics, all necessary for building workflow models but not expressive enough to account for the large variety of biological phenomena and experimental approaches to depict. To fill this gap, graphical languages are being developed to achieve more detailed specifications. Cook [9] proposed a basic lexicon of icons and arrows for describing the function of complex biological systems. This approach was rooted in the work of Khon [10] that delineated large sets of molecular interactions maps. These initiatives and others [11, 12] have been synthesized to propose a standard graphical notation for specifying biological networks; introducing more structured specifications of biological systems in terms of expressiveness, consistency, extensibility, mathematical translation and software support [13].

Otherwise, ontologies [14] are under development to provide standardized vocabulary; they concern mostly the GO consortium that provides well-structured controlled terms on *Molecular Function*, *Biological Process*, *Cellular Component* [15], as well as related projects not to mention the BioProcess ontology that distinguish *logical* and *biochemical actions* to describe biochemical pathways [16].

At last, semantic mapping is an important part of the language structure in identifying important concepts and how these concepts fit together. This approach has been launched in molecular biology by the pioneer work of Paton [17] that identifies core question or concept, subordinate ideas that help explain or clarify the main concept, details, inferences and generalization that are related to each. This approach, which expressed biological knowledge as a society of graph, could be of great help in further topology mappings between models; for example, by mapping concepts from a vertex set of to a single vertex or from a path to a single edge.

4 Systemic Metamodel and UML Profile for Systems Biology

Of invaluable interests, all these initiatives can be merged within metamodel(s)in a virtualization perspective. Developing metamodeling approaches for systems biology require identifying primitive concepts, properties, operations and relations between concepts necessary for the specification of biological systems in terms of structure and behavior as well as the methodological approaches involved (i. e., genomics, transcriptomics, proteomics, etc.). In a more practical view, this can be approached through the metaphor of reactive systems to organize concepts and data in systems biology, just as the Windows makes use of the metaphor of desktop that was more familiar to office worker.

This issue can be achieved in the framework of the systemic paradigm [18], which allows to state that:

- a functional entity can be efficiently represented as the interface between an internal and an external environment in which it is evolving and on which it is acting;
- the behavior of this entity can be described as the trajectory of its states within a *Time*, *Space*, *Form* frame;
- events occurring from either internal and/or external environment may allow some changes in state variables and the consecutive firing of state transitions;
- all these changes can be modelled as the mapping between state description and process description.

It must be mentioned that the concept of *action* is central for virtualization [19], and the systemic paradigm in centered on. In these respects, viewing molecular entities as *processes*, interacting molecules as *communicating processes*, the change in interacting molecules as the *change in process states*, etc. [20], emphasizes the isomorphism between biological systems on the one hand, and reactive systems on the other hand, making real-time extensions of the UML available for customization to systems biology [21, 22]. This takes advantage of state diagrams to depict dynamic systems, which are grounded on the pioneer Harel's work on statecharts. This formalism was recently applied to biology by Kam et al. for the modeling of the immune systems [23]; nevertheless, Kam's approach missed a reference to an explicit systemic metamodel to organize data.

As defined in section 2, a metamodel must bring all elements to define a model; in these respects, a metamodel must acknowledge the mandatory requirements to integrate main data in systems biology, in terms of substances, constraints and processes (figure 3). In the systemic metamodel, this was achieved as follows:

- Substances consist in biological entities, which have a persistent identity. This must be clearly distinguished from the set of states taken by theses entities and that refer to their history. In our model, substances referred to any kind of biological entities, from organisms to molecules, and were arranged into specialization (Is-a) and composition (Has-a) hierarchies. In other words, substances were concerned with the identity (permanent) of the entities and not their states, which are transitory. As a consequence, the system was described with a limited number of classes (and relations) as the entities derived from these classes have several states. Substances were specified in the main class *Substance*, the child class *S_Molecular* has a specialized class *S_Protein* with a proteinId which stores accession numbers to database.
- The constraints (relationships) between system components constitute important aspects of living systems and most of the information we have on constraints in pathways comes from biochemistry chemical. But such changes are only half the story and our understanding of the functioning has to be completed with the spatial-temporal location of molecules in the cells as well as the properties attached to their three-dimensional behavior, i.e. all context effects. In our model, three kinds of class accounted for such Space-Time-Form constraints: (1)The *SpaceOccurrence* specified the position of any entity with regards to its external environment, (2) the *TimeOccurrence* referred to the age, time, period of any active entity (the time, the period this entity is functioning), (3) the *FormOccurrence* specified the functional isoform (if any) of the substance. The *FormOccurrence* was described as the set of *BioTransformation* (for example, phosphorylation, acetylation, etc.) that operated on the *BioSubstance.*
- Processes are represented - according to the systemic guidelines - as the state trajectories of entities functioning over time. Understanding processes requires the description, the modeling and the simulation of state trajectories of these entities. To achieve this goal, the concept of active object is very well adapted as active objects have their own behavior that can be described with subsets of state machines. In our metamodel, this allows us

to delineate the elementary entity involved in process and named *Functional Unit* (*FUn*). A *FUn* has internal and external environment. Internal environment delineates the roles of *FUns* as *infraFunctionalUnit* (*infraFUn*): a functional entity has components that assume specific tasks to function; this corresponds to its internal environment; for example, the components of the general transcription factor TFIID that consist in TBP together with $8-12$ tightly bound subunits, constitute the internal environment of TFIID and play the role of "infra" functional units. In addition, *FUns* play two kinds of roles according to the external environment: they are *FUns* nesting *FUns*; in our metamodel, this role is named *supraFunctionalUnit* (*supraFUn*); furthermore, in their external environment, *FUns* have neighbor reacting entities, they referred to their *neighborFunctionalUnit* (*neighborFUn*); for example, their reactiveness can be assigned according to distances and/or domain affinity at the molecular level, or concentration at the population level. This can be modeled as messages passing between *FUns* and results into state transition, from the current state to a new state.

Summarizing the major features of the systemic metamodel needs to underscore the clear separation between structural and behavioral aspects with respect to the functional entities (*FUns*) which were modeled as processes using active objects, in contrast to substance that was modeled using passive objects. This was achieved in a perspective to extend the database methodology to the virtualization approaches.

Accordingly, the metamodel-driven strategy can be used to guide data integration as all concepts were being contained in it. Shortly, if we consider the Microarray gene expression data model [24] that is detailed in the adopted specification of the OMG [formal/03-02-03], the *BioSequence* package, which contains representations of a DNA, RNA or protein sequence, could be integrated into the *Substance* package in the systems biology metamodel. Otherwise, limited part of a data model could be integrated to the metamodel; for example, the EntityLink.entity_id_(1,2) field of the Macromolecular Structure Specification [OMG Formal/02-05-01) that represents the entity ids of the two entities joined by a linkage, could be integrated at the metamodel level to specify the binding between FUns. The same approach could be achieved with both the *Bind-action-*

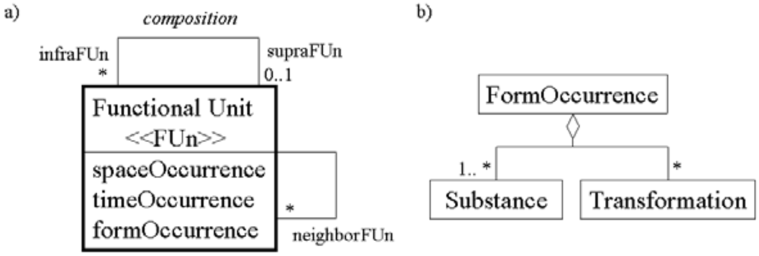


Fig. 3. Systemic metamodel: (a) the active upper class *FUn*, (b) the passive class *Substance*

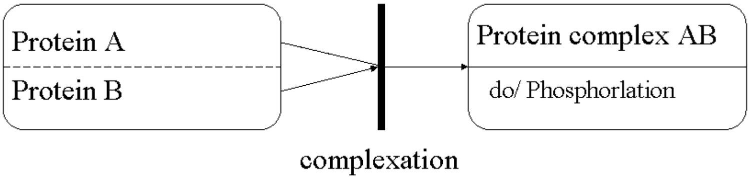


Fig. 4. Complexation is a synchronization process

type in the BIND database model [25] and the *Action-type* in [16], that can be integrated in the metamodel to specify action occurring in a particular state (figure 4).

Thus, a metamodel for systems biology would allow describing, in a common way, the data found in the large variety of physical sources by clarifying the hypotheses and the axioms that hold among concepts, as previously stressed in reference [18] concerning relationships between *Being* (OMB) [14] *Structural element* (EcoCyc)[26], *Cellular function* (GO) [15], *Cellular role* (YPD) [27] *Structural element* (EcoCyc) [26], *Processes* (GO) [15] *Pathways* (KEGG) [27], etc.

Because of the isomorphism between the systemic specifications of biological systems and the reactive systems used as a metaphor to drive the virtualization process, we consider the customization of the UML to systems biology, named SB-UML, instead of developing a new language. In order to assess the relevance of developing such UML extensions to systems biology (UML profile), we initiated the writing of Khon's molecular interaction maps into SB-UML. Shortly, we found SB-UML more expressive than the referenced graphical notation as it allows representing additional dynamic features. The figure 4 presents the complexation of protein A to protein B showing concurrent behavior of protein A and B until synchronization into complex AB is achieved. This approach emphasizes pattern occurrence allowing factored processes with, among important advantages, software reusability (to be published elsewhere).

Details on the way one entity changes its respective states, take advantage of the reactive systems metaphor. Figure 5a shows aspartate transcarbamoylase, an instance of the the allosteric enzyme active class, containing an allosteric region and an enzymatic region, all stereotyped as <<FUns>>. The figure 5b gives the structure view of the enzyme that shows how the regulatory and enzymatic regions communicate with their environment through specific amino-acid residues symbolized as black squares. When the required interaction (signal) targets site-specific amino-acids, a transition is fired from the initial state to the final state. This corresponds to the changing from a free state to a bound state for the allosteric region and from an inactive state to an active state for the enzymatic one; both processes are concurrent and theses changes occur simultaneously. When the new states become occupied, the enzyme is allowed to perform carbamoyl transfer (figure 5c). As shown, extending UML to systems biology allows accounting for details that are no more mentioned in usual specifications because of some limits in language expressiveness.

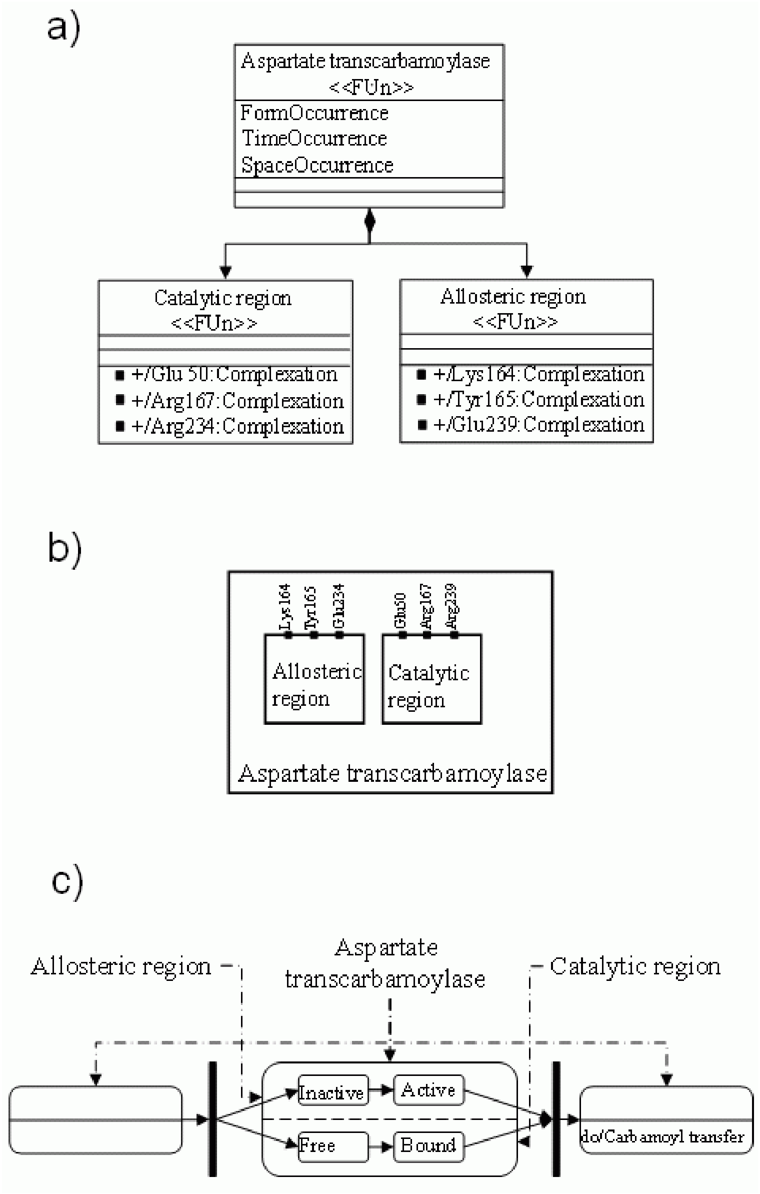


Fig. 5. Instantiation of the systemic metamodel: aspartate transcarbamoylase. (a) class diagram, (b) structure diagram, (c) state diagram

5 Conclusion and Perspectives

In this paper, we presented the metamodel-driven strategy as a key step in the virtualization process. As this approach requires a metaphor relevant to the final goals in the field of systems biology, we found that biological systems could be efficiently modeled as reactive systems within the systemic framework as previously reported [18]. This allowed us specifying any biological entity as an attribute vector depending of time, space and form variables, $\overrightarrow{v}\ (t, s, f)$, and it was achieved in the object-oriented paradigm using a systems biology extension of the UML (SB-UML). This aims to delineate a UML profile for systems biology, taking advantage of the UML expressiveness with regards to reactive systems.

In reactive systems, entities have their own thread of control and can behave concurrently with steps for synchronization. This shows isomorphism to biological entities which behave independently, although in a synchronized manner. The reference to the systemic framework was achieved according to the design of the attribute vector centered on the concept of form. This allowed to clearly distinguish the structure of biological entities from their behavior, as most of the structural data can be assigned to the substance passive class, whereas the form attribute of the active Functional Unit (FUn) class can be referred to the dynamic substance transformation according to time and space occurrences.

It must be strongly emphasized that a metamodel-driven strategy is not just setting a model upward from the other models but it has a core function for designing a new model from a former one or from the physical reality. This function is central to the process of virtualization, which realises the coupling of a physical reality with a constructed virtual reality. This is achieved by preserving the diversity of data, without any modification at the data model level and without any hypothesis on their future improvements. As matter of fact, data evolution only impacts on the metamodel and the mapping rules between the physical reality and the virtual reality.

Virtualization leads to operationability, in the sense of actionability, since the virtual reality actualizes the physical reality; so that, any change in physical reality is reflected into virtual reality. In these respects, operationability is the main difference between abstraction and virtualization: abstraction aims to provide a general and synthetic point of view on reality, it does not provide any way to act on the abstracted reality. Conversely, virtualization neither aims to generalize nor to simplify: it aims to create a purpose-oriented virtual reality with action capability. In systems biology, the major aims for virtualization deal with heterogeneous data, data integration, analysis and simulation.

In these perspectives, our goal is to develop a method for virtualizing systems biology in any dimension of such systems i.e., data, process, experimental methodologies, modeling, etc. Part of the method, using SB-UML, will take advantage of the many efforts for translating UML diagrams into formal models suitable to carry out analysis on firm grounds [29]. Such transformations have different applications which may concern model checking to verify the global consistency, property verification at a low level of detail, simulation and prediction of properties, etc. Numerous works are ongoing in the fields of systems biology and we aim virtualization would help to integrate although preserving these different and complementary contributions.

Acknowledgments. We thank Nicolas Caritey, Laurent Gaubert and Bénédicte Le Grand for helpful discussions.

References

1. Spring, M. B., Jennings, M. C.: Virtual reality and abstract data: virtualizing information. Virtual Reality World. **1** (1) (1993), pp. c-m.
2. Moore, R. W.: Integrating Data and Information Management. International Supercomputer Conference, June 22-25 (2004), Heidelberg (D).
3. Auffray, C., Imbeau, S., Roux-Rouquié, M., Hood, L. (2003) C. R. Biologies. From functional genomics to systems biology. 326, 879-892.
4. http://www.omg.org
5. Franza, B. R.: From play to laws: Language in Biology. Sci STKE, pe9 (2004)
6. Sauro, H., Hucka, M., Finney, A., Wellock, C., Bolouri, H., Doyle, J., Kitano,H.: Omics: A journal of integrative biology. **7** (2003) 355-372
7. Garvey, TD., Lincoln, P., Pedersen, CJ., Martin, D., Johnson, M.: Omics: A journal of integrative biology. **7** (2003) 411-420
8. http://www.cellml.org/public/specification/20021106/index.html
9. Cook, D. L., Farley, J. F., Tapscott, S. J.: A basis for a visual language for describing, archiving and analyzing functional models of complex biological systems Genome Biology **2**(4) (2001) research0012.1-0012.10
10. Kohn, K.W.: Molecular interaction map of the mammalian cell cycle control and DNA repair systems. Mol Biol Cell **10** (8) (1999) 2703-34. 3.
11. Pirson, I., Fortemaison, N., jacobs, C., Dremier, S., Dumont, J., Maenhaut, C. The visual display of regulatory information and networks. Trends Cell Biol **10**(10) (2000):404-408.
12. Maimon, R., Browning, S.: Diagramatic Notation and Computational Structure of Gene Networks. In: The Second International Conference on Systems Biology (2001). Pasadena.
13. Kitano H.: A Graphical Notation for Biological Networks BIOSILICO **1** (2003) 169-176.
14. Schulze-Kremer, S.: Ontologies for Molecular Biology, in Proc of 3rd Pacific Symposium on Biocomputing PSB98 (1998) 693-704.
15. 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.: Gene ontology: tool for the unification of biology. The Gene Ontology Consortium, Nat Genet. **25** (2000) 25-29
16. Rzhetsky, A., Koike, T., Kalachikov, S., Gomez, SM., Krauthammer, M., Kaplan, SH., Kra, P., Russo, JJ., Friedman, C.: A knowledge model for analysis and simulation of regulatory networks. Bioinformatics **16** (12) (2000) 1120-1128
17. Paton, RC.: Diagrammatic Representations for Modelling Biological Knowledge. BioSystems **66** (2002) 43-53
18. Roux-Rouquié, M., Le Moigne., JL.: The systemic paradigm and its relevance for modeling biological functions. C. R. Biologies, Special Issue : Model driven Acquisition **325** (2002) 419-430
19. Soto., M.: Semantic approach of virtual worlds interoperability. In: Michael Capps (ed.): Proceedings of IEEE WET-ICE '97, Cambridge, MA, June 1997. IEEE Press.
20. Regev, A., Shapiro, E.: Cells as computation. Nature **419** (6905) (2002) 343

21. Roux-Rouquié, M., Renner, J., Sautejeau, G., Rosenthal-Sabroux, C.: Modeling Systems and Processes in Molecular Biology with active objects. In: Objects in bio- and chem-informatics (OiBCI02), OMG conference, Washington, USA (2002)
22. Roux-Rouquié, M., Caritey, N., Gaubert, L., Rosenthal-Sabroux, C.: Using the Unified Modeling Language (UML) to guide systemic description of biological processes and systems Biosystems (2004), in press.
23. Kam, N., Irun, R., Cohen, Harel, D.: The Immune System as a Reactive System: Modeling T Cell Activation With Statecharts. In: IEEE 2001 Symposia on Human Centric Computing Languages and Environments (HCC'01) Stresa, Italy, (2001) september 05-07
24. Spellman, Miller, PTM., Troup, C., Sarkans, U., Chevitz, S., Berhnart, D., Sherlock, G., Ball, C., Lepage, M., Swiatek, M., Marks, WL., Goncalves, J., Markel, S., Iordan, D., Shojatalab, M., Pizarro, A., White, J., Hubley, R., Deutsch, E., Senger, M., Aronow, BJ., Robinson, A., Bassett, D., Stoeckert, CJ. Jr., Brazma, A.: Genome Biology **3** (9) (2002) research0046.1_0046.9
25. G D Bader and C W V Hogue. BIND-a data specification for storing and describing biomolecular interactions, molecular complexes and pathways (2000) Bioinformatics 16, 465-477.
26. Karp, P.D., Riley, M., Paley, S.M., Pellegrini-Toole, A., Krummenacker, M.: EcoCyc: Encyclopedia of E. Coli Genes and Metabolism, Nucleic Acid Res. **27** (1999) 55-58.
27. Hodges, P.E., McKee, A.H.Z., David, B.P., Payne, W.E., Garrels, J.I.: The Yeast Proteome Database (YPD): a Model for the Organization and Presentation of Genome-Wide Funtional Dat, Nucleic Acid Res, **27** (1999) 69-73.
28. Kaneshisa, M.: A Database for Post-Genome Analysis, Trends genet. **13** (1997) 375-376.
29. Korenblat, K., Priami, C.: Extraction of Pi-calculus specifications from a UML sequence and state diagrams. DEGAS IST-2001-32072, technical report (2003) #DIT-03-07.

Genome Size and Numbers of Biological Functions

Ernest Feytmans[1], Denis Noble[2], and Manuel C. Peitsch[3]

[1] Swiss Institute of Bioinformatics, Switzerland*
ernest.feytmans@isb-sib.ch
[2] University Laboratory of Physiology, Parks Road, OX1 3PT, UK
denis.noble@physiol.ox.ac.uk
[3] Novartis Institutes for Biomedical Research, 4002, Basel, Switzerland
manuel.peitsch@pharma.novartis.com

Abstract. Calculations of potential numbers of interactions between gene products to generate physiological functions show that we can expect a highly non-linear relation between genome size and functional complexity. Moreover, very small differences in gene numbers or sequence can translate into very large differences in functionality.

1 Introduction

With the sequencing of the genomes of a substantial number of different species including the human [1-3] it has become possible to compare genomes in terms of their overall size, their total numbers of genes, and their degree of similarity. Two features have emerged that have generated frequent comment and analysis.

The first is that the genomes of higher (complex) organisms are not very large compared to those of lower (simpler) organisms. For example, the human genome may have about 1.5 times the number of genes of a worm, *c. elegans* [4]. Complexity, however defined, does not seem to scale linearly with genome size or number [see also 5 page 280].

The second is that the differences between the genomes of, for example, a monkey and a human turn out to be very small. So small, in fact, that the difference in functional capability must be represented by less than a 2% difference in genome sequence. These observations raise important questions concerning how biological complexity arises, how it is coded and how many genes are necessary for certain degrees of complexity in function.

We are far from having answers to these fundamental questions. Nevertheless, even with the present information it is possible to do some simple calculations that demonstrate why there is, in principle, a much larger scope for biological complexity arising from, say, 30,000, genes than may initially be apparent, and why even small differences at the level of the genome, perhaps 1 or 2%, can translate into immense differences at a functional level. Furthermore, the *Mycoplasma genitalium* genome is composed of 517 genes, expressing 480 proteins of which about 300 are thought to be essential under laboratory growth conditions [6]. Similar calculations can be performed to evaluate the level of biological complexity in what is considered the minimal genome for an independently replicating cell.

* The authors are listed alphabetically.

C. Priami et al. (Eds.): Transactions on Computational Systems Biology, LNBI 3380, pp. 44 – 49, 2005.

2 Results

The basis of the calculations presented here is that gene products (proteins) act in combination to generate biological functions. For example, to simulate most of the metabolism of *E. coli* it suffices to model about 120 gene products [7-9]. To model pacemaker activity in the pacemaker cells of the heart, or how these cells handle calcium signalling, even fewer protein components are sufficient [10, 11]. In some cases, at least, nature seems to be highly modular. We can therefore ask a simple question. In a genome of size *n*, with *r* genes required to code for a single biological function, how many possible distinct combinations could there be that are available for functional translation?

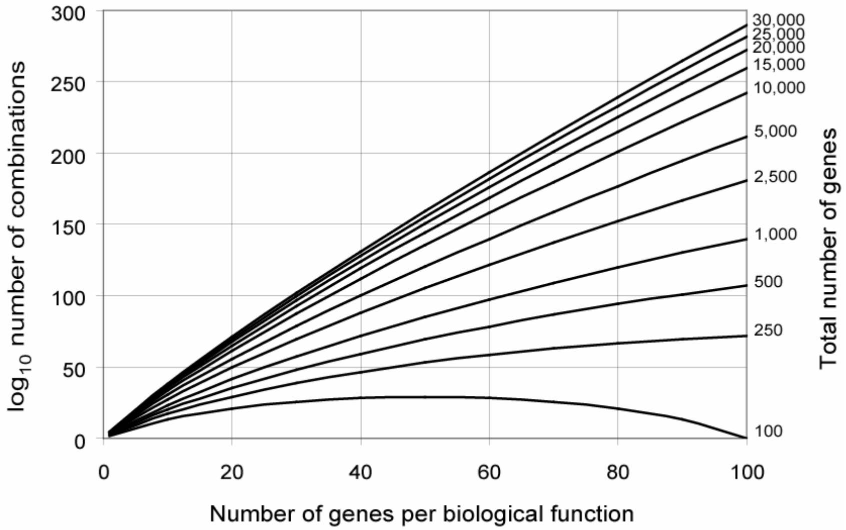


Fig. 1. Ordinate: Number of potential biological functions. Abscissa: Number of genes required to define each biological function. The curves show results for genomes of various sizes between 100 and 30,000

We consider the number of combinations of *r* objects taken out of *n* objects. Then

$$nPr = n(n\text{-}1)(n\text{-}2) \ldots\ldots (n\text{-}r\text{+}1) = n\,!/(n\text{-}r)\,! \qquad (1)$$

where, for a human genome, *n* is taken to be 30,000 (though for illustrative purposes, we will show the results for smaller numbers of genes) and *r* is variable according to how many gene combinations may underlie physiological functions.

Figure 1 shows the results for various overall numbers of genes *n* between 100 and 30,000, as *r* (the number of genes per function) increases from 0 to 100. The lowest curve (for 100 genes) is included largely for illustration of the principles involved. Clearly, in a genome of 100 genes, if 100 were required per function, then there would be only one function. So, the curve rises to a peak and then falls to 1. This result would be obtained for all genome sizes when investigating the total number of functions as the number of genes per function rises towards the total number of genes available.

The other curves show the results for increasing total numbers of genes, when the maximum number of genes involved in a function rises to 100. The result of interest here is that for 30,000 genes (estimated human genome size), the total number of functions that could be defined by 100 genes per function is truly enormous: approaching 1 x 10^{300}. In comparison, the curve for 500 genes approximates the level of complexity of the minimal genome defined for *M. genitalium.*

Note also that the total number of possible functions increases much more rapidly than the total number of genes. Compare, for example, 500 genes with 5000. At 100 genes per function, we have more than 1 x 10^{100} possible functions for 500 genes, and over 1 x 10^{200} for 5000 genes. Thus increasing the gene number by one order of magnitude leads to multiplying the available combinations by a further factor of 10^{100}: one order of magnitude in gene numbers generates 100 orders of magnitude in potential functions. The dependence of potential functions on number of genes is therefore highly non-linear.

The significance of this result becomes even clearer when we compare genomes of similar sizes or of great similarity in sequence. We have done this by investigating the effect of adding one gene to the genome, or of adding one gene for the biological function.

Table 1 shows how surprising the results of these calculations can be. If we add just one gene to a genome of 30,000 genes, the number of possible new functions is about 10^{287}. Conversely if a function requires just one more gene, then the number of newly created combinations would be about 10^{292}.

If we compare the number of functions potentially generated by the *M. genitalium* genome with those of the human genome, we see that the difference in complexity is almost equal to the human complexity itself (10^{289}-$10^{81} \approx 10^{289}$).

Table 1. Number of potential biological functions. Column D1 represents the number of potential new functions obtained by adding one gene to a genome of size 30,000, while line D2 shows the effect of increasing by one the number of genes required for a biological function

		Number of Genes in the genome		
		30000	**30001**	**D1**
Number of genes required for a biological function	**100**	4.6815x10^{289}	4.6971x10^{289}	1.5657x10^{287}
	101	1.3859x10^{292}	1.3906x10^{292}	4.6815x10^{289}
	D2	1.3812x10^{292}	1.3859x10^{292}	

Table 2. Number of potential biological functions for the human and *M. genitalium* genomes. Column D1 represents the number of potential new functions that can be obtained by increasing the size of the genome by two orders of magnitude, while line D2 shows the effect of increasing by one order of magnitude the number of genes required for a biological function. The number of potential new functions appearing in the human genome when compared to *M. genitalium* is shown in D3

		Number of Genes in the genome			
		300	**30000**	**D1**	**D3**
Number of genes required for a biological function	**10**	1.3983x10^{18}	1.6248x10^{38}	1.6248x10^{38}	
	100	4.1583x10^{81}	4.6815x10^{289}	4.6815x10^{289}	
	D2	4.1583x10^{81}	4.6815x10^{289}		
	D3				4.6815x10^{289}

3 Discussion

The calculations presented here are very simple. Equation (1) would have been known even to 19th century gamblers! But they reveal not only the expected high degree of non-linearity between gene numbers and functional possibilities, but also a surprisingly large effect of relatively small differences between genomes and of tiny changes to a genome.

It is easy to understand the basis of this effect. If we add one gene to a genome, and because the mathematics is that of a geometric progression, we will still have all the possible functions for the smaller genome, and all of those again in combination with the added gene.

We have investigated many variations of these calculations. The conclusions are valid for almost any values of *n* and *r*. Particularly if genomes are already fairly large, say greater than 5000, truly enormous differences in functional possibility can be coded by very tiny differences either in total number of genes, number of genes per function, or changes in sequence.

While these numbers have no realistic physical meaning, we can compare them with the number of possible protein sequences of let's say 100 residues, which is 20^{100} or 1.3x10^{130}, and shows us that nature has indeed been forced to select a infinitesimal sub-ensemble of the possible outcomes to create life. As these numbers surpass the total number of atoms in the Universe (estimated to be 10^{80}. [12]), it is impossible that the combinations have been and will be all tested. In fact, the numbers are so large that we are inclined to invert the usual question that is asked when genomes of great similarity are compared. The question should not be "how can this small difference possibly code for all the functional differences between the species, or for the increased complexity?" but rather "how many of these immensely large potential differences does nature actually use and how are they chosen?" Surely, nature has vastly more possibilities than are actually manifested in existing species. And that is what we should expect. The chemistry of protein-protein interaction will impose limits on how many other proteins, metabolites and signalling molecules a given protein can interact with. Some are hubs occupying central positions within networks; others are relatively unreactive, occupying peripheral positions. Compartmentation of proteins within cells and organs is also a large limiting factor. These will be factors limiting the number of interactions within a given organism. In a study [9] of metabolic networks in E. coli, for example, estimates of the total number of possible networks within *E Coli* metabolism were found to be somewhere around 4.4 x 10^{21}, while actual used circuits are around 500,000. So the ratio in this case for 'possible' to 'actual' is around 10^{16}. What can however not be estimated with our current knowledge, is the number of new combinations, generated by the addition of a single gene or gene variant, that are accessible within the constraints of the evolutionary choices already taken. Even if this number is infinitesimally small compared to the total number of combinations, it certainly remains very large and accounts for the intra and inter-species diversity.

On the other hand, during evolution nature has had even more possibilities available than are indicated by our calculations, which are based on each gene being a single fixed entity. In reality, many different isoforms are available. Our calculations also ignore the influence of splice variants and post-transcriptional regulation. Most genes have more than one exon so that there are 3 or more possible splice variants. One gene (Dscam) in *Drosophila melanogaster* has been shown to have as many as 115 exons with 38,000 putative protein products. Moreover, during development, some of the variants are regulated by more than an order of magnitude (from 1% to 40% between birth and adulthood [13]). This will further increase the estimated complexity. For instance, if we predict that each gene has just two variants on average then, according to eq. 1, 100 genes can produce 4950 potential functions.

We also ignore the effects of systematic variations in gene expression levels. Thus, in the heart, such regional variations are critical for defining the differences between cells in different regions with different functions (such as pacemaker activity, rapid conduction and contractility). Even within the wall of the ventricle there are substantial variations that are important for explaining the form of the electrocardiogram [14] and, possibly, for preventing arrhythmia.

Finally, the non-linearity of combinatorial effects may have important implications for drug development. Very few drugs act on a single gene product. The great majority have multiple effects, many of which are unwanted side-effects. But given the combinatorial nature of biological functions there must be some combinations that are more beneficial than a single action compound. These will be those combinations of actions that most closely mimic natural biological combinations. Similar considerations to those presented in this paper show that, for just a few hundred drug binding sites, there will be immensely large numbers of possibly active combinations. For instance, if we consider a receptor family of 500 members, (same order of complexity as the GPCR family), and provided that a drug binds to only two of them, there will be over 100,000 (strictly 124750) possible pairs of receptors susceptible to binding. Most drugs, however, act on more receptors albeit with varying affinities, dramatically increasing the complexity of drug action. The results in Figure 1, considered in this context, show that the potential number of combinations for 500 sites can easily rise to the order of 10^{100}. As with gene combinations, the great majority of these potential combinations will be either harmful or chemically forbidden. This supports the need for pathway and function-based approaches to drug discovery.

One of the great challenges for systems and computational biology will be to identify the logic of the tiny proportion of functional gene interactions that are successful, in order to narrow down the immense potential numbers to more manageable proportions. Nature has had a vast canvas on which to evolve the species we have today.

Acknowledgements. DN is supported by British Heart Foundation, Medical Research Council and The Wellcome Trust.

References

1. International Human Genome Mapping Consortium, *A physical map of the human genome.* Nature, 2001. **409**: p. 934-941.
2. Venter, C., et al, *The sequence of the human genome.* Science, 2001. **291**: p. 1304-1351.
3. Collins, F.S., et al., *A vision for the future of genomics research.* Nature, 2003. **422**: p. 835--847.
4. Chervitz, S.A., et al., *Comparison of the Complete Protein Sets of Worm and Yeast: Orthology and Divergence.* Science, 1998. **282**: p. 2022-2028.
5. Sulston, J. and G. Ferry, *The Common Thread.* 2002, London: Bantam Press.
6. Hutchison, C.A., et al., *Global Transposon Mutagenesis and a Minimal Mycoplasma Genome.* Science, 1999. **286**: p. 2165-2169.
7. Palsson, B.O., *The challenges of in silico biology.* Nature Biotechnology, 2000. **18**: p. 1147-1150.
8. Edwards, J.S. and B.O. Palsson, *In Silico predictions of Escherichia coli metabolic capabilities are consistent with experimental data.* Nature Biotechnology, 2001. **19**(2): p. 125-130.
9. Stelling, J., et al., *Metabolic network structure determines key aspects of functionality and regulation.* Nature, 2002. **420**: p. 190-193.
10. Garny, A., et al., *Comparative study of sino-atrial node cell models.* Chaos, Solitons and Fractals, 2002. **13**: p. 1623-1630.
11. Winslow, R.L., J.J. Rice, and S. Jafri, *Modeling the cellular basis of altered excitation-contraction coupling in heart failure.* Progress in Biophysics and Molecular Biology, 1998. **69**: p. 497-514.
12. Magnan, C., *personal communication.* 2003.
13. Celotto, A.M. and B.R. Graveley, *Alternative splicing of the Drosophila Dscam pre-mRNA is both temporally and spatially regulated.* Genetics, 2001. **159**: p. 599-608.
14. Antzelevitch, C., et al., *Influence of transmural gradients on the electrophysiology and pharmacology of ventricular myocardium. Cellular basis for the Brugada and long-QT syndromes.* Philosophical Transactions of the Royal Society, series A, 2001. **359**: p. 1201-1216.

Operational Patterns in Beta-Binders*

Corrado Priami and Paola Quaglia

Dipartimento di Informatica e Telecomunicazioni, Università di Trento, Italy

Abstract. As a preliminary step in testing the expressiveness of Beta-binders against realistic case studies, we comment on a number of operational properties of the formalism and present a set of derived patterns that can be useful when modeling complex biosystems.

1 Introduction

The post-genomics era offers accurate descriptions of the fundamental components of living systems, and in particular of proteins and cells. The understanding of the functional interactions of individual components when gathered in complex systems is however far from complete. This poses new challenges to computer scientists working in bioinformatics.

Till very recently, the main contributions to life sciences from the research community in computer science and information technology have been centered around databases and algorithms to organize and compare static information. Biologists now turn their attention to the investigation of networks of hundreds of functionally distinct components, and are interested in discovering their possible (chemical) interactions. Computational methods able to support modeling and simulation of the dynamics of biological systems can aid this effort. Hopefully, those methods will serve as foundational models to mechanized tools for 'in silico' predictive research on the behaviour of complex systems. The tools on their side could ease the analysis of the response to artificial perturbations, or help uncovering evolutionary behaviours.

The interactions among genes, molecules, and biological entities in general are regulated by principles analogous to those typically used to describe distributed and mobile systems. Interactions depend on (chemical) messages, they are localized at specific sites, and can change the future behaviour of the global system.

Given the above analogies, a number of *ad hoc* process calculi and the like have been proposed by the research community in concurrency theory (see, e.g., [9, 11, 3, 10, 1, 5, 8]) as a response to the need of modeling the dynamics of biological systems. Generically speaking, these formalisms provide the means to specify and reason on protein interactions and complex protein pathways. Each of them, however, has been conceived to cope with some specific modelization

* This work has been partially supported by the FIRB project "Modelli formali per Sistemi Biochimici".

C. Priami et al. (Eds.): Transactions on Computational Systems Biology, LNBI 3380, pp. 50–65, 2005.

concern, and it is yet not clear which of them (if any) can be considered as a general model to formally reason on biology.

The biochemical stochastic π-calculus [9, 11] is a variant of the π-calculus [7, 12] that enriches the actions with basal rates and allows stochastic reasoning on process behaviours. CCS-R [3] offers the explicit machinery, on a CCS [6] ground, to model reversible molecular reactions. Formal Molecular Biology [5] is a language supported by specialized graph-rewriting techniques to illustrate protein-to-protein interactions and bindings. Bio-ambients [10], Brane calculi [1], and Beta-binders [8] share a common design principle: each of them provides the means to model enclosing surfaces of entities and hence allows easy representation of the relative locations of components (think, e.g., of the position of the nucleus in a cell, of a cell in a tissue, etc.). The three formalisms largely differ, however, on the semantic role associated with surfaces.

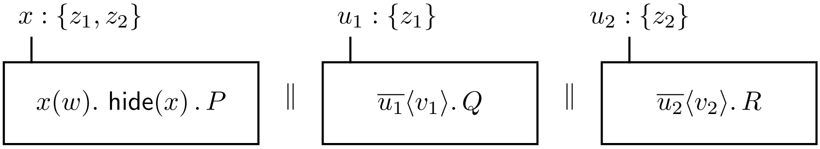
First, Bio-ambients and Brane calculi allow nested wrapping of processes and then permit the specification of hierarchical objects. Beta-binders disregards nesting (at least in its explicit form, see below for more details on this point).

Second, Brane calculi, a calculus of membranes, is strongly specialized in the representation of the dynamic evolutions of surfaces. In this respect, both Bio-ambients and Beta-binders have a more general-purpose flavor: boxes can represent any sort of limited biological environment, from molecules to tissues.

Last, and more interestingly, the possible activities of the enclosing surfaces are totally different in Bio-ambients, Brane calculi, and Beta-binders. Surfaces are just wrappers in Bio-ambients: processes can exit or enter a box, and boxes can join or split, but in any case the box is passive and any movement of the system is driven from the inside of boxes. Brane calculi, a calculus of *membranes*, takes exactly the opposite point of view. Borders, i.e. membranes, are themselves specialized coordinators which can perform a number of distinguished actions (e.g. change the orientation of the membrane). W.r.t. the role played by surfaces, Beta-binders is somewhere in the middle between Bio-ambients and Brane calculi. Borders are equipped with typed sites that are used to discriminate the kind of interactions boxes may be involved in. Nonetheless the evolution of boxes is partially driven from the inside (see below, e.g., the formalization of interactions vs the merge of boxes).

Summing up, each of the process algebra based formalisms defined to model the dynamics of biological systems tackles slightly different phenomena in a mildly different perspective. As we mentioned, it is not clear yet if (and to which extent) one of them can be set as the basis for formal reasoning about biological dynamics. To acquire confidence in the expressiveness of the above formalisms it is at least necessary to test them against realistic case studies. In this paper we work towards this direction for Beta-binders. Specifically, we comment on a number of operational properties of the formalism and present a set of derived patterns that can be useful when modeling complex biosystems.

Beta-binders is strongly inspired by the π-calculus, and hence takes communication as the primitive form of interaction between entities running (i.e. living) in parallel. As in the π-calculus the basic ingredient of communication is the synchronization of input and of output actions on named channel (in the π-calculus ‘names’ are synonyms of ‘channels’). Differently from the π-calculus, Beta-binders provide the means to model the enclosing surfaces of entities and possible interactions taking place at the level of the virtual surfaces. This is due to a special class of binders which are used to wrap (quasi-) π-calculus processes into boxes with interaction capabilities. More precisely, boxes represent the borders of biological entities and are equipped with typed interaction sites (receptors). The graphical representation of a simple process is shown below.


$x : \{z_1, z_2\}$
$x(w).\,\mathsf{hide}(x)\,.\,P$
$\|$
$u_1 : \{z_1\}$
$\overline{u_1}\langle v_1\rangle.\,Q$
$\|$
$u_2 : \{z_2\}$
$\overline{u_2}\langle v_2\rangle.\,R$

Such a process, textually written

$$\begin{aligned}&\beta(x : \{z_1, z_2\})\,\big[x(w).\,\mathsf{hide}(x)\,.\,P\big] \parallel\\ &\beta(u_1 : \{z_1\})\,\big[\overline{u_1}\langle v_1\rangle.\,Q\big] \parallel\\ &\beta(u_2 : \{z_2\})\,\big[\overline{u_2}\langle v_2\rangle.\,R\big]\end{aligned}$$

shows three parallel components with interaction sites $x : \{z_1, z_2\}$, $u_1 : \{z_1\}$, and $u_2 : \{z_2\}$, respectively. The set $\{z_1, z_2\}$, acting as the type of the site named x, denotes that the leftmost box can potentially interact with both the middle box and the rightmost one. This is ensured by the fact that these latest boxes exhibit sites whose types ($\{z_1\}$ and $\{z_2\}$, resp.) have non empty intersection with $\{z_1, z_2\}$. The actual interaction between boxes consumes the input prefix $x(w)$ on one side, and the output prefix in the other box ($\overline{u_1}\langle v_1\rangle$ or $\overline{u_2}\langle v_2\rangle$, non deterministically). After interaction, the site x of the leftmost box is hidden (by consuming the prefix $\mathsf{hide}(x)$), and the box gets isolated from the external environment. As a final remark about the above example, notice that we are assuming here the simplest form of types for sites (sets of names) and the simplest relationship between types to compute their compatibility in interactions (non empty intersection of sets). Different and more complex forms of types (based, e.g., on XML patterns) and relationships between them can be envisaged without altering the results presented in this work.

More generally, in Beta-binders the evolution of boxes is described by a limited number of macro-operations: communication between components within the same box (intra-communication); communication between two boxes (inter-communication); addition of a site to a box; hiding and unhiding of an interaction site; joining of two boxes; splitting of a box in two boxes. Adding, hiding and unhiding sites have a fundamental role in modelling the dynamics of box interfaces and hence, e.g., the functional dependency of the interaction capabilities of biological components on their particular shape or folding. The join and split operations, that are related to the evolution of the structure of boxes rather than to the dynamics of their sites or interfaces, are described in a parametric way to accommodate possible distinct instances of the same macro-behaviour. In more detail, the operational axiom for join (split, respectively) depends on a function that checks the conditions under which two boxes can be merged and also contributes to determine the identity of the resulting box (boxes, respectively). Distinct instances of the above functions, and hence distinct instances of the operational axioms for join and split, can live together in the same formal system to allow the modeling of phenomena which are intrinsically analogous but happen to be regulated by different factors.

The rest of the paper is organized as follows. First we present an overview of the formal definition of Beta-binders. Then a number of operational patterns are commented on in Section 3. In particular, we deal with: possible modeling of endocytosis, meiosis, and exocytosis; the irrelevance of adding a top level replication operator; the interplay between join and split of boxes; the intrinsically different nature of inter-actions and intra-actions; the directionality of the operations over the structure of boxes (join and split); the simulation of hierarchical nesting of boxes; and the dynamic change of the type of sites. Eventually, Section 4 concludes the paper with some final remarks.

2 Beta-Binders

This section presents an overview of Beta-binders. The reader will benefit from some familiarity with the π-calculus.

The π-calculus is a process calculus where names are the media and the values of communication. The same point of view is taken for Beta-binders, where we assume the existence of a countably infinite set N of names (ranged over by lowercase letters). Beta-binders allows the description of the behaviour of π-calculus processes wrapped into boxes with interaction capabilities (hereafter called *beta-processes* or simply *boxes*). Processes encapsulated into boxes (ranged over by capital letters distinct from B) are given by the following syntax.

$$P ::= \mathsf{nil} \;\Big|\; x(w).\,P \;\Big|\; \overline{x}\langle y\rangle.\,P \;\Big|\; P \mid P \;\Big|\; \nu y\, P \;\Big|\; !\,P \;\Big|$$
$$\mathsf{expose}(x,\, \Gamma)\,.\,P \;\Big|\; \mathsf{hide}(x)\,.\,P \;\Big|\; \mathsf{unhide}(x)\,.\,P$$

For simplicity, despite the difference w.r.t. the usual π-calculus syntax, we refer to the processes generated by the above grammar as to pi-processes. The deadlocked process nil, input and output prefixes ($x(w).\,P$ and $\overline{x}\langle y\rangle.\,P$, respectively), parallel composition ($P \mid P$), restriction ($\nu y\, P$), and the bang operator (!) have exactly the same meaning as in the π-calculus.

The expose, hide, and unhide prefixes are intended for changing the external interface of boxes by adding a new site, hiding a site, and unhiding a site which has been previously hidden, respectively.

The π-calculus definitions of *name substitution* and of *free* and *bound names* (denoted by $\mathsf{fn}(\text{-})$ and $\mathsf{bn}(\text{-})$, respectively) are extended to the processes generated by the above syntax in the obvious way. It is sufficient to state that neither $\mathsf{hide}(x)$ nor $\mathsf{unhide}(x)$ act as binders for x, while the prefix $\mathsf{expose}(x, \varGamma)$ in $\mathsf{expose}(x, \varGamma)\,.\,P$ is a binder for x in P.

Beta-processes are defined as pi-processes prefixed by specialized binders that suggest the name of the formalism and are defined as follows.

Definition 1. *An* elementary beta binder *has either the form $\beta(x : \varGamma)$ or the form $\beta^h(x : \varGamma)$, where*

1. *the name x is the* subject *of the beta binder, and*
2. *$\varGamma$ is the* type *of x. It is a non-empty set of names such that $x \notin \varGamma$.*

Intuitively, the elementary beta binder $\beta(x : \varGamma)$ is used to denote an active (potentially interacting) site of the box. Binders like $\beta^h(x : \varGamma)$ denote sites which have been hidden to forbid further interactions through them.

Definition 2. Composite beta binders *are generated by the following grammar:*

$$\boldsymbol{B} ::= \beta(x : \varGamma) \;\Big|\; \beta^h(x : \varGamma) \;\Big|\; \beta(x : \varGamma)\,\boldsymbol{B} \;\Big|\; \beta^h(x : \varGamma)\,\boldsymbol{B}$$

A composite beta binder is said to be well-formed *when the subjects of its elementary components are all distinct. We let well-formed beta binders be ranged over by $\boldsymbol{B}, \boldsymbol{B}_1, \boldsymbol{B}_2, \ldots, \boldsymbol{B}', \ldots$.*

The set of the subjects of all the elementary beta binders in $\boldsymbol{B}$ is denoted by sub$(\boldsymbol{B})$*, and we write $\boldsymbol{B} = \boldsymbol{B}_1\boldsymbol{B}_2$ to mean that $\boldsymbol{B}$ is the beta binder given by the juxtaposition of $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$.*

Also, the metavariables $\boldsymbol{B}^*, \boldsymbol{B}^*_1, \boldsymbol{B}^*_2, \ldots$ stay for either a well-formed beta binder or the empty string. The above notation for the subject function and for juxtaposition is extended to these metavariables in the natural way.

Beta-processes (ranged over by $B, B_1, \ldots, B', \ldots$) are generated by the following grammar:

$$B ::= \mathsf{Nil} \;\Big|\; \boldsymbol{B}[P] \;\Big|\; B \parallel B$$

Nil denotes the deadlocked box and is the neutral element of the parallel composition of beta-processes, written $B \parallel B$. But for Nil, the simplest form of beta-process is given by a pi-process encapsulated into a beta binder ($\boldsymbol{B}[P]$). Notice that nesting of boxes is not allowed.

To any beta-process consisting of n parallel components corresponds a simple graphical notation, given by n distinct boxes, one per parallel component. Each box contains a pi-process and has as many sites (hidden or not) as the number of elementary beta binders in the composite binder. The relative position of sites along the perimeter of the box is irrelevant, just as the relative positions of parallel boxes in the two-dimensional space.

Beta-processes are given an operational reduction semantics that makes use of both a structural congruence over beta-processes and a structural congruence over pi-processes. We overload the same symbol to denote both congruences, and let the context disambiguate the intended relation.

Definition 3. Structural congruence *over pi-processes, denoted* $\equiv$*, is the smallest relation which satisfies the following laws.*

- $P_1 \equiv P_2$ *provided* P_1 *is an* α*-converse of* P_2
- $P_1 \mid (P_2 \mid P_3) \equiv (P_1 \mid P_2) \mid P_3$, $P_1 \mid P_2 \equiv P_2 \mid P_1$, $P \mid \mathsf{nil} \equiv P$
- $\nu z\, \nu w\, P \equiv \nu w\, \nu z\, P$, $\nu z\ \mathsf{nil} \equiv \mathsf{nil}$, $\nu y\, (P_1 \mid P_2) \equiv P_1 \mid \nu y\, P_2$ *provided* $y\ /\in$ $\mathsf{fn}(P_1)$
- $!\,P \equiv P \mid !\,P$.

Structural congruence *over beta-processes, denoted* $\equiv$*, is the smallest relation which satisfies the laws listed below, where* $\hat{\beta}$ *is intended to range over* $\{\beta, \beta^h\}$.

- $\boldsymbol{B}[P_1] \equiv \boldsymbol{B}[P_2]$ *provided* $P_1 \equiv P_2$
- $B_1 \parallel (B_2 \parallel B_3) \equiv (B_1 \parallel B_2) \parallel B_3$, $B_1 \parallel B_2 \equiv B_2 \parallel B_1$, $B \parallel \mathsf{Nil} \equiv B$
- $\boldsymbol{B}_1\boldsymbol{B}_2[P] \equiv \boldsymbol{B}_2\boldsymbol{B}_1[P]$
- $\boldsymbol{B}^*\hat{\beta}(x : \Gamma)[P] \equiv \boldsymbol{B}^*\hat{\beta}(y : \Gamma)[P\{y/x\}]$ *provided* y *fresh in* P *and* $y \notin \mathsf{sub}(\boldsymbol{B}^*)$.

The laws of structural congruence over pi-processes are the typical axioms to formalize structural congruence in the π-calculus. The laws over beta-processes state, respectively, that (i) the structural congruence of pi-processes is reflected at the upper level as congruence of boxes; (ii) the parallel composition of beta-processes is a monoidal operation with neutral element Nil; (iii) the actual ordering of elementary beta binders within a composite binder is irrelevant; (iv) the subject of elementary beta binders is a placeholder that can be changed at any time under the proviso that name clashes are avoided and well-formedness of beta binder is preserved.

The *reduction relation*, $\longrightarrow$, is the smallest relation over beta-processes obtained by applying the axioms and rules in Table 1.

The reduction relation describes the evolution within boxes (intra), as well as the interaction between boxes (inter), the dynamics of box interfaces (expose, hide, unhide), and the structural modification of boxes (join, split).

The rule intra lifts to the level of beta-processes any 'reduction' of the enclosed pi-process. Notice indeed that no reduction relation is defined over pi-processes.

The rule inter models interactions between boxes with complementary internal actions (input/output) over complementary sites (sites with non-disjoint types). Information flows from the box containing the pi-process which exhibits the output prefix to the box enclosing the pi-process which is ready to perform the input action.

The rules expose, hide, and unhide correspond to an unguarded occurrence of the homonymous prefix in the internal pi-process and allow the dynamic modification of external interfaces.

The rule expose causes the addition of an extra site with the declared type. The name x used in expose(x, Γ) is a placeholder which can be renamed to meet the requirement of well-formedness of the enclosing beta binder.

The rules hide and unhide force the specified site to become hidden and unhidden, respectively. They cannot be applied if the site does not occur unhidden, respectively hidden, in the enclosing beta binder.

Table 1. Axioms and rules for the reduction relation

(intra)
$$\frac{P \equiv \nu\tilde{u}\,(x(w).\,P_1 \mid \overline{x}\langle z\rangle.\,P_2 \mid P_3)}{\boldsymbol{B}\big[P\big] \longrightarrow \boldsymbol{B}\big[\nu\tilde{u}\,(P_1\{{}^{z}\!/_{w}\} \mid P_2 \mid P_3)\big]}$$

(inter)
$$\frac{P \equiv \nu\tilde{u}\,(x(w).\,P_1 \mid P_2) \qquad Q \equiv \nu\tilde{v}\,(\overline{y}\langle z\rangle.\,Q_1 \mid Q_2)}{\beta(x : \mathit{\Gamma})\,\boldsymbol{B}_1^*\big[P\big] \parallel \beta(y : \mathit{\Delta})\,\boldsymbol{B}_2^*\big[Q\big] \longrightarrow \beta(x : \mathit{\Gamma})\,\boldsymbol{B}_1^*\big[P'\big] \parallel \beta(y : \mathit{\Delta})\,\boldsymbol{B}_2^*\big[Q'\big]}$$

where $P' = \nu\tilde{u}\,(P_1\{{}^{z}\!/_{w}\} \mid P_2)$ and $Q' = \nu\tilde{v}\,(Q_1 \mid Q_2)$

provided $\mathit{\Gamma} \cap \mathit{\Delta} \neq \emptyset$ and $x, z \notin \tilde{u}$ and $y, z \notin \tilde{v}$

(expose)
$$\frac{P \equiv \nu\tilde{u}\,(\mathsf{expose}(x,\ \mathit{\Gamma})\,.\,P_1 \mid P_2)}{\boldsymbol{B}\big[P\big] \longrightarrow \boldsymbol{B}\,\beta(y : \mathit{\Gamma})\,\big[\nu\tilde{u}\,(P_1\{{}^{y}\!/_{x}\} \mid P_2)\big]}$$
provided $y \notin \tilde{u}$, $y \notin \mathsf{sub}(\boldsymbol{B})$ and $y \notin \mathit{\Gamma}$

(hide)
$$\frac{P \equiv \nu\tilde{u}\,(\mathsf{hide}(x)\,.\,P_1 \mid P_2)}{\beta(x : \mathit{\Gamma})\,\boldsymbol{B}^*\big[P\big] \longrightarrow \beta^h(x : \mathit{\Gamma})\,\boldsymbol{B}^*\big[\nu\tilde{u}\,(P_1 \mid P_2)\big]}$$
provided $x \notin \tilde{u}$

(unhide)
$$\frac{P \equiv \nu\tilde{u}\,(\mathsf{unhide}(x)\,.\,P_1 \mid P_2)}{\beta^h(x : \mathit{\Gamma})\,\boldsymbol{B}^*\big[P\big] \longrightarrow \beta(x : \mathit{\Gamma})\,\boldsymbol{B}^*\big[\nu\tilde{u}\,(P_1 \mid P_2)\big]}$$
provided $x \notin \tilde{u}$

(join)
$$\boldsymbol{B}_1\big[P_1\big] \parallel \boldsymbol{B}_2\big[P_2\big] \longrightarrow \boldsymbol{B}\big[P_1\sigma_1 \mid P_2\sigma_2\big]$$

provided that f_{join} is defined in $(\boldsymbol{B}_1, \boldsymbol{B}_2, P_1, P_2)$ and

with $f_{join}(\boldsymbol{B}_1, \boldsymbol{B}_2, P_1, P_2) = (\boldsymbol{B}, \sigma_1, \sigma_2)$

(split)
$$\boldsymbol{B}\big[P_1 \mid P_2\big] \longrightarrow \boldsymbol{B}_1\big[P_1\sigma_1\big] \parallel \boldsymbol{B}_2\big[P_2\sigma_2\big]$$

provided that f_{split} is defined in $(\boldsymbol{B}, P_1, P_2)$ and

with $f_{split}(\boldsymbol{B}, P_1, P_2) = (\boldsymbol{B}_1, \boldsymbol{B}_2, \sigma_1, \sigma_2)$

(redex)
$$\frac{B \longrightarrow B'}{B \parallel B'' \longrightarrow B' \parallel B''}$$

(struct)
$$\frac{B_1 \equiv B_1' \qquad B_1' \longrightarrow B_2}{B_1 \longrightarrow B_2}$$

The axiom join models the merge of boxes. The rule, being parametric w.r.t. the function f_{join}, is more precisely an axiom schema. The function f_{join} determines the actual interface of the beta-process resulting from the aggregation of boxes, as well as possible renamings of the enclosed pi-processes via the substitutions σ_1 and σ_2. Such a reduction, in a formal context that disallows nesting of boxes, can be used to render biological *endocytosis*, namely the absorption of substances from the external environment. It is intended that as many different instances of f_{join} (and hence of join) can be defined as it is needed to model the system at hand.

The axiom split formalizes the splitting of a box in two parts, each of them taking away a subcomponent of the content of the original box. Analogously to join, the rule split is an axiom schema that depends on the specific definition of the function f_{split}. This function is meant to refine the conditions under which a beta-process can be split in two boxes. With the same care as in the case of endocytosis, split can be used to render *exocytosis*, i.e. the expulsion of biological sub-components from a given compartment. Analogously to the case of the join axiom, many instances of split can live together in the same formal system.

The rules redex and struct are typical rules of reduction semantics. They are meant, respectively, to interpret the reduction of a subcomponent as a reduction of the global system, and to infer a reduction after a proper structural shuffling of the process at hand.

3 Operational Properties and Examples

In this section we present a collection of observations on the Beta-binders operational semantics, and a set of derived patterns that can be useful in modeling complex biosystems.

Hereafter, we assume the following notational conventions. The symbol $\perp$ stays for undefinedness, and the identity substitution is denoted by σ_{id}. As usual $B \longrightarrow^{n} B'$ means that B transforms into B' in $n > 1$ steps, i.e. there exist $B_1, B_2, \ldots$ such that $B \longrightarrow B_1 \longrightarrow B_2 \longrightarrow \ldots \longrightarrow B'$. Also, $B \longrightarrow_{\equiv} B'$ is a short-hand to denote that there exists B_1 such that $B \longrightarrow B_1 \equiv B'$.

3.1 Endocytosis and Meiosis

A first remark on the operational semantics of Beta-binders is about the possibility of modeling biological *endocytosis* and *meiosis*, namely the absorbtion of substances from the external environment, and, respectively, the separation of a cell and of the contained genetic material that is typical of reproductive cells.

Since the formalism disallows nesting of boxes, endocytosis is rendered by augmenting the internal pi-process with a parallel component representing the engulfed substances. At the same time, the external interface can be modified to represent those cases when the absorbed material can still have interactions with the external environment. This effect is obtained by using appropriate instances of f_{join}. Consider for instance the definition below.

$$\begin{aligned} f_{join} = \lambda \boldsymbol{B}_1 \boldsymbol{B}_2 P_1 P_2.\ & \text{if } (\boldsymbol{B}_1 = \beta(x : \mathit{\Gamma})\, \boldsymbol{B}_1^{*} \text{ and } \boldsymbol{B}_2 = \beta(y : \mathit{\Delta})\, \boldsymbol{B}_2^{*} \text{ and} \\ & \qquad \mathit{\Gamma} \cap \mathit{\Delta} \neq \emptyset) \\ & \text{then } (\boldsymbol{B}_1, \sigma_{id}, \{x/y\}) \\ & \text{else } \perp \end{aligned} \tag{1}$$

The specific instance in (1) imposes that the join reduction can take place only if absorbing and absorbed beta-process have complementary sites (elementary beta binders with non disjoint types). It also states that the absorbed process can keep interacting with the external environment through the same site which has been used to engulf it (possible occurrences of input and output actions on channel y in P_2 are renamed by $\sigma_2 = \{^x\!/_y\}$).

Meiosis is directly rendered in Beta-binders by using the split reduction. Similarily to the case of join, the precise hypotheses leading to meiosis have to be tuned by appropriately defining f_{split}.

3.2 Exocytosis

Exocytosis is the biological dual to endocytosis and consists in the expulsion of biological sub-components. The representation of exocytosis, just as that of endocytosis, is influenced by the architectural choice of preventing box nesting. In Beta-binders exocytosis is encoded by expelling a parallel component of the internal pi-process, but taking care of wrapping it by the suitable interface.

Suppose that the component P has to be expelled from $\boldsymbol{B}[P \mid Q]$. Then we distinguish two principal cases. If P shows interaction capabilities with the external world via some binder in $\boldsymbol{B}$ then exocytosis can be rendered by appropriately tuning f_{split} in such a way that the split reduction

$$\boldsymbol{B}[P \mid Q] \longrightarrow \boldsymbol{B}_1[P\sigma_1] \parallel \boldsymbol{B}_2[Q\sigma_2]$$

leaves the original interaction potentials of both P and Q unaffected. Notice that this kind of representation also requires, e.g., $\boldsymbol{B}$ be composed by at least two elementary binders. If this is not the case, then an extra ‘dummy’ site can be exposed before the split reduction is made applicable.

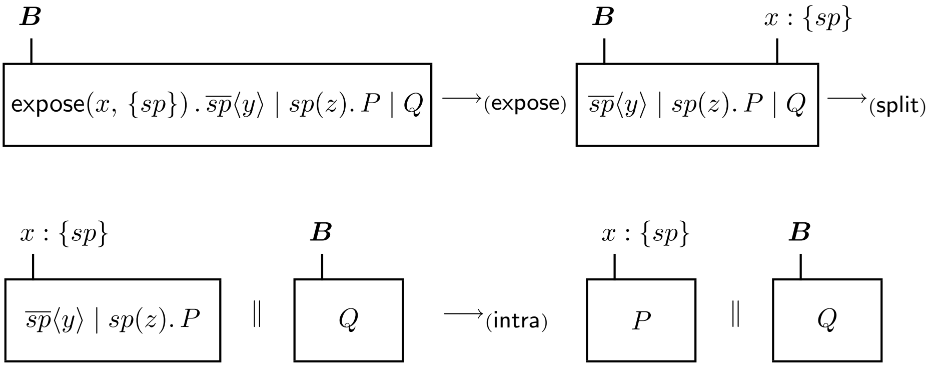
Assume now that all the sites in $\boldsymbol{B}$ are meant to model the external interaction capabilities of Q, and that a split should not move out any of them. To model this kind of situation two ancillary new names sp and z can be used, and $\boldsymbol{B}[P \mid Q]$ can be translated into the following beta-process:

$$\boldsymbol{B}[(\mathsf{expose}(x,\ \{sp\})\,.\,\overline{sp}\langle y\rangle \mid sp(z).\,P) \mid Q]$$

Then, taking for instance the following instance of f_{split},

$$\begin{array}{rll} f_{split} = \lambda \boldsymbol{B} P_1 P_2. & \text{if} & (P_1 \equiv (\overline{sp}\langle y\rangle \mid sp(z).\,P) \text{ and } \boldsymbol{B} = \boldsymbol{B}^*\ \beta(x : \{sp\})) \\ & \text{then} & (\beta(x : \{sp\}), \boldsymbol{B}^*, \sigma_{id}, \sigma_{id}) \\ & \text{else} & \perp \end{array}$$

we would would get, graphically, the derivation below (where each transition is labelled by the main axiom involved, while the possible use of redex and struct is not mentioned for notational convenience).



3.3 Replication of Beta-Processes

Recursion could be explicitly added to beta-processes by augmenting the syntax with a bang constructor, say $!!\,\boldsymbol{B}[P]$. As usual, to give semantics to this replication constructor the following extra structural law would be used:

$$!!\,\boldsymbol{B}[P] \equiv \boldsymbol{B}[P] \parallel !!\,\boldsymbol{B}[P] \tag{2}$$

Here we argue that the bang operator over pi-processes, together with an appropriate instance of the axiom split, is enough to simulate replication of beta-processes. Consider, in fact, the following definition of f_{split}:

$$f_{split} = \lambda \boldsymbol{B} P_1 P_2.\ \begin{array}{ll} \text{if} & (P_1 \equiv P \text{ and } P_2 \equiv\, !\, P) \\ \text{then} & (\boldsymbol{B}, \boldsymbol{B}, \sigma_{id}, \sigma_{id}) \\ \text{else} & \bot \end{array} \tag{3}$$

Correspondingly to (2), by instantiating the split axiom with (3) and then using the struct rule, we get the following behaviour for $\boldsymbol{B}[!\,P]$.

$$\frac{\dfrac{!\,P \equiv P \mid !\,P}{\boldsymbol{B}[!\,P] \equiv \boldsymbol{B}[P \mid !\,P]} \qquad \boldsymbol{B}[P \mid !\,P] \longrightarrow \boldsymbol{B}[P] \parallel \boldsymbol{B}[!\,P]}{\boldsymbol{B}[!\,P] \longrightarrow \boldsymbol{B}[P] \parallel \boldsymbol{B}[!\,P]}$$

This suggests that a spawn of $!!\,\boldsymbol{B}[P]$ can be mimicked by a reduction of $\boldsymbol{B}[!\,P]$.

Remark 1. Whichever move of $!!\,\boldsymbol{B}[P]$ *can be matched by a multi-step derivation from* $\boldsymbol{B}[!\,P]$.

To support Remark 1 it is sufficient to notice that any operational derivation for $!!\,\boldsymbol{B}[P]$ takes the shape:

$$\dfrac{!!\,\boldsymbol{B}[P] \equiv B_1 \qquad \begin{array}{c} B \longrightarrow B_2 \\ \vdots\ \textsf{(redex, struct)} \\ B_1 \longrightarrow B' \end{array}}{!!\,\boldsymbol{B}[P] \longrightarrow B'}\ \textsf{(struct)} \tag{4}$$

where $!!\,\boldsymbol{B}[P]$ has been spawned either once or more times to get B_1 and the triggering step $B \longrightarrow B_2$ depends on either a single copy of $\boldsymbol{B}[P]$ or on an interaction (or join) between to distinct copies of the replicated box. Namely:

- B_1 is structurally congruent to either $\boldsymbol{B}[P] \parallel !!\,\boldsymbol{B}[P]$ or to
  $$B_1{}^n = \underbrace{\boldsymbol{B}[P] \parallel \ldots \parallel \boldsymbol{B}[P]}_{n>1} \parallel !!\,\boldsymbol{B}[P]$$
- B is either $\boldsymbol{B}[P]$ or $\boldsymbol{B}[P] \parallel \boldsymbol{B}[P]$.

Moreover, as shown above, the struct rule with a split in its premise can be used to transform $\boldsymbol{B}[!\,P]$ into a beta-process having $\boldsymbol{B}[P]$ as one of its parallel components. Hence, correspondingly to the move inferred for $!!\,\boldsymbol{B}[P]$ in (4) some B'' matching B' exists such that

$$\begin{array}{ll} \boldsymbol{B}[!\,P] \longrightarrow \boldsymbol{B}[P] \parallel \boldsymbol{B}[!\,P] \longrightarrow B'' & \text{if } B_1 \equiv \boldsymbol{B}[P] \parallel !!\,\boldsymbol{B}[P] \\ \boldsymbol{B}[!\,P] \longrightarrow^n \underbrace{\boldsymbol{B}[P] \parallel \ldots \parallel \boldsymbol{B}[P]}_{n>1} \parallel \boldsymbol{B}[!\,P] \longrightarrow B'' & \text{if } B_1 \equiv B_1{}^n. \end{array}$$

3.4 Splitting Joins (and Joining Splits)

We now investigate the conditions under which a derivation inferred by using the split (join) axiom can be immediately ‘undone’ by a derivation that makes use of join (split), so leading to a cycle like:

$$B_1 \longrightarrow\equiv B_2 \longrightarrow\equiv B_1.$$

The following observations are easy consequences of the definition of the operational semantics.

Remark 2. If $\boldsymbol{B}[P_1 \mid P_2] \longrightarrow \boldsymbol{B}_1[P_1\sigma_1] \parallel \boldsymbol{B}_2[P_2\sigma_2]$ *and* $f_{join}(\boldsymbol{B}_1, \boldsymbol{B}_2, P_1\sigma_1, P_2\sigma_2) = (\boldsymbol{B}, \sigma_1', \sigma_2')$ *and, for* $i = 1, 2$, $\sigma_i\sigma_i'$ *is the identity over* $\mathsf{fn}(P_i)$*, then* $\boldsymbol{B}[P_1 \mid P_2] \longrightarrow^2\equiv \boldsymbol{B}[P_1 \mid P_2]$.

Remark 3. If $\boldsymbol{B}_1[P_1] \parallel \boldsymbol{B}_2[P_2] \longrightarrow \boldsymbol{B}[P_1\sigma_1 \mid P_2\sigma_2]$ *and* $f_{split}(\boldsymbol{B}, P_1\sigma_1, P_2\sigma_2) = (\boldsymbol{B}_1, \boldsymbol{B}_2, \sigma_1', \sigma_2')$ *and, for* $i = 1, 2$, $\sigma_i\sigma_i'$ *is the identity over* $\mathsf{fn}(P_i)$*, then* $\boldsymbol{B}_1[P_1] \parallel \boldsymbol{B}_2[P_2] \longrightarrow^2\equiv \boldsymbol{B}_1[P_1] \parallel \boldsymbol{B}_2[P_2]$.

3.5 Inter-actions Versus Intra-actions

We now comment on the difference between inter-actions and intra-actions. A very first observation is that inter-actions, distinctly from intra-actions, are non driven by the syntactic identity of input and output channels. The interesting point, however, comes from the possible interplay of the join axiom with the inter and the intra rule, respectively. In particular, we focus on the conditions under which the application of an inter followed by a join can be mimicked (up to structural equivalence) by the application of a join followed by an intra.

Assume that the beta-processes $\boldsymbol{B}_1[P]$ and $\boldsymbol{B}_2[Q]$ can inter-communicate. Then, by the join rule in Table 1, appropriate binders and pi-processes exist such that:

1. $P \equiv \nu\tilde{u}\,(x(w).\,P_1 \mid P_2)$ and $Q \equiv \nu\tilde{v}\,(\overline{y}\langle z\rangle.\,Q_1 \mid Q_2)$
2. $\boldsymbol{B}_1 = \beta(x : \varGamma)\,\boldsymbol{B}_1^*$ and $\boldsymbol{B}_2 = \beta(y : \varDelta)\,\boldsymbol{B}_2^*$
3. $\varGamma \cap \varDelta \neq \emptyset$ and $x, z \notin \tilde{u}$ and $y, z \notin \tilde{v}$

and

$$\begin{aligned}&\boldsymbol{B}_1[P] \parallel \boldsymbol{B}_2[Q] \longrightarrow\\ &\boldsymbol{B}_1\big[\nu\tilde{u}\,(P_1\{{}^{z}\!/\!_{w}\} \mid P_2)\big] \parallel \boldsymbol{B}_2\big[\nu\tilde{v}\,(Q_1 \mid Q_2)\big] = B_{inter}\end{aligned} \tag{5}$$

Suppose now that that the join axiom could be applied to $\boldsymbol{B}_1[P] \parallel \boldsymbol{B}_2[Q]$, too. Then, for $\boldsymbol{B}$, σ_1 and σ_2 such that $f_{join}(\boldsymbol{B}_1, \boldsymbol{B}_2, P, Q) = (\boldsymbol{B}, \sigma_1, \sigma_2)$, and up to structural congruence, we have:

$$\begin{aligned}&\boldsymbol{B}_1[P] \parallel \boldsymbol{B}_2[Q] \longrightarrow\\ &\boldsymbol{B}\big[\nu\tilde{s}\,(x\sigma_1(w').\,P_1\{{}^{w'}\!/\!_{w}\}\sigma_1 \mid P_2\sigma_1 \mid \overline{y\sigma_2}\langle z\sigma_2\rangle.\,Q_1\sigma_2 \mid Q_2\sigma_2)\big] = B_{join}\end{aligned} \tag{6}$$

The beta-process B_{inter} in (5) could undergo a join reduction, even relative to an instance of f_{join} distinct from that used to derive B_{join} in (6). Such a reduction could render B_{inter} structurally congruent to a single-box beta-process, say $B'_{join} = \boldsymbol{B}'[R]$. Nonetheless, in order to be able to infer that $B_{join} \longrightarrow\equiv \boldsymbol{B}'[R]$ and hence close the diamond below

$$\begin{tikzcd} & \boldsymbol{B}_1[P] \parallel \boldsymbol{B}_2[Q] \arrow[dl] \arrow[dr] & \\ B_{inter} \arrow[dr] & & B_{join} \\ & B'_{join} & \end{tikzcd}$$

it would at least be necessary to infer, via the intra rule, a communication between the input action $x\sigma_1(w')$ and the output action $\overline{y\sigma_2}\langle z\sigma_2\rangle$. Hence it should be $x\sigma_1 = y\sigma_2$.

We conclude with an observation on the relative applicability of inter- and intra-actions.

Remark 4. Whenever $f_{join}(\boldsymbol{B}_1, \boldsymbol{B}_2, P_1, P_2) = (\boldsymbol{B}, \sigma_1, \sigma_2)$ and σ_1, σ_2 do not cause the clash of names in $\mathsf{sub}(\boldsymbol{B}_1, \boldsymbol{B}_2)$, an inter-reduction followed by the joining of the two boxes cannot be mimicked by first joining the boxes and then using the intra *rule.*

The above remark suggests that beta binders are *active* borders, in the sense that communications over the sites of boxes cannot in general be rendered by resorting to synchronizations between components of the inner pi-process.

3.6 Directionality of Join (and Split)

In what follows we focus on the possible simmetry deriving from the interplay of the struct rule with the join or the split axiom. Consider, for instance, the beta-process

$$\beta(x : \Gamma)\,[P_1] \parallel \beta(y : \Delta)\,[P_2] \tag{7}$$

and suppose that $\Gamma \cap \Delta \neq \emptyset$ and that the join axiom of the reduction system is used with the instance of f_{join} defined in (1). Then, letting $\boldsymbol{B}_1 = \beta(x : \Gamma)$ and $\boldsymbol{B}_2 = \beta(y : \Delta)$, the following derivations are both legitimate:

$$\boldsymbol{B}_1[P_1] \parallel \boldsymbol{B}_2[P_2] \longrightarrow \boldsymbol{B}_1[P_1 \mid P_2\{x/y\}]$$

$$\frac{\boldsymbol{B}_1[P_1] \parallel \boldsymbol{B}_2[P_2] \equiv \boldsymbol{B}_2[P_2] \parallel \boldsymbol{B}_1[P_1] \qquad \boldsymbol{B}_2[P_2] \parallel \boldsymbol{B}_1[P_1] \longrightarrow \boldsymbol{B}_2[P_2 \mid P_1\{y/x\}]}{\boldsymbol{B}_1[P_1] \parallel \boldsymbol{B}_2[P_2] \longrightarrow \boldsymbol{B}_2[P_2 \mid P_1\{y/x\}]}$$

Biologically speaking, the above symmetry is not necessarily a good feature. It could corresponds, e.g., to assessing that a bacterium can engulf a macrophage in the same way as a macrophage can engulf a bacterium. Unwanted bi-directional applications of the join can be avoided by properly acting on the types of sites by adding names that identify the component they refer to. An analogous reasoning holds of the split axiom.

In [8] we suggested to adopt a partial ordering $\sqsubseteq$ on a specialized subset of names (say $n_1 \sqsubseteq n_2 \sqsubseteq \ldots$) used to denote the 'endocytosis propension' of sites. Then the bi-directionality we commented upon could be disrupted by first imposing that the type of each site contains one of those special names, and finally dicriminating the absorbing process by the absorbed one on the basis of the endocytosis propension that they exhibit at their sites. For example, the definition in (1) could be refined into the following one.

$$\begin{aligned} f_{join} = \lambda \boldsymbol{B}_1 \boldsymbol{B}_2 P_1 P_2.\ & \text{if } (\boldsymbol{B}_1 = \beta(x : \{n_j\} \cup \Gamma)\, \boldsymbol{B}_1^* \text{ and} \\ & \qquad \boldsymbol{B}_2 = \beta(y : \{n_i\} \cup \Delta)\, \boldsymbol{B}_2^* \text{ and} \\ & \qquad \Gamma \cap \Delta \neq \emptyset \text{ and } n_i \sqsubseteq n_j) \\ & \text{then } (\boldsymbol{B}_1, \sigma_{id}, \{x/y\}) \\ & \text{else } \bot \end{aligned}$$

When the complete system is not so big, and its players are a-priori known, resorting to a partial ordering on names is surely superfluous. Unwanted reductions can be cut out by just augumenting the type of each of the relevant sites with one special name that identifies the component it belongs to. Then it is sufficient to let f_{join} inspect such a special name. Suppose for instance that in (7) the beta-process $\beta(x : \Gamma)\,[P_1]$ plays the macrophage and $\beta(y : \Delta)\,[P_2]$ the bacterium, and consider the following refinement of (7) and (1) respectively:

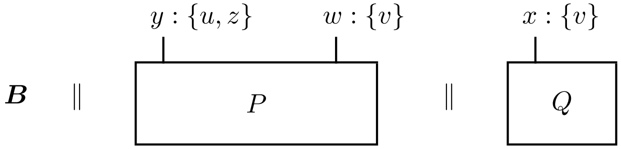
$$\beta(x : \Gamma \cup \{m\})\,[P_1] \parallel \beta(y : \Delta)\,[P_2]$$

$$f_{join} = \lambda \boldsymbol{B}_1 \boldsymbol{B}_2 P_1 P_2.\ \begin{array}{l} \text{if } (\boldsymbol{B}_1 = \beta(x : \Gamma \cup \{m\})\,\boldsymbol{B}_1^* \text{ and } \boldsymbol{B}_2 = \beta(y : \Delta)\,\boldsymbol{B}_2^* \text{ and} \\ \quad \Gamma \cap \Delta \neq \emptyset) \\ \text{then } (\boldsymbol{B}_1, \sigma_{id}, \{x/y\}) \\ \text{else } \bot \end{array}$$

where m (for macrophage) is supposed to be a fresh name. The above refinement would be sufficient to cut out the unwanted behaviour of the system.

3.7 Nesting

In Beta-binders nesting of boxes was forbidden to keep the formalism as simple as possible. The role of typing for sites, however, together with the operational semantics of interactions between boxes, ensures that a virtual (if not graphical) form of nesting can be represented. Take for instance the following system:



Then, under the proviso that v is fresh w.r.t. the names exploited in the types of the sites in $\boldsymbol{B}$, the beta-process $\beta(x : \{v\})\,[Q]$ can only perform intra-actions or be involved in inter-actions with $\beta(y : \{u, z\})\,\beta(w : \{v\})\,[P]$ through its site x. Then, at least as long as its interface is not changed, $\beta(x : \{v\})\,[Q]$ behaves like a compartment nested in $\beta(y : \{u, z\})\,\beta(w : \{v\})\,[P]$.

3.8 Changing the Type of Sites

Changing the type of sites can be particularly useful for modeling purposes. Imagine for instance that the beta-process $\boldsymbol{B}[P]$ has a site $x : \Delta$, and suppose that, due to biochemical modifications of the component, or even to its evolutionary behaviour, the type of the site has to be changed to Γ. This can be the case, e.g., when modeling the fact that a molecule can be alternatively phosphorilated and dephosphorilated (see, e.g., [2]). Phosphorilation can be rendered as an interaction at a certain site $x : \{ph\}$. After phosphorilation, the site $x : \{ph\}$ is made unavailable to further interactions, and the molecule shows to be ready to possible dephosphorilation by exposing another suitable site, say $y : \{deph\}$.

The modification of site types can be rendered in Beta-binders by a combination of the hide and of the expose prefixes as reported below.

Remark 5. Given $\boldsymbol{B}[P]$*, the modification of the site* $x : \Delta$ *into* $x : \Gamma$ *can be rendered by translating* P *into* $\mathsf{hide}(x)\,.\,\mathsf{expose}(x, \Gamma)\,.\,P$*. Relatively to the beta-process* $\boldsymbol{B}[\,\mathsf{hide}(x)\,.\,\mathsf{expose}(x, \Gamma)\,.\,P]$ *notice that, by definition of the operational* *semantics, when the* $\mathsf{expose}(x,\, \Gamma)$ *prefix is fired the name* x *is refreshed in* P *to avoid clashes with the subject of the hidden site.*

As a final observation, recall that the names occurring in the types declared in expose prefixes are free names, and hence can be affected by substitutions. For this reason, the types of sites can dynamically change simply due to the evolution of beta-processes. This happens, e.g., in the following case:

$$\beta(x : \Delta)\, \big[x(y_1).\, x(y_2).\, \mathsf{expose}(y,\, \{y_1, y_2\})\, .\, P\big] \parallel \beta(w : \Delta)\, \big[\overline{w}\langle z_1\rangle .\, \overline{w}\langle z_2\rangle .\, Q\big] \longrightarrow^2$$

$$\beta(x : \Delta)\, \big[\, \mathsf{expose}(y,\, \{z_1, z_2\})\, .\, P\big] \parallel \beta(w : \Delta)\, \big[Q\big].$$

4 Conclusions

We overviewed Beta-binders and commented on a set of operational properties and patterns that can be useful schemata when modeling complex case studies. Most of the features that have been tackled depend on the level of parametricity that is ensured by building the formalism around typed interaction sites. For instance, we noticed that appropriate types can be used to simulate hierarchies of nested boxes.

More generally, the typing of sites allows an improved promiscuity of interaction between entities, and we believe that this can be a relevant point in modeling biological behaviours, especially in the perspective of predictive research and in the analysis of the responses of biological systems to artificial perturbations.

References

1. L. Cardelli. Membrane interactions. In *BioConcur '03, Workshop on Concurrent Models in Molecular Biology*, 2003.
2. F. Ciocchetta, C. Priami, and P. Quaglia. Modeling Kohn interaction maps with Beta-binders: an example. Submitted for publication., 2004.
3. V. Danos and J. Krivine. Formal molecular biology done in CCS-R. In *BioConcur '03, Workshop on Concurrent Models in Molecular Biology*, 2003.
4. V. Danos and C. Laneve. Core formal molecular biology. In P. Degano, editor, *Proc. 12th European Symposium on Programming (ESOP '03)*, volume 2618 of *Lecture Notes in Computer Science*. Springer, 2003.
5. V. Danos and C. Laneve. Core formal molecular biology. Full version of [4], submitted for publication, 2004.
6. R. Milner. *Communication and Concurrency.* International Series in Computer Science. Prentice Hall, 1989.
7. R. Milner. *Communicating and mobile systems: the π-calculus.* Cambridge Universtity Press, 1999.
8. C. Priami and P. Quaglia. Beta binders for biological interactions. In *Proc. CMSB '04*, Lecture Notes in Bioinformatics, 2004.
9. C. Priami, A. Regev, W. Silverman, and E. Shapiro. Application of a stochastic name-passing calculus to representation and simulation of molecular processes. *Information Processing Letters*, 80(1):25–31, 2001.

10. A. Regev, E.M. Panina, W. Silverman, L. Cardelli, and E. Shapiro. Bioambients: An abstraction for biological compartments. *Theoretical Computer Science*, 2004. To appear.
11. A. Regev, W. Silverman, and E. Shapiro. Representation and simulation of biochemical processes using the pi-calculus process algebra. In *Proc. of the Pacific Symposium on Biocomputing (PSB '01)*, volume 6, pages 459–470. World Scientific Press, 2001.
12. D. Sangiorgi and D. D. Walker. *The π-calculus: a Theory of Mobile Processes.* Cambridge Universtity Press, 2001.

Discrete Event Multi-level Models for Systems Biology

Adelinde M. Uhrmacher[1], Daniela Degenring[1], and Bernard Zeigler[2]

[1] Institute for Computer Science, University of Rostock, Albert-Einstein-Str. 21,
D-18059 Rostock, Germany
{lin, dd058}@informatik.uni-rostock.de

[2] Arizona Center of Integrative Modeling and Simulation, University of Arizona,
Tucson, AZ 85721, USA
zeigler@ece.arizona.edu

Abstract. Diverse modeling and simulation methods are being applied in the area of Systems Biology. Most models in Systems Biology can easily be located within the space that is spanned by three dimensions of modeling: continuous and discrete; quantitative and qualitative; stochastic and deterministic. These dimensions are not entirely independent nor are they exclusive. Many modeling approaches are hybrid as they combine continuous and discrete, quantitative and qualitative, stochastic and deterministic aspects. Another important aspect for the distinction of modeling approaches is at which level a model describes a system: is it at the "macro" level, at the "micro" level, or at multiple levels of organization. Although multi-level models can be located anywhere in the space spanned by the three dimensions of modeling and simulation, clustering tendencies can be observed whose implications are discussed and illustrated by moving from a continuous, deterministic quantitative macro model to a stochastic discrete-event semi-quantitative multi-level model.

1 Introduction

The goal of Systems Biology is to analyze the behavior and interrelationships between entities of entire functional biological systems [1, 2]. As the systems under study do not support an easy experimental access and analysis, models play an important role in gaining an insight into the systems‘ behavior and structure. Models can be evaluated differently. For instance, properties of the system can be derived by using methods like model checking [3]. Simulation is a different approach as it means an experiment based on the model. Thereby, it completes the *in-vivo* or *in-vitro* experiments of Systems Biology by *in-silico* experiments [1, 2]. Diverse modeling and simulation methods are being applied in the area of Systems Biology. Efforts like the "Systems Biology Workbench" are aimed at integrating different data analysis, visualization, modeling, and simulation tools [4]. In this context SBML (Systems Biology Markup Language) is being developed to support the exchange of models between different simulation systems

C. Priami et al. (Eds.): Transactions on Computational Systems Biology, LNBI 3380, pp. 66–89, 2005.

[5]. Similar to CellML [6], SBML focuses on continuous systems modeling and simulation. In continuous systems models, the system is described by a set of state variables, whose time-dependent changes are usually specified by a set of differential equations [7, 8, 9, 10].

Aside from the continuous modeling approaches the discrete approaches also have increasingly gained momentum: the behavior of the system is modeled by states changing at arbitrary points on a still continuous time scale [11, 12]. State transitions are triggered by external and internal events, which is in fundamental contrast to the continuous state changes of a differential equation system. With the discrete approaches models have emerged that integrate qualitative and stochastic aspects: the values of some variables and/or of the modeling parameters are qualitatively scaled or taken from probability distributions [13]. Qualitative continuous systems modeling approaches exist as well even though those are more rare [14, 15]. The same can be observed with respect to stochastic continuous approaches, e.g. [16].

Most models in Systems Biology can easily be located within the space that is spanned by the three dimensions of modeling: continuous and discrete [17]; quantitative and qualitative; stochastic and deterministic, although this categorization is neither exclusive in each dimension, nor are the dimensions entirely independent.

In the following we will add a comparatively less explored dimension to distinguish modeling approaches in Systems Biology: the question at which level a model describes a system: is it at the “macro” level, at the “micro” level, or at multiple levels of organization.

The more mature a field becomes, the more hybrid approaches gain ground. In Systems Biology, the number of models steadily increases that are no longer purely quantitative or qualitative, or purely continuous or discrete. Thus, we expect the same to happen with respect to the organizational level: one level of explanation will hardly suffice. The more so, as the goal of Systems Biology is to describe the dynamics of cellular systems in their entirety [1]. In this context, not only interdependencies at one organizational level but between different ones become of interest, as “the whole is to some degree constrained by the parts (upward causation), but at the same time the parts are to some degree constrained by the whole (downward causation).” [18]. The importance of these interdependencies has been emphasized for systems in general [19] and biological systems in particular [20, 21], and also recently for Systems Biology with increasing urgency [22, 23, 24, 25] motivating the development of concrete models, e.g. [26, 27, 28, 29].

This paper is organized as follows: in chapter two the modeling approaches in the area of Systems Biology are categorized into the three modeling dimensions, which were described above (quantitative-qualitative, continuous-discrete and stochastic-deterministic); chapter three introduces the idea of the different organization levels for the distinction of the modeling approaches (micro, macro and multi-level models) and interrelates the categorization approach to the modeling dimensions studied in chapter two with a focus on multi-level models; after theoretical considerations about the different modeling dimensions and their interdependencies, chapter four illustrates the explorations based on a biological application example; chapter five gives a more in-detail discussion of the composition and interaction of multi-level models; it is followed by a general discussion, summary and outlook.

2 Structuring the Space of Model Approaches

A formal model is described in a formal language to be interpretable by a computer system. Each model forms an abstraction of a system to support some concrete objective. Thus, we follow the definition of Minsky [30] that "A Model (M) for a system (S) and an experiment (E) is anything to which E can be applied in order to answer questions about S.". As Cellier [31] points out, this definition does not describe "models for systems" per se, a model is always related to the tuple system and experiment. A model of a system might therefore be valid for one experiment and invalid for another. One consequence of this definition is that it is very unlikely to derive a model, which is valid for all possible experiments, unless it is an identical copy of the system and thus no longer a model. Modeling is a process of abstraction. It involves simplification, aggregation, and omission of details. Although processes of omission and simplification become particularly obvious if the model is described in a formal language, these processes play also a role in *in-vitro* and *in-vivo* experiments. Whereas it seems natural to conclude that the physical medium of *in-vitro* or *in-vivo* experiments restrain the experiments and thus the question that can be answered, one is often not aware about that and what constraints are implied by the respective modeling approach. However, the diversity of modeling approaches applied in Systems Biology illustrates and suggests that, depending on the biological system, the available data and knowledge about the system, and the objective of the simulation study, modeling approaches are chosen deliberatively on demand and thus address the diverse needs of modeling and simulation in Systems Biology - if we do not assume that the diversity is caused merely by the diverse backgrounds which the modelers come from. So the question is to be asked what do certain approaches offer in modeling biological systems when compared to others. By introducing and discussing the dimension of organizational levels we will try to partially answer this question for the case of discrete-event, multi-level modeling approaches.

First we will use the dimensions of continuous and discrete, quantitative and qualitative, and stochastic and deterministic modeling to structure the space of modeling approaches applied in Systems Biology.

2.1 Continuous, Discrete, and Hybrid System Models

Distinguishing between continuous and discrete systems modeling and simulation has a comparatively long tradition. Although different modeling formalisms do exist, e.g. systems dynamics, bond graphs, or block diagrams, the continuous realm of modeling and simulation is unified by differential equations for model representation and numerical integration algorithms for execution [31]. Thereby time-dependent variables are assigned to different measuring or non-measurable quantities of the system. The continuous state changes are modeled by a sum of rates describing the increase and decrease of the quantities amounts. Frequently kinetic rate equations, like the Michaelis-Menten or some mass action kinetics, are used for that purpose. Such modeling approaches are perfectly suited for the reproduction of measured time-dependent trajectories and also easily allow the fitting of the model parameters. Continuous systems models are the dominant type of model used in Systems Biology [32]. A series of simulation tools for continuous systems modeling and simulation in general and Systems Biology applications in particular support a comfortable developing of these types of models, e.g. Gepasi [33], ProMoT/Diva [34], Jarnac [35], DBsolve [36], and Cellerator [37]. Continuous models reflect nicely what is measured in cellular biology. Small samples of cell cultures are analyzed by extracting the DNA, enzymes, or metabolites, and by quantifying the concentration of the respective species over time.

Often a cell's activity is perceived as being discrete rather than continuous motivating the design of discrete systems models. In contrast to continuous systems models, discrete systems models assume only a finite number of state changes within a time interval. Depending on the time base that underlies the model, discrete time stepped approaches and discrete event approaches are distinguished. The latter allows to associate arbitrary time spans with each state of the system and thus is based on a continuous notion of time, whereas the former is based on time that advances in equidistant steps. Regular, time-stepped Petri Nets have been applied to qualitatively describe biochemical reaction networks [38]. The use of stochastic Petri Nets marks the transition to discrete event simulation and the integration of quantitative and stochastic aspects [39]. In discrete event models state transition functions define into which state to change triggered by external events, e.g. the collisions of species like enzymes and metabolites in a biological model, or triggered by the flow of time, e.g. after the time required for intra-molecular rearrangements. In discrete event simulation, situation-based and time-based events can occur at any point in time and the resulting state and the time span needed for reaction can be randomly chosen. Thus, stochasticity comes natural to discrete event simulation (see section 2.2).

Continuous systems models can easily be translated into a set of differential equations, independently of being defined as bond graphs, as block diagrams, or as set of chemical reactions. The discrete modeling and simulation realm lacks such a common denominator that is widely accepted, even though general approaches exist. E.g. Devs [40], Petri Nets [41], and π-Calculus [42, 43] are formal and generally applicable approaches toward discrete event systems modeling. Each has been developed with a rather different objective in mind. The goal of Devs has been to combine the functional, network and hierarchical perspective in describing systems, and thus stands in the tradition of general systems theory [44]. Devs distinguishes between atomic models and coupled models. Whereas atomic models describe the behavior in terms of state transitions that might be triggered by external events or the flow of time, and output functions, coupled models define how their components, which might be atomic or coupled, interact with each other and thus control the interaction between them. Thus, a hierarchical, modular construction of models is supported. An abstract simulator defines the execution semantics of typical DEVS models [40]. Extensions of DEVS support variable structure models [45], models that entail in their description the ability to change their own composition and interaction structure which is important in modeling and simulating biological systems [46]. DEVS models emphasize the definition of states and state transitions and therefore, are closely related to STATECHARTS, a model formalism that is widely applied in Systems Biology [47]. STATECHARTS can easily be transformed into DEVS. A transformation of DEVS models into the graphical notation of STATECHARTS facilitates the understanding of models [48].

PETRI NETS and π-CALCULUS have been developed for describing concurrent processes and are best known in the context of computer and engineering sciences. Whereas PETRI NETS focus on concurrent processes competing for resources, the π-CALCULUS is aimed at describing concurrent mobile processes, channels, locations, and interactions respectively. Thus, processes like protein to protein interactions can be described easily [49, 50, 51, 52]. Its extension in form of the stochastic π-CALCULUS, supports the definition of discrete event models and their execution by discrete event simulation. Thus, established approaches to transform existing continuous models into discrete event models [53], can be used to define and refine models in the stochastic π-CALCULUS. Openly available simulation systems like BIOSPI also push the application of the stochastic π-CALCULUS [54]. Recent developments like BIOAMBIENTS which is based on the stochastic π-CALCULUS, allow the description of spatial cell compartments, and entities moving from one compartment to the next and thus increase the expressiveness of the language [55]. The BRANE CALCULI [56, 57] addresses the need for modeling constructs of cellular coordination via membranes. It forms an application specific refinement of the general modeling and simulation approach. In the PROJECTIVE BRANE CALCULI the membrane actions become directed thereby, moving the calculi even closer to the perception of the activities within biological membranes [58]. These recent extensions are aimed at providing means and places for describing coordination and cooperation within biological cells [50], and lend additional structure and expressiveness to the modelling language. To specifiy the executional semantics of a model in a non-ambiguous manner an abstract simulator has been developed for the π-CALCULI [54], as has been done for DEVS like models [40]. One might note that when a continuous model is executed, numerical integration algorithms discretize the state and time base, and so diminish the conceptual distance toward discrete models. However, the assumption underlying continuous models is still that the system behaves continuously with an infinite number of infinitely close state transitions in each time interval. The numerical integration merely serves to approximate this behavior. In discrete event models in contrast, no continuity of behavior needs to be assumed. However, the situation becomes more interesting since recently it has

Table 1. Modeling formalisms: time and state space

	Discrete event	Discrete step-wise	Differential equation
input and output	arbitrary	arbitrary	real vector
state space	arbitrary	arbitrary	real vector
time base	real	discrete	real

been shown that discrete event models can be used to obtain approximations to the solutions of differential equation systems [17, 59]. This is done using a process of quantization in which events are scheduled based on predicted threshold crossings rather than time steps. Executed by a discrete event simulation engine it will reproduce the trajectories, in some cases in significantly less time [60].

Often systems can best be described by a combination of discrete and continuous models, e.g. if continuous processes exhibit discontinuities which require to switch from one continuous model to another one, or if leaving or entering a discrete phase depends on continuous processes that reach certain thresholds. Hybrid systems models combine continuous and discrete systems behavior. Many modeling and simulation approaches for discrete and continuous systems have been extended to support hybrid systems models. Hybrid Petri Nets have been developed by adding continuous places and continuous transitions to the discrete places and transitions of regular Petri Nets [61]. The continuous transitions of Hybrid Petri Nets are used to describe kinetic reactions which are turned on and off by the marking of discrete places. These discrete places form the interface between continuous and discrete partitions of the Petri Net [62]. Hybrid Petri Nets as a graphical tool are well suited to describe metabolic processes, as they visualize chemical reactions and interdependencies. Similar arguments motivate the use of Block Diagrams, that allow to specify graphically continuous and hybrid models [63] and are supported by many simulation tools, e.g. [64]. The origin of block diagrams, unlike that of Petri Nets, lies in the continuous realm. To allow the integration of discontinuities they have been extended by discrete elements, e.g. switching blocks. Both Hybrid Petri Nets and Block Diagrams support the *mixed signal approach* in describing hybrid systems [65]. In contrast to that, Hybrid Automata [66] move the distinction of phases into the focus of modeling. State transitions of Hybrid Automata are triggered by continuous processes that are responsible for describing the continuous behavior of a system while being in one phase and determining the time and situation when to leave a phase and enter another one [67, 68, 69]. The growing need to integrate discontinuous behavior into Systems Biology models is reflected in extending existing simulation systems, e.g. Gepasi, or in the design of recent simulation systems for Systems Biology, e.g. the e-Cell simulation system.

2.2 Deterministic and Stochastic Systems Models

Modeling is the process of structuring our knowledge about a given system [40]. In this perspective, stochastic processes represent one means to express the uncertainty of our knowledge. A plethora of methods are dedicated to the problems of stochastic modeling, e.g. to estimate suitable distributions for random variates, and to interpret the results of the simulation runs [70]. ¿From the view of the modelled system, integrating stochasticity into the models might also serve a slightly different purpose: randomness or ”noise” arising from small numbers of molecules involved in processes like gene expression and regulation can directly be represented in the model [71, 72]. Although stochastic elements are often associated with discrete event models, they are also applicable to continuous system models. In Systems Biology, inclusions of stochastic elements for modeling continuous processes have gained ground recently. E.g. chemical reaction equations are described by so called stochastic differential equations [73]. These equations determine the probability with which a combination of molecules will react in a given time interval.

Interestingly, to solve these equations Gillespie [74, 75] suggested an algorithm that transforms the set of equations into a discrete event stochastic model. The representation in discrete event form is particularly striking in more recent implementations of the algorithm. E.g. the simulation system Stochastirator is a discrete event simulator with the typical event queues and the handling of time and situation triggered events [73]. Stode [53] transforms automatically reaction rates and model parameters of a deterministic differential equation model internally into a stochastic discrete event model. The probabilities of single reactions depend on the number of reactants which again is subject to change via occurrence of reactions [73]. The stochastic discrete event models address specific constraints of continuous, deterministic models: concentrations do not necessarily change continuously, particularly if the dynamics of a small amount of entities, like DNA molecules and plasmids, shall be modeled [76]. In addition, sometimes, the dynamics of biological systems can be best approached in a stochastic manner, e.g. if the gene regulation is to be described [77], where stochastic fluctuations are abundant [78]. The exact stochastic simulation approach is not practical for the simulation of metabolic processes, in which large numbers of molecules of the same kind are involved, due to the computational cost for the calculation of all individual molecular collisions. Extensions of the approach overcome these difficulties and allow the stochastic simulation of systems composed of both intensive metabolic reactions and regulatory processes involving small numbers of molecules [79, 80, 81]. The combination of stochastic discrete with continuous sub-models has stimulated the desire for an easy integration of stochastic aspects into continuous models. One common approach is to assume a normal distribution for key parameters of the differential equation system. The result is that stochasticity can now permeate the entire model [16].

2.3 Qualitative, Quantitative and Semi-quantitative Systems Models

Continuous models are usually associated with quantitative models, i.e. models whose variables are numerically scaled, in the case of differential equations the state space is given by real values vectors. However, continuous behavior can also be described qualitatively. E.g QSIM [82] assumes a continuous, respectively hybrid behavior of a system and describes this in qualitative terms, e.g. rising trends, falling trends, landmarks, etc. This approach has also been used in Systems Biology, e.g. to describe the development of a λ-Phage in an eukaryotic cell [83]. Often the lack of quantitative data motivates the use of qualitative methods. Qualitative methods are often used as a first step to develop a quantitative model [84]. E.g. one obtains useful structural information by determining what variables play a role for certain kinetics and whether there exists a positive or negative influence between variables [85, 86]. Another motivation for the application of qualitative methods is that they are aimed at answering different kinds of questions than quantitative methods and offer different possibilities for analysis, for example whether certain states can be reached by the system and under which conditions. On the other hand, if not only the existence but also the degree, and the effect of opposite regulations are of interest purely qualitative models will not prove to be sufficiently expressive [32].

It is one advantage of discrete event simulation that its models combine easily qualitative and quantitative aspects of the system [12] (see also table 1), even though the assumption, that discrete event simulation requires less data than continuous one, as stated in [12], has to be inspected critically. In hybrid systems models, the “qualitatively scaled variables” come into play to describe the different phases or to initiate switching from one differential equation system to another.

3 Micro, Macro, and Multi-level Systems Models

Traditionally, two dichotomous views on systems prevail. “With individualism, macroscopic processes are either emergents or totally reducible aggregates, while with holism microscopic actions occur as local manifestations of system-wide processes” [87]. In sociology the distinction between micro, macro, and, to mediate between both, multi-level models is comparatively well established [88, 89]. Macro models describe a system as one entity. Variables and their interdependencies, which can be expressed as rules, equations, constraints etc., are attributed to this entity. Typical representatives of this class are differential equation models, which describe e.g. a biochemical system based on concentrations and reaction rates.

Micro models are models that represent systems as comprising huge numbers of rather homogeneously structured entities. Only the behavior of the individuals is explicitly modeled. The macro level of the system exists only as it aggregates results of the activities at micro level and is used for reflecting emergent phenomena, e.g. the development of specific spatial patterns. They do not have any behavior of their own. Typical representatives of this class are cellular automata and Lindenmeyer systems which are also applied for reconstructing spatial biochemical processes in Systems Biology [90, 91, 92, 93, 94].

Micro models often form only a transition to multi-level models which describes a system at least at two different levels. Interactions are taking place within and between those levels. The description of systems at different levels of abstraction and different time scales facilitates taking spatial and temporal structured processes into consideration, e.g. [95].

Multi-level models allow us to explicitly describe “upward”,- and “downward causation”, i.e. “the whole is to some degree constrained by the parts (upward causation), but at the same time the parts are to some degree constrained by the whole (downward causation).” [18]. Their importance has been emphasized for systems in general [19] and biological systems in particular [20, 21]. The relevance of interrelating micro and macro models has also been raised recently for Systems Biology [22, 23, 24, 25].

The structure of multi-level models typically reveals whether they originated from macro or micro models. If the latter is the case we find a multiplicity of homogeneously structured entities, that describe e.g. different population of enzymes and proteins. If a macro model has been successively extended and refined to describe a system at different levels of organization, then comparatively few sub-models typically exist and those are heterogeneously structured with different patterns of behavior.

Individual-based models, which describe systems at two levels of organization, i.e. a micro and macro level [96, 97], belong to the class of multi-level models. They reveal their close relationship to micro models. In individual-based models the individual entities and the macro level are explicitly modeled. The individuals typically do not interact directly but via the macro level. Individual-based approaches are also increasingly being applied in cellular biology [98, 99, 100].

In the following we will shortly discuss the relationships of multi-level modeling and the previously discussed dimensions.

Continuous and Discrete Modeling. Multi-level models are neither restricted to discrete models nor to continuous ones. If multiple levels are formed by a successive extension and refinement of macro models they might be both, continuous or discrete. Continuous models can easily be structured into different cellular compartments, e.g. [101, 102], each of which behaves continuously. Even if multilevel models contain many homogeneously interacting and structured submodels, these might form continuous systems models [103], although in the case of many homogeneously interacting entities discrete models of the individuals prevail. They allow to combine a qualitative discrete perception of individuals and their behavior with a quantitative, concentration-oriented view at macro level and thus a comparison with measured concentration changes.

Deterministic and Stochastic Modeling. Again multi-level models might work deterministically or stochastically. The question whether stochastics plays a major role in multi-level models is closely related to the question whether discrete event models are part of the multi-level model. Most discrete event models consider stochastic effects in determining when and what will happen. In this case a simulation run turns into a random experiment and has to be treated as such [70].

When Gillespie suggested the transformation of deterministic continuous models into stochastic discrete event models, he also prepared the way for a micro perspective of cells. Although most implementations of the Gillespie algorithm, e.g. [73], record the number of molecules being in certain states and determine the time of next event and the most likely reaction to occur based on this "macro view", they consider the molecules as atomic entities to be added or deleted from the bulk solution. A next step has been taken in StochSim which attributes properties to these entities and thereby, allows to observe individual molecules over time. The model is based on a discrete step-wise execution [104]. The dominance of discrete approaches in multi-level modeling and single individuals being represented motivates the integration of stochastic aspects. So most multi-level models are stochastic models.

Quantitative and Qualitative Modeling. A multi-level model might be qualitative, quantitative, or semi-quantitative. If only continuous differential equation models are considered the state space presents itself as a vector of real numbers. Discrete event models of individuals support the representation of the modeled system by arbitrarily scaled variables. To allow a discrete event simulation to jump from one event to the next after some pre-defined time interval has elapsed, the time base of discrete event models is continuous and introduces typically some quantitative information - when does the next event occur or how long does a state persist per se. Purely qualitatively scaled variables are also perceivable. However, the combination of discrete, qualitative models at individual level and quantitative (discrete or continuous) models at macro level holds probably the most appeal to biologists as they allow to re-unify two perspectives in dealing with natural systems.

To support the modeling of complex systems, many formalisms, languages and tools allow to hierarchically compose models. Supporting a hierarchical structure of a model helps realizing multi-level models however not all hierarchically composed models are designed as multi-level models. They do not necessarily describe a system at different organizational levels, they use the model hierarchy for modularization.

If we categorize a model as being quantitative, stochastic, discrete and multi-level, it is therefore interesting to ask what this means not only for the model itself but for potential sub- and super-models: what can be deduced along a compositional model hierarchy, which might or might not reflect the different organizational levels. A quantitative model implies that all sub-models are quantitative, the supermodel might be semi-quantitative or quantitative. If one sub-model contains stochastic aspects the entire model becomes a stochastic one. If a model is continuous all sub-models are continuous and all supermodel will be either hybrid or continuous. If a model is a multi-level model, its sub-models might be micro, macro, or multi-level models, its super-model will definitely be a multi-level model. If we have a micro model, we will have many homogeneously structured sub-models each of which describes an individual at macro level, its super-model will be a micro or multi-level model. A macro model might not have any sub-models. In case it has components, its components are all macro models; the macro model itself can be part of a micro, macro, and multi-level model.

4 Biological Example: Diverse Models for the Tryptophan Synthase

After these theoretical considerations about the different modeling dimensions and their interdependencies, we will illustrate our exploration based on a biological example.

The Tryptophan Synthase is the last enzyme of the reaction cascade, which is responsible for the synthesis of the aromatic amino acid Tryptophan. The whole enzyme is a homo-dimer, whereas each monomer consists of two subunits, the α- and the β-subunit, which are connected by a largely hydrophobic tunnel. The enzyme has been isolated from microbial cells in [105] and characterized by *in-vitro* experiments using radiolabelled substrates. For the description of the *in-vitro* determined kinetics a quantitative, deterministic macro-model was developed [106]. The deduced reaction mechanism of the enzyme is shown in figure 1.

For each binding-state of the enzyme during the conversion from Indoleglycerol-3-phosphate (IGP) and Serine to Tryptophan and Glyceraldehyde-3-phosphate (G3P) at the α- and the β- subunit, respectively, a distinct variable was introduced, so that a system of ordinary differential equations could be derived (Figure 4).

After numerical integration and parameter fitting the trajectories resulting from several *in-vitro* experiments could be reproduced by the corresponding simulation experiments.

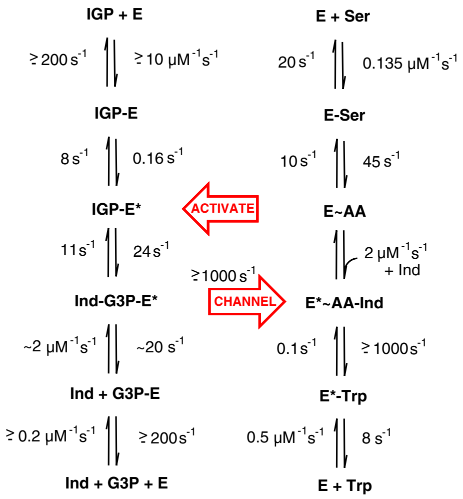


Fig. 1. Reaction scheme of tryptophan synthase [29]

$$
\begin{aligned}
\dot{IGP} &= 200IGP\text{-}E - 10IGP \cdot E \\
\dot{IGP\text{-}E} &= 10IGP \cdot E + 8IGP\text{-}E^* - 200IGP \cdot E - 0.16IGP\text{-}E \\
\dot{IGP\text{-}E^*} &= 0.16IGP\text{-}E + 11Ind\text{-}G3P\text{-}E^* - 8IGP\text{-}E^* - 24IGP\text{-}E^* \\
\dot{Ind\text{-}G3P\text{-}E^*} &= 24IGP\text{-}E^* + 2G3P\text{-}E \cdot Ind - 20Ind\text{-}G3P\text{-}E^* - 11Ind\text{-}G3P\text{-}E^* \\
\dot{G3P\text{-}E} &= 20Ind\text{-}G3P\text{-}E^* + 0.2G3P \cdot E - 2G3P\text{-}E \cdot Ind - 200G3P\text{-}E \cdot Ind \\
\dot{G3P} &= 200G3P\text{-}E - 0.2G3P \cdot E \\
\dot{Ind} &= 20Ind\text{-}G3P\text{-}E^* - 2G3P\text{-}E \cdot Ind - 2EAA \cdot Ind \\
\dot{Ser} &= 20E\text{-}Ser - 0.135Ser \cdot E \\
\dot{E\text{-}Ser} &= 0.135Ser \cdot E + 10EAA - 20E\text{-}Ser - 45E\text{-}Ser \\
\dot{EAA} &= 45E\text{-}Ser - 10EAA - 2EAA \cdot Ind \\
\dot{E^*AA\text{-}Ind} &= 2EAA \cdot Ind + 0.1E^*\text{-}Trp - 1000E^*AA\text{-}Ind \\
\dot{EsTrp} &= 1000E^*AA\text{-}Ind + 0.5E \cdot Trp - 0.1E^*\text{-}Trp - 8E^*\text{-}Trp \\
\dot{Trp} &= 8E^*\text{-}Trp - 0.5E \cdot Trp
\end{aligned}
$$

Fig. 2. Continuous macro model of the tryptophan synthase

Summarizing the model it is clear that:

- it is a continuous systems model since the equation system 4 describes continuous change rates of the metabolite concentrations via balancing the reaction velocity terms for the building and decay of the different entities;
- it is a macro-model – as the model contains only one level and no individual entities are modeled, but homogeneous populations of the entities are regarded as one variable and all variables are attributed to the same entity: the tryptophan synthase system;
- it is deterministic – as equations 4 do not contain stochastic elements, like distribution functions for inter-arrival times or the different enzyme-metabolite populations (IGP-E, etc.);
- it is quantitative - as the state space is a real-valued vector (IGP-E, etc.);

As mentioned above, the model is particularly suited for the description of the experimentally determined concentration changes. Nevertheless some known structural characteristics of the enzyme Tryptophan Synthase are not reflected equally well. Especially the macro-models' description of the hydrophobic channel is strongly simplified: it is known from independently performed X-Ray experiments for the structural analysis, that the tunnel can store up to four indole molecules. This could imply a time delay for tunneling the indole from the α- to the corresponding β-subunit, that was not taken into account by the macro-model. Integrating the tunnel's capacity into the described macro model would significantly complicate the model's execution, since time-delayed differential equations have to be defined. In addition, it would burden the model structure reducing the transparency of the model.

Therefore a discrete-event stochastic multi-level model was generated [100] to allow a more detailed description of the individual enzymes including additional structural (qualitative) information about the enzymes and to allow at the same time the reproduction of the *in-vitro* experiments. In spite of the additional com-

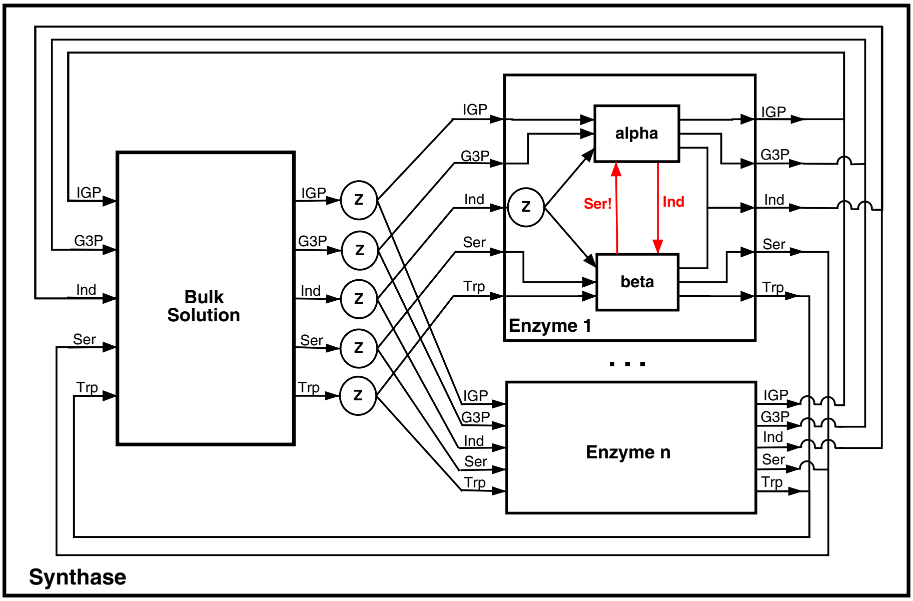


Fig. 3. A multi-level model of tryptophan synthase [29]

plexity the multi-level model should remain transparent for the experimentalists, see figure 3.

In the following the conversion of the continuous deterministic macro-model into the discrete-event stochastic multi-level model is discussed in more detail step by step:

1. Transformation of a *continuous* deterministic macro-model to a *discrete-event* stochastic macro-model:
   - this is e.g. done by the Gillespie Algorithm [74], that was deduced to exactly simulate a stochastic differential equation system describing chemical reactions systems. Depending on the actual numbers of each molecule and enzymes in each binding-state at a discrete time, the algorithm deduces, when the next reaction will take place and what reaction it will be. After that time the number of molecules and enzymes with different binding-states is updated and the next reaction time is determined;
2. Transformation of a discrete-event stochastic *macro model* to a step-wise-discrete stochastic *micro model*:
   - to form a micro model from the discrete stochastic macro-model individual entities with their properties, i.e. in our case mainly the different binding-states of the enzymes, have to be modeled;
   - corresponding simulations can be realized in StochSim: at each time step of the simulation, which is determined by the fastest reaction step,

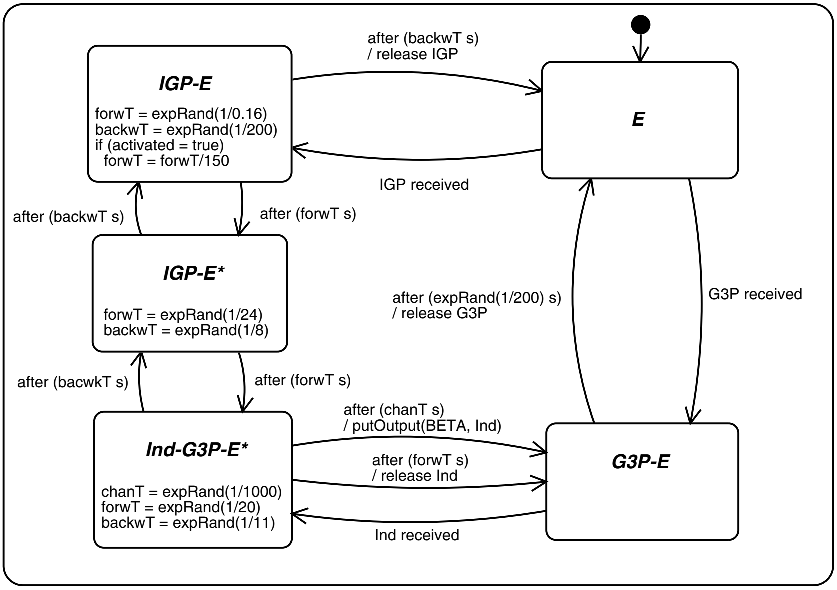


Fig. 4. Phases and transitions for an α-subunit of a single enzyme [29]

i.e. in our case the tunneling reaction, two molecules were randomly chosen. Their current properties, i.e. binding-states, decide whether a reaction can take place at all. According to the probability defined for this reaction (which is correlated to the reaction's velocity) the reaction will actually be executed;

- the simulation is time-consuming due to the many time-steps, at which no reaction takes places;

3. Transformation of a *step-wise-discrete* stochastic *micro model* to a *discrete-event* stochastic *multi-level model*:

   - micro to multi-level: The individual enzyme with its different binding-states is subdivided into an α- and a β-subunit, which communicate via the tunnel.

     In addition to the individual enzymes, a macro-level is introduced which records the IGP, Ser, Trp, Ind, G3P molecules and the α- β-models in the bulk solution. It has the function to distribute the substrate and product molecules to and from the individual enzymes. The frequency is determined by the current concentration of metabolites and enzymes and the velocity of the different reactions. It offers a macro perspective on the system, where the bulk solution is the system of interest having concentrations and equations for describing the change of concentrations attributed to it. As the bulk solution contains only metabolites that are of interest for the tryptophan synthase, each change of concentration is translated into forwarding metabolites to individual enzymes. In addition, the macro level records the trajectories of the metabolite concentrations over time and can be used for validating the model based on the *in-vitro* experiments.

- step-wise-discrete to discrete-event: the internal state, i.e. the binding-state, of the α/β-model determines whether a reaction will take place and when, this is defined according to the Gillespie algorithm (Figure 4).

5 Different Perspectives of Multi-level Models

By adopting the object metaphor, the number of discrete and continuous simulation systems that integrate the different traditional views in modeling systems, i.e. as functional models, as networks of interactions, and as hierarchical composition of models is steadily increasing, e.g. James [107], GenomicObjectnet, [108], e-cell [109], and BiosPI [110]. Thereby, composition and interaction determine the overall structure of a model in general and of a multi-level model in particular.

At the lowest layer we find functional models of individuals. They might be represented as quantitative, or qualitative, continuous or discrete model, integrating stochastic aspects or describing the system's behavior deterministically. To define the interaction between models, interfaces have to be defined. To let models interact it is important to distinguish between so called "value couplings" that support a direct exchange of values, so each change in one submodel is directly reflected to a change in another submodel, and an exchange of values by events. Whereas the former supports the coupling of continuous models the latter is used to support the coupling of discrete systems models. A combination of both allows to support the coupling of hybrid models. To facilitate the interaction, often the interaction of hybrid submodels is restricted to exchanging discrete events at discrete times [111, 112].

Grouping stongly interacting submodels into one model supports a hierarchical composition of models. Thus, a compositional hierarchy is introduced bottom up. Similarly we can assume that a hierarchy is introduced top down by starting with the coupled or composite model and asking for its components. Most modeling formalisms assume a strong composition, i.e. one model component belongs only to one coupled model. Often coupled models or composite models simply frame a group of models so that they can be treated as one model, e.g. as it is the case in Devs, in composite hierarchical Petri Nets, like the GenomicObjectNet [113], and in BioAmbients [55]. To belong to the components of such a coupled model can easily be interpreted as residing in one space. This view is emphasized in BioAmbients [55], which, based on the stochastic π-Calculus, is directed toward supporting higher level abstractions and the description of complex, spatial phenomena in Systems Biology.

If a coupled model is interpreted as representing a spatial cell compartment, the ability to support variable structure models, i.e., models that are able to change their own composition and interaction structure [45], becomes a prerequisite to describe phenomena of proteins joining and leaving cell compartments. Composite models have no behavior of their own, their behavior is spec-

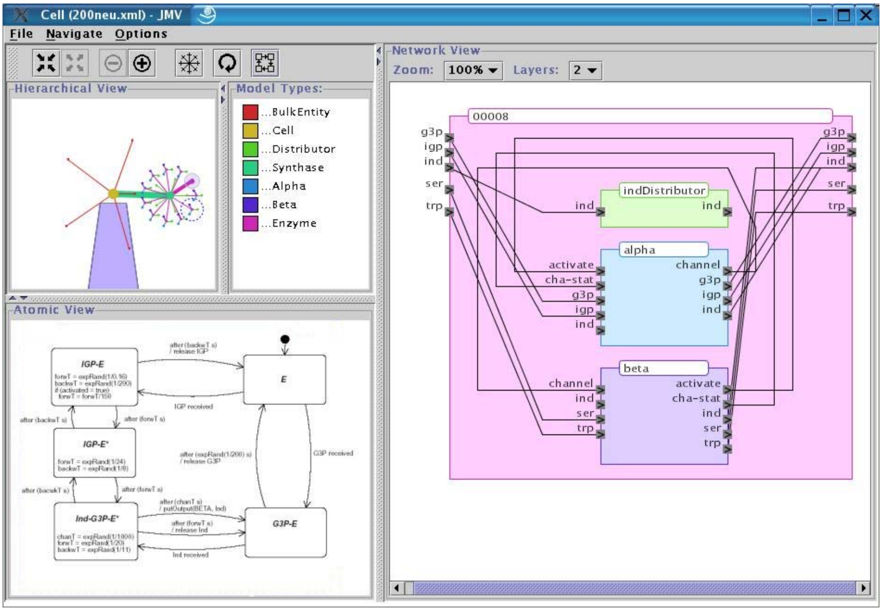


Fig. 5. Screen shot explicating the atomic, the network and the hierarchy perspective in multi-level modeling [114]

ified by their components and their interactions. This lack of own state and behavior does not hamper to use them to introduce a notion what does and what does not belong to a single cell compartment. However, to explicitly describe a macro view, a separate model has to be introduced to describe state and dynamic at the macro level, as has been done to model the Tryptophan synthase in James (Fig. 5). The multi-level model contains sub-models that describe enzymes as micro models, and sub-models that describe the behavior of entire enzyme populations as macro models.

Figure 5 represents the different perspectives in modeling. The macro level contains models that describe the state and dynamics of the different populations of the bulk solution (see also figure 3). The macro models responsible for the indole, the serine, the IGP, and the G3P interact with the "micro model" responsible for the synthase. The former keeps track of the amounts of substrates, products and enzymes and defines the behavior at the level of concentrations and collision probability.

The micro model synthase contains thousands of models each of which describes a single enzyme, (figure 5 in upper left corner). Thus, the overall composition tree is highly unbalanced, one of the children has more than 800 children. As we are interested in the role the channel plays in the tryptophan synthase, we define the enzyme model to consist of two different subunits, i.e. α and β, which communicate via the channel (see figure 5 on the right hand side). The behavior of each subunit is modeled as discrete transitions from one state to the other. State changes might be triggered by the arrival of metabolites or by the flow of time (see figure 5 in the lower, left corner).

Multi-level models promise a flexible approach toward the understanding of cellular systems. However, they also provide new challenges for modeling, simulation, and visualization techniques, alike – which is illustrated in the above figure. Different perspectives on the model, that can be interactively selected and refined, are needed to visualize the model structure in a compact manner and to enable users to rapidly manipulate the model structure [114].

6 Discussion

Thinking about variables and their continuous change rates appears closely related to a macro perception rather than a micro perception of a system. Continuous models reflect the observation of experiments in cellular biology nicely. The starting point of multi-level models seems somehow different. It is focused on the active entities of the processes. Their states, behavior, and interaction with others are directly described. In continuous models the structural information are indirectly deducible from the model parameter and the structure of the differential equations. Though continuous models can easily be structured into components or objects to describe a system as being comprised of interacting subsystems, often the focus is on the global scheme of reaction mechanisms.

Discrete modeling approaches prevail if single entities and their dynamics shall be described. Since many phenomena can only be measured on population level, models of single enzymes are typically only checked for plausibility. For validation model populations can be created, thus turning to micro models embracing many individuals. To consider the slight deviation between individuals stochastic effects are introduced supporting realistic phenomena on macro level. Individual-based models combine a macro and micro view on the system under study. They form a first step toward multi-level models, where different description levels of systems are integrated. Interaction and coordination are taking place within and between levels of organization. Modularity, hierarchical structure, and flexible modeling come natural to multi-level models, however at the cost of simulation efficiency requiring special solutions [115]. There however, appears to be no silver bullet for modeling cellular systems. The objective of the simulation study should drive the level of resolutions chosen, structure, and the formalism employed. The background of the modelers might bias the choice of approach unintentionally. Therefore they should educate themselves to appreciate the variety of choices that have become available in the last several years.

The multi-level modeling approach offers a way of bridging between micro and macro level constructs. The concept of homomorphism has been proposed as the way to express macro level constructs in terms of micro level ones in a way that preserves their behavior. Although several examples have been developed to illustrate this approach [44, 116, 117], more research and more attempts to apply the research results are sorely needed. Advances in modeling, simulation, and computational biology in general, may well hinge on achieving better ways to include multiple levels of resolution. Multi-level models move the focus of modeling and simulation from seeking the most simplistic model able to reproduce the observed data, to a flexible, easily refinable and re-usable "middle-out" model-design that suits the structure of our knowledge and the current question of interest best.

Acknowledgement

We would like to thank Celine Kuttler, Mathias Röhl, and the anonymous reviewer for their comments on a previous version of this paper. Part of this research has been funded by the DFG.

References

1. Kitano, H.: Systems Biology: A Brief Overview. Science **295** (2002) 1662–1664
2. Wolkenhauer, O.: Systems biology: the reincarnation of systems theory applied in the biology? Briefings in Bioinformatics **2** (2001) 258–270
3. Chabrier-Rivier, N., Fages, F., Soliman, S.: The Biochemical Abstract Machine Biocham. In: Proceedings of the 2nd International Workshop on Computational Methods in Systems Biology. (2004)
4. Hucka, M., Finney, A., Sauro, H., Bolouri, H.: The erato systems biology workbench: Architectural evolution. In Yi, T.M., Hucka, M., Morohashi, M., Kitano, H., eds.: The Proceedings of the 2nd International Conference on Systems Biology. (2001)
5. Hucka, M., Finney, A., Sauro, H.M., Bolouri, H., Doyle, J.C., Kitano, H., Arkin, A., Bornstein, B.J., Bray, D., Cornish-Bowden, A., Cuellar, A.A., Dronov, S., Gilles, E.D., Ginkel, M., Gor, V., Goryanin, I.I., Hedley, W.J., Hodgman, T., Hofmeyr, J.H., Hunter, P.J., Juty, N., Kasberger, J.L., Kremling, A., Kummer, U., Le Novere, N., Loew, L.M., Lucio, D., Mendes, P., Minch, E., Mjolsness, E.D., Nakayama, Y., Nelson, M.R., Nielsen, P.F., Sakurada, T., Schaff, J.C., Shapiro, B., Shimizu, T.S., Spence, H.D., Stelling, J., Takahashi, K., Tomita, M., Wagner, J., Wang, J.: The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics **19** (2003) 524–531
6. Cuellar, A., Lloyd, C., Nielsen, P., Bullivant, D., Nickerson, D., Hunter, P.: An overview of CellML: 1.1, A Biological Model Description Language. Simulation - Transactions of the SCS **79** (2003) 740–747
7. Domach, M.M., Leung, S.K., Cahn, R.E., Cocks, G.G., Shuler, M.L.: Computer model for glucose-limited growth of a single cell of *Escherchia coli* B/r-A. Biotechnology and Bioengineering **26** (1984) 203–216
8. Teusink, B., Passarge, J., Reijenga, C.A., Esgalhado, E., van der Weijden, C.C., Schepper, M., Walsh, M.C., Bakker, B.M., van Dam, B., van Dam, K., Westerhoff, H.V., Snoep, J.L.: Can yeast glycolysis be understood in terms of *in vitro* kinetics of the constituent enzymes? Testing biochemistry. European Journal of Biochemistry **267** (2000) 5313–5329
9. Hynne, F., Donø, S., Sørenson, P.G.: Full-scale model of glycolysis in *Saccharomyces cerevisiae.* Biophysical Chemistry **94** (2001) 121–163

10. Santillán, M., Mackey, M.C.: Dynamic regulation of the tryptophan operon: A modeling study an comparison with experimental data. Proceedings of the National Academy of Sciences of the USA **98** (2001) 1364–1369
11. Reddy, V.N., Liebman, M.N., Mavrovouniotis, M.L.: Qualitative analysis of biochemical reaction systems. Computers in Biology and Medicine **26** (1996) 9–24
12. Xia, X.Q., Wise, M.J.: DiMSim: A Discrete-Event Simulator of Metabolic Networks. Journal of Chemical Information and Computer Science **43** (2003) 1011–1019
13. Jones, M.E., Berry, M.N., Phillips, J.W.: Futile Cycles Revisited: A Markov Chain Model of Simultaneous Glycolysis and Gluconeogenesis. Journal of Theoretical Biology **217** (2002) 509–523
14. Arkin, A., Ross, J.: Computational functions in biochemical reaction networks. Biophysical Journal **67** (1994) 560–578
15. Hjemfelt, A., Ross, J.: Implementation of logic functions and computations by chemical kinetics. Physica D **84** (1995) 180–193
16. Bentele, M., Eils, R.: General stochastic hybrid method for the simulation of chemical reaction processes in cells. In: Proceedings of the 2nd International Workshop on Computational Methods in Systems Biology. (2004)
17. Zeigler, B., Praehofer, H., T.G., K.: Theory of Modeling and Simulation. Academic Press, London (2000)
18. Heylighen, F. In: Downward Causation. Principia Cybernetica Web, http://pespmc1.vub.ac.be/DOWNCAUS.html (access date: 12.05.2004)
19. Bunge, M.: Ontology II: A World of Systems. Volume 4 of Treatise of Basic Philosophy. Reidel, Dordrecht (1979)
20. Campbell, D.: Downward causation in Hierarchically Organized Biological Systems. In Ayala, F., Dobzhanzky, J., eds.: Studies in the Philosophy of Biology. University of California Press, Berkeley (1974) 179–186
21. Salthe, S.: Evolving Hierarchical Systems. Columbia University Press (1985)
22. Strohmann, R.: Organization becomes cause in the matter. Nature Biotechnology **18** (2000) 575–576
23. Whitesides, G., Boncheva, M.: Beyond molecules: Self-assembly of mesoscopic and macroscopic components. PNAS **99** (2002) 4769–4774
24. Hartwell, L.H., Hopfield, J.J., Leibler, S., Murray, A.W.: From molecular to modular cell biology. Nature **402** (1999) C47–C52
25. Vilar, J.M.G., Guet, C.C., Leibler, S.: Modeling network dynamics: the lac operon, a case study. The Journal of Cell Biology **161** (2003) 471–476
26. Kremling, A., Jahreis, K., Lengeler, J.W., Gilles, E.D.: The Organization of Metabolic Reaction Networks: A Signal Oriented Approach to Cellular Models. Metabolic Engineering **2** (2000) 190–200
27. Kremling, A., Gilles, E.D.: The Organization of Metabolic Reaction Networks II. Signal Processing in Hierarchical Structured Functional Units. Metabolic Engineering **3** (2001) 138–150
28. Kremling, A., Bettenbrock, K., Laube, B., Jahreis, K., Lengeler, J.W., Gilles, E.D.: The Organization of Metabolic Reaction Networks: III. Application for Diauxic Growth on Glucose and Lactose. Metabolic Engineering **3** (2001) 362–379
29. Degenring, D., Röhl, M., Uhrmacher, A.: Discrete Event, Multi-Level Simulation of Metabolite Channeling. BioSystems **75** (2004) 29–41
30. Minsky, M.: Models, Minds, Machines. In: Proc. IFIP Congress. (1965) 45–49
31. Cellier, F.E.: Continuous System Modeling. Springer, New York (1992)

32. de Jong, H.: Modeling and Simulation of Genetic Regulatory Systems: A Literature Review. Journal of Computational Biology **9** (2002) 67–103
33. Mendes, P.: GEPASI: a software package for modelling the dynamics, steady states and control of biochemical and other systems. Computer Applications in the Biosciences **9** (1993) 563–571
34. Ginkel, M., A., K., Nutsch, T., Rehner, R., Gilles, E.: Modular modeling of cellular systems with ProMoT/Diva. Bioinformatics **19** (2003) 1169–1176
35. Sauro, H.: Jarnac: A system for interactive metabolic analysis. In: Animating the cellular map: Proceedings of the 9th International Meeting on BioThermoKinetics, Stellenbosch University Press (2000)
36. Goryanin, I., Hodgman, T., Selkov, E.: Mathematical simulation and analysis of cellular metabolism and regulation. Bioinformatics **15** (1999) 749–758
37. Shapiro, B.E., Levchenko, A., Meyerowitz, E.M., Wold, B.J., Mjolsness, E.D.: Cellerator: extending a computer algebra system to include biochemical arrows for signal transduction simulations. Bioinformatics **19** (2003) 677–678
38. Fuss, H.: Simulation of Biological Systems with PetriNets - Introduction to Modelling of Distributed Systems. In Moller, D., ed.: Advances in System Analysis. Vieweg, Braunschweig, Wiesbaden (1987) 1–12
39. Goss, P., Peccoud, J.: Biochemistry Quantitative Modeling of Stochastic Systems in Molecular Biology by Using Stochastic Petri Nets. Proceedings of National Academy of Sciences of the USA **95** (1998) 6750–6755
40. Zeigler, B.: Multifacetted Modelling and Discrete Event Simulation. Academic Press, London (1984)
41. : Petri Nets World. http://www.daimi.au.dk/PetriNets/ (access date: 08.11.2004)
42. Milner, R.: Communicating and Mobile Systems: The π Calculus. Cambridge University Press (1999)
43. Priami, C.: The Stochastic pi-Calculus. The Computer Journal **38** (1995) 578–589
44. Zeigler, B.: A Note on System Modelling, Aggregation and Reductionism. J. of Biomedical Computing **2** (1971) 277–280
45. Uhrmacher, A.: Dynamic Structures in Modeling and Simulation - A Reflective Approach. ACM Transactions on Modeling and Simulation **11** (2001) 206–232
46. Uhrmacher, A.M.: Reasoning about Changing Structure: A Modeling Concept for Ecological Systems. International Journal on Applied Artificial Intelligence **9** (1995) 157–180
47. Kam, N., Harel, D., Kugler, H., Marelly, R., Pnueli, A., Hubbard, E., Stern, M.: Formal Modelling of *C. elegans* Development: A Scenario Based Approach. In C., P., ed.: Computational Methods in Systems Biology. Volume 2602 of Lecture Notes in Computer Science., Springer Verlag Heidelberg (2003) 3–20
48. Borland, S., Vangheluwe, H.: Transforming Statecharts to DEVS. In: Summer Computer Simulation Conference. (2003) 154–159
49. Danos, V., Laneve, C., eds.: BioConcur - Workshop on Concurrent Models in Molecular Biology, Electronic Notes in Theoretical Computer Science (2003)
50. Kuttler, C., Blossey, R., Niehren, J.: Gene Regulation in the Pi Caluculus: Modelling Cooperativity at the Lambda Switch. In: BioConcur 2004, Elsevier (2004)
51. Regev, A., Shapiro, E.: Cells as computation. Nature **419** (2002) 343 www.wisdom.weizmann.ac.il˜aviv.
52. Lecca, P., Priami, C., Quaglia, P., Rossi, B., Laudanna, C., Constantin, G.: Language Modelling and Simulation of Autoreactive Lymphocytes Recruitment in Inflamed Brain Vessels. SCS Simulation (Submitted)

53. Van Gend, K., U., K.: STODE - Automatic Stochastic Simulation of Systems Described by [differential equations. In Yi, T.M., Hucka, M., Morohasi, M., Kitano, H., eds.: Proceedings of the 2nd International Conference on Systems Biology, Omnipress, Madison, USA (2001) 326–333
54. Philipps, A., Cardelli, L.: A correct abstract machine for the stochastic pi-calculus. In: Proc. of BIO-CONCUR'04. Electronic Notes in Theoretical Computer Science, Elsevier (2004)
55. Regev, A., Panina, E., Silverman, W., Cardelli, L., Shapiro, E.: BioAmbients: An Abstraction for Biological Compartments. Theoretical Computer Science (2004)
56. Cardelli, L.: Brane Calculi. In: Proc. of BIO-CONCUR'03. Electronic Notes in Theoretcial Computer Science, Elsevier (2003)
57. Mc Collum, J., Cox, C., Simpson, M., Peterson, G.: Accelerating Gene Regulatory Network Modeling Using Grid-Based Simulation. Simulation - Transactions of the SCS (2004)
58. Danos, V., Pradalier, S.: Projective brane calculus. In: Proceedings of the 2nd International Workshop on Computational Methods in Systems Biology. (2004)
59. Zeigler, B.: Discrete Event Abstraction: An Emerging Paradigm For modeling complex adaptive systems. In Booker, L., ed.: Perspectives on Adaptation in Natural and Artificial Systems - Essays in Honor of John Holland,. Oxford University Press (2004)
60. Nutaro, J., Zeigler, B., Jammalamadaka, R., Akerkar, S.: Speeding-Up the Simulation of Continuous Systems with Parallel DEVS:A Gas Shock Wave Example. In Darema, F., ed.: Dynamic Data Driven Applications Systems. Academic Publishers (2004)
61. Chen, M., Hofestädt, R., Freier, A.: A Workable Approach for Modeling and Simulation of Biochemical Processes with Hybrid Petri Net System. In: 1st International MTBio Workshop on Function and Regulation of Cellular Systems: Experiments and Models,, Dresden (2001)
62. Matsuno, H., Fujita, S., Doi, A., Nagasaki, M., Miyano, S.: Towards biopathway modeling and simulation. In VanDerAalst, W., Best, E., eds.: Applications and Theory of Petri Nets. Volume 2679 of Lecture Notes in Computer Science. (2003) 3–22
63. Cho, K.H., Johansson, K., Wolkenhauer, O.: A Hybrid Systems Framework for Cellular Processes. submitted for publication (2004)
64. : Matlab Simulink. http://www.mathworks.com (access date: 08.11.2004)
65. Liu, J., Lee, E.: A component-based approach to modeling and simulating mixed-signal and hybrid systems. ACM Transactions on Modeling and Computer Simulation **12** (2002) 343–368
66. Henzinger, T.: The theory of hybrid automata. In: Proceedings of the 11th Annual Symposium on Logic in Computer Science (LICS), IEEE Computer Society Press (1996) 278–292
67. Alur, R., Belta, C., Ivancic, F., Kumar, V., Rubin, H., Schug, J., Sokolsky, O., Webb, J.: Visual programming for modeling and simulation of biomolecular regulatory networks. In: International Conference on High Performance Computing. (2002)
68. Mishra, B., Policriti, A.: Systems Biology and Automata. In: 3rd Workshop on Computation of Biochemical Pathways and Genetic Networks, Heidelberg, Springer Verlag (2003)

69. Belta, C., Finin, P., Habets, L., Halasz, A., Irnieliniksi, M., Kurnar, V., Rubin, H.: Understanding the bacterial stringent response using reachability analysis of hybrid systems. In: Lecture Notes in Computer Science. Volume 2993. Springer (2004)
70. Law, A., Kelton, W.: Simulation, Modeling, and Analysis. MCGraw Hill International Editions, New York (1991)
71. Rao, C.V., Wolf, D.M., Arkin, A.P.: Control, exploitation and tolerance of intracellular noise. Nature **420** (2002) 231–237
72. Fedoroff, N., Fontana, W.: Small Numbers of Big Molecules. Science **297** (2002) 1129–1131
73. Gibson, M.A., Bruck, J.: EfficientExact Stochastic Simulation of Chemical Systems with Many Species and Many Channels. Journal of Physical Chemistry A **104** (2000) 1876–1889
74. Gillespie, D.T.: A General Method for Numerically Simulating the Stochastic Time Evolution of Coupled Chemical Reactions. The Journal of Physical Chemistry B **22** (1976) 403–434
75. Gillespie, D.T.: Exact Stochastic Simulation of Coupled Chemical Reactions. The Journal of Physical Chemistry B **81** (1977) 2340–2361
76. Kuo, D., Keasling, J.D.: A Monte Carlo simulation of plasmid replication during the bacterial division cycle. Biotechnology and Bioengineering **52** (1996) 633–647
77. Kierzek, A.M.: STOCKS: STOChastic Kinetic Simulations of biochemical systems with Gillespie algorithm. Bioinformatics **18** (2002) 470–481
78. Cowan, R.: Stochastic models for DNA replication. In Shanbhag, D., Rao, C., eds.: Stochastic Processes. Handbook of Statistics. (2003)
79. Gillespie, D.T.: Approximate accelerated stochastic simulation of chemically reacting systems. The Journal of Chemical Physics **115** (2001) 1716–1733
80. Gillespie, D.T., Petzold, L.R.: Improved leap-size selection for accelerated stochastic simulation. The Journal of Chemical Physics **119** (2004) 8229–8234
81. Puchulka, J., Kierzek, A.M.: Bridging the Gap between Stochastic and Deterministic Regimes in the Kinetic Simulations of the Biochemical Reaction Networks. Biophysical Journal **86** (2004) 1357–1372
82. Kuipers, B.: Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge. MIT Press (1994)
83. Heidtke, K.R., Schulze-Kremer, S.: Design and implementation of a qualitative simulation model of λ phage infection. Bioinformatics **14** (1998) 81–91
84. Ideker, T., Lauffenburger, D.: Building with a scaffold: emerging strategies for high- to low-level cellular modeling. Trends in Biotechnology **21** (2003) 255–262
85. Thomas, R., Kaufman, M.: Multistationarity, the basis of cell differentiation and memory. I. structural conditions of multistationarity and other nontrivial behavior. Chaos **11** (2001) 170–179
86. Thomas, R., Kaufman, M.: Multistationarity, the basis of cell differentiation and memory. II. Logical analysis of regulatory networks in terms of feedback circuits. Chaos **11** (2001) 180–195
87. Tilly, C.: Micro, Macro, or Megrim? Paper for the Göttinger Gespräch zur Geschichtswissenschaft, Microhistory - Macrohistory: Complementary or Incommensurable ? (1997)
88. Knorr-Cetina, K., Cicourel, A., eds.: Advances in Social Theory and Methodology - Towards an Integration of Micro and Macro Sociologies. Routledge and Kegan Paul, Boston (1981)

89. Troitzsch, K.: Multilevel Simulation. In Troitzsch, K., Mueller, U., Gilbert, G., Doran, J., eds.: Social Science Microsimulation. Springer (1996) 107–120
90. Kokai, G., Toth, Z., Vanyi, R.: Modelling blood vessels of the eye with parametric L-systems using evolutionary algorithms. In Horn, W., Shahar, Y., Lindberg, G., Andreassen, S., Wyatt, J., eds.: Artificial Intelligence in Medicine. Volume 1620 of Lecture Notes in Artificial Intelligence. (1999) 433–442
91. Garcia-Olivares, A., Villarroel, M., Marijuan, P.C.: Enzymes as molecular automata: a stochastic model of self-oscillatory glycolytic cycles in cellular metabolism. Biosystems **56** (2000) 121–129
92. Wurthner, J., Mukhopadhyay, A., Piemann, C.: A cellular automaton model of cellular signal transduction. Computers in Biology and Medicine **30** (2000) 1–21
93. Alber, M., Kiskowski, M., Glazier, J.A., Jiang, Y.: On cellular automaton approaches to modeling biological cells. In Rosenthal, J., Gilliam, D.S., eds.: Mathematical Systems Theory in Biology, Communications, Computation and Finance. Volume 134 of IMA Volumes in Mathematics and its Applications. (2003) 1–39
94. Kniemeyer, O., Buck-Sorlin, G.H., Kurth, W.: Representation of genotype and phenotype in a coherent framework based on extended L-systems. In Banzhaf, W., Christaller, T., Dittrich, P., Kim, J.T., Ziegler, J., eds.: Advances in Artificial Life. Volume 2801 of Lecture Notes in Artificial Intelligence. (2003) 625–634
95. Swameye, I., Müller, T., Timmer, J., Sandra, O., Klingmüller, U.: Identification of nucleocytoplasmic cycling as a remote sensor in cellular signaling by data-based dynamic modeling. PNAS **100** (2003) 1028–1033
96. Uhrmacher, A.M., Swartout, W.: Agent-Oriented Simulation. In Obaidat, M., Papadimitriou, G., eds.: Applied System Simulation, Amsterdam, Kluwer Academic Press (2003)
97. Uhrmacher, A., Degenring, D.: From macro- to Multi-Level Models in Systems Biology. In Gauges, R., Kummer, U., Pahle, J., Rost, U., eds.: Proc. of the 3rd Workshop on Computation of Biochemical Pathways and Genetic Networks. (2003)
98. Kreft, J., Booth, G., Wimpenny, J.: BacSim a simulator for individual based modelling of bacterial colony growth. Microbiology **144** (1998) 3275–3287
99. Gregory, R.: An Individual Based Model for Simulating Bacterial Evolution. In: Evolvability and Individuality Workshop, University of Hertfordshire (2002)
100. Degenring, D., Röhl, M., Uhrmacher, A.M.: Discrete event simulation for a better understanding of metabolite channeling- A system-theoretic approach. In C., P., ed.: Computational Methods in Systems Biology. Volume 2602 of Lecture Notes in Computer Science., Springer Verlag Heidelberg (2003) 114–126
101. Rizzi, M., Baltes, T., Theobald, U., Reuss, M.: In Vivo Analysis of Metabolic Dynamics in *Saccheromyces cerevisiae* II. Mathematical Model. Biotechnology and Bioengineering **55** (1997) 592–608
102. Takahashi, K., Yugi, K., Hashimoto, K., Yamada, Y., Pickett, C., Tomita, M.: Computational challenges in cell simulation. IEEE Intelligent Systems **17** (2002) 64–71
103. Henson, M., Müller, D., Reuss, M.: Cell Population Modelling of Yeast Glycolytic Oscillations. Biochemical Journal **368** (2002) 433–446
104. Morton-Firth, C.J., Bray, D.: Predicting Temporal Fluctuations in an Intracellular Signalling Pathway. Journal of Theoretical Biology **192** (1998) 117–128
105. Anderson, K., Miles, E., Johnson, K.: Serine Modulates Substrate Channeling in Tryptophan Synthase. The Journal of the Biological Chemistry **266** (1991) 8020–8033

106. Anderson, K., Kim, A., Quillen, J., Sayers, E., Yand, X., Miles, E.: Kinetic Characterization of Channel Impaired Mutants of Tryptophan Synthase. The Journal of Biological Chemistry **270** (1995) 29936–29944

107. Uhrmacher, A.M., Tyschler, P., Tyschler, D.: Modeling Mobile Agents. Future Generation Computer System **17** (2000) 107–118

108. Elmquist, H., Mattson, S.: Modelica - The Next Generation Modeling Language - An International Design Effort. In: First World Congress of System Simulation, Singapore (1997)

109. Takahashi, K., Kaizu, K., Hu, B., Tomita, M.: A multi-algorithm, multi-timescale method for cell simulation. Bioinformatics **20** (2004) 538–546

110. : Biospi simulator. http://www.wisdom.weizmann.ac.il˜biospi/ (access date: Okt. 2004)

111. Lynch, N., Segala, R., Vaandraager, F.: Hybrid I/O automata. Technical Report MITLCS-TR-827d, MIT Laboratory for Computer Science (2003)

112. : Anylogic - Simulation Software. http://www.xjtek.com/anylogic/ (access date: May 2004)

113. Nagasaki, M., Doi, A., Matsuno, H., Miyano, S.: Genomic Object Net: A platform for modeling and simulating biopathways. Applied Bioinformatics (2003)

114. Biermann, S., Uhrmacher, A., Schumann, H.: Supporting Multi-Level Models in Systems Biology by Visual Methods. In: Proceedings of European Multi-Simulation Conference. (2004)

115. Fujimoto, R.: Parallel and Distributed Simulation Systems. John Wiley and Sons (2000)

116. Zeigler, B.: Statistical Simplification of Neural Nets. Intl. J. of Machine Studies **7** (1975) 371–393

117. Zeigler, B.: Simplification of Biochemical Systems. In Segel, L., ed.: Mathematical Models in Molecular and Cellular Biology, Cambridge University Press (1981)

A New Time-Dependent Complexity Reduction Method for Biochemical Systems

Jürgen Zobeley[1], Dirk Lebiedz[2], Julia Kammerer[2], Anton Ishmurzin[2],
and Ursula Kummer[1]

[1] EML Research gGmbH, Schloss-Wolfsbrunnenweg 33,
69118 Heidelberg, Germany
juergen.zobeley@eml-r.villa-bosch.de
[2] IWR, University of Heidelberg, Im Neuenheimer Feld 368,
69120 Heidelberg, Germany

Abstract. Systems biology aims at an understanding of increasingly large and complex cellular systems making use of computational approaches, e.g. numerical simulations. The size and complexity of the underlying biochemical reaction networks call for methods to speed up simulations and/or dissect the biochemical network into smaller subsystems which can be studied independently. Both goals can be achieved by so-called complexity reduction algorithms. However, existing complexity reduction approaches for biochemical reaction networks are mostly based on studying the steady state behavior of a system and/or are based on heuristics. Given the fact that many complex biochemical systems display highly nonlinear dynamics and that this dynamics plays a crucial role in the functioning of the organism, a new methodology has to be developed. Therefore, we present a new complexity reduction method which is time-dependent and suited not only for steady states, but for all possible dynamics of a biochemical system. It makes use of the evolution of the different time–scales in the system, allowing to reduce the number of equations necessary to describe the system which is speeding up the computation time. In addition, it is possible to study the way different variables/metabolites contribute to the reduced equation system which indicates how strongly they interact and couple. In the extreme case of variables decoupling in a specific state, the method allows the complete dissection of the system resulting in subsystems that can be studied in isolation. The whole method provides a systematic tool for an automated complexity reduction of arbitrary biochemical reaction networks. With the aid of a specific example, the oscillatory peroxidase-oxidase system, we show that coupling of time–scales depends heavily on the specific dynamics of the system. Therefore, neither computational improvement nor systematic understanding can be achieved by studying these aspects solely under steady state conditions.

1 Introduction

Improved experimental techniques enable researchers to study molecules and their concentrations in living cells instead of separately in test tubes. In addition, high-throughput techniques allow for the massive accumulation of data.

C. Priami et al. (Eds.): Transactions on Computational Systems Biology, LNBI 3380, pp. 90–110, 2005.

These developments have led to a growing interest in systems biology which aims at an understanding of increasingly complex biochemical systems *in vivo*. This understanding can only be achieved if quantitative experimental technologies are accompanied by computational research, not only because of the amount of data, but also because of the complexity of the underlying biochemical networks. Therefore, more and more computational techniques for this purpose have been developed in the last few years [1].

One important aspect of computational research in systems biology is the development of methods for complexity reduction of the systems. Complexity reduction is used in two different ways. First of all, complexity reduction is aiming at an increased speed for simulations. This is usually achieved by mathematically reducing the number of equations necessary to describe the system which results in a facilitated simulation. However, a reduced number of equations does not necessarily mean an accompanying reduction of the biochemical species in the system, since many different species might contribute to one equation after the mathematical transformation. Therefore, a likewise or even more important aspect is that complexity reduction is needed in order to reduce the biochemical system by dissecting it into several modules which can be studied independently. This is needed to understand the interplay of specific subsystems and facilitates research on these defined subsystems.

Rational and automatic approaches to both of the above issues should ensure that complexity reduction will not lead again to a limited understanding (focusing on arbitrary subsystems), but rather to an increased understanding which enables researchers to determine in detail how and when subsystems interlink with each other.

Past approaches to complexity reduction of biochemical systems have focused mainly on methods studying the steady state behavior of the system [2, 3] and on dissecting the system based on its network topology using heuristic rules [4, 5]. The first approach is valid and helpful for biochemical systems that indeed can be expected to display steady state behavior, e.g. simple microorganisms in a fermenter. However, recent experimental data show that many complex biochemical systems display highly nonlinear dynamics. Prominent examples are calcium oscillations in plants and animals responsible for information processing in cells [6], metabolic oscillations in neutrophils [7], and glycolytic oscillations [8, 9]. Moreover, apart from a few cases, most organisms are not subject to a constant environment and therefore will not display steady state behavior, but rather transient behavior of different kinds at all times.

One of the simplest and best characterized representatives for nonlinear behavior in biochemistry is the so-called peroxidase-oxidase reaction (PO reaction). During this reaction NAD(P)H is oxidized by molecular oxygen and the reaction is catalyzed by the enzyme peroxidase [10]. The reaction was shown to display a wide variety of nonlinear dynamics like simple periodic oscillations, complex periodic oscillations, quasiperiodicity and chaos [10]. The system is well studied and quantitative computational models exist [11].

Given this dynamic nature of biochemistry in the living cell, we think that complexity reduction methods should take it into account rather than focusing on steady state behavior. Therefore, complexity reduction has to be performed in a time-dependent way, following the dynamic behavior of the system and enabling researchers to reduce systems also during transient behavior. Methods for such a time-dependent complexity reduction exist in other fields, e.g. chemistry and physics [12, 13]. Here, one of the most common approaches is *time–scale decomposition*. The concept of time–scale decomposition is based on the fact that complex reaction networks typically consist of processes taking place on largely differing characteristic time–scales. Depending on the actual time–scale of interest processes being exhausted on a sufficiently fast time–scale are assumed to be relaxed, whereas processes taking place on a sufficiently slow time–scale can be assumed to be stationary. The time–scale of interest might of course change in the course of the simulation of the system dynamics. This system dynamics is then described with a reduced set of equations representing the time–evolution of the processes being active on the actual time–scale. Thus, the system is described with a reduced set of equations which does not necessarily imply that these equations represent a smaller system of real chemical species. Therefore, this approach is only meant to speed up computation time rather than dissect the biochemical network into modules. The most prominent examples building on the concept of time–scale decomposition are the *Computational Singular Perturbation* (CSP) method [14] and methods based on the computation of so-called *Low-Dimensional Manifolds* (LDM) [15, 16, 17, 18]. Several variants of these two methods have been successfully used e.g. in atmospheric and combustion chemistry modeling (see, e.g., [19, 20, 21]).

Time–scale decomposition should be an excellent approach to dynamic complexity reduction in biochemical systems since biochemical processes proceed on a wide range of time–scales spanning several orders of magnitude. Thus, events like gene expression usually occur in the range of minutes to hours, whereas signal transduction and metabolic reactions take fractions of seconds to seconds to evolve [22]. Looking even more into detail, e.g. at the elementary reaction steps which are involved in a single metabolic reaction and on protein movements etc., one often observes time–scales in the order of fs to ms (see, e.g. [23]). Therefore, separation of time–scales has indeed often been used in a simplified and heuristic manner in the context of modeling biochemical processes. Examples include the simulation of metabolic events without considering gene expression of the associated enzymes or the description of an overall kinetics for a biochemical reaction instead of describing all elementary steps. In the latter case, the famous Michaelis-Menten approximation relies mainly on a separation of time–scales which leads to a quasi-steady state approximation for reactive intermediates like the enzyme-substrate-complex [24]. Apart from often being heuristic, the usage of time–scale decomposition in this context has been mainly limited to assuming the system to operate in a steady state and so far, no automated, time-dependent attempt has been made to study complex metabolic reaction networks, except for very small sample systems (see, e.g., [18]).

In the following, we describe the adaptation of a LDM method for the use in the computational decomposition and analysis of biochemical systems with respect to time–scales. Compared to existing dynamic complexity reduction approaches our new adapted time–scale decomposition method for reaction networks has several advantages. The method can be applied to arbitrary biochemical reaction networks and works independently of restrictive assumptions on the specific dynamical regime of the system like e.g. the steady state approximation. Furthermore, the decomposition and analysis of the system dynamics is performed in a fully automated way, therefore avoiding the error–prone *a priori* identification of reactive intermediates / fast processes that are in most cases valid only for a restricted dynamical regime of the reaction network.

In comparison to the above described LDM methods for chemical systems, our method has an additional focus on the reduction of the underlying biochemical network in a time–dependent manner and not only on the reduction of the mathematical equations. In addition, some numerical differences are introduced as described below.

The suitability of the presented time–scale decomposition method is demonstrated by applying it to the analysis of the PO reaction system. We show that it is not only highly interesting to follow the complexity reduction in time and see how subsystems couple and decouple in the course of an oscillation, but we further show that for nonlinear systems the decoupling of the system strongly depends on the specific dynamics displayed at a specific time of the simulation. Thus, time–scales decouple differently while the system is displaying relaxation oscillations compared to regular oscillations. We show that therefore both aspects of complexity reduction, the improvement of computational speed and the dissection into subsystems have to be discussed in the context of a specific dynamic behavior. These results underline the need for time–dependent complexity reduction methods for the use in systems biology.

2 Methodology

In the following we will describe how mathematical transformations are used to reduce the number of equations necessary to describe the system at any point in time. This formalism is then used to analyze if and how the biochemical network may be dissected into subnetworks in order to achieve a real decomposition of the system.

In the context of a deterministic, homogeneous modeling framework, the state of a biochemical reaction network is represented by the time-dependent state vector $\boldsymbol{c}(t)$ of reacting species concentrations c_i, $(i = 1, \ldots, n)$. The dynamics of the system is determined by a set of n ordinary differential equations (ODEs) together with an initial state $\boldsymbol{c}_0$.

$$\frac{d\boldsymbol{c}(t)}{dt} = \boldsymbol{f}(\boldsymbol{c}(t), \boldsymbol{k}) = \underline{N}\boldsymbol{v}(\boldsymbol{c}(t), \boldsymbol{k}), \qquad \boldsymbol{c}(t = 0) = \boldsymbol{c}_0 \tag{1}$$

The stoichiometric matrix $\underline{N}$ contains the information about the structure of the reaction network. The coefficient N_{ij}, e.g., indicates the participation of species X_i in reaction r_j. The kinetics of the individual reactions is described by the reaction rate vector $\boldsymbol{v}(\boldsymbol{c}(t), \boldsymbol{k})$, where $\boldsymbol{k}$ is a parameter vector containing reaction rate constants etc.

Due to the existence of mass conservation relationships the set of ODEs often exhibits linear dependencies. These linear dependencies can be detected and removed in a systematic, automated way by inspection of the stoichiometric matrix (stoichiometric network analysis) [25, 26]. Each conservation relation can be used to reduce the dimension of the ODE system by one unit. In the following, the ODE system is always assumed to be in its reduced, linearly independent form.

Although the representation of reaction rates $d\boldsymbol{c}(t)/dt$ in terms of contributions from individual reactions in Equation 1 is very intuitive, it is not appropriate for an analysis of the dynamic capabilities of the reaction system from a time–scale point of view. For the latter purpose, a different representation has to be chosen. In order to probe the time–scales inherent in the dynamics of the reaction network, the response of the system to a perturbation at some reference state $\boldsymbol{c}_r$ is analyzed. As a starting point of this analysis a linearization with respect to the state vector $\boldsymbol{c}_r$ is performed and the nonlinear system dynamics is replaced by a linear approximation in a neighborhood of $\boldsymbol{c_r}$ using first-order Taylor expansion.

$$\frac{d\boldsymbol{c}(t)}{dt} = \boldsymbol{f}(\boldsymbol{c}(t), \boldsymbol{k}) \approx \boldsymbol{f}(\boldsymbol{c}_r, \boldsymbol{k}) + \underline{J}_{\boldsymbol{c}_r}(\boldsymbol{c}(t) - \boldsymbol{c}_r), \qquad \underline{J}_{\boldsymbol{c}_r} = \frac{\partial \boldsymbol{f}(\boldsymbol{c}_r)}{\partial \boldsymbol{c}} \tag{2}$$

Here, $\underline{J}_{\boldsymbol{c}_r}$ denotes the Jacobian matrix evaluated at reference state $\boldsymbol{c}_r$.

Using this linear approximation of the local dynamics of the system at state $\boldsymbol{c}_r$ the typically strongly coupled system of ODEs can be partitioned by an appropriate transformation of the representation. In the conceptually simplest case, assuming $\underline{J}$ to be diagonalizable, the eigenvector basis $\underline{E}$ of the real, non-symmetric Jacobian matrix is used to transform the ODE system.

$$\underline{J} \cdot \underline{E} = \underline{E} \cdot \underline{\Lambda}, \qquad \boldsymbol{x} = \underline{E}^{-1} \cdot \boldsymbol{c} \tag{3}$$

Here $\underline{\Lambda}$ represents the diagonal matrix of real or complex eigenvalues λ_i of $\underline{J}$. The components of the transformed state vector $\boldsymbol{x}$ are called modes. Due to the fact that $\underline{\Lambda}$ is a diagonal matrix, the transformed ODE system is fully decoupled.

$$\frac{d\boldsymbol{x}(t)}{dt} = \underline{\Lambda}\boldsymbol{x}(t) \tag{4}$$

Solving the decoupled ODE system yields the time evolution of the individual modes x_i.

$$x_i(t) = x_{i,0}e^{\lambda_i t}, \qquad \tau_i = \frac{1}{|\Re(\lambda_i)|} \tag{5}$$

According to Equation 5 each mode evolves on a *characteristic time–scale* τ_i; i.e. the real parts $\Re(\lambda_i)$ of the eigenvalues of $\underline{J}$ determine the time–scale of the processes taking place locally in the reaction system. The modes x_i may be classified according to their qualitative behavior:

$$\begin{aligned}\Re(\lambda_i) < 0 &\rightarrow \text{relaxing mode (exp. decay)}\\ \Re(\lambda_i) = 0 &\rightarrow \text{constant mode}\\ \Re(\lambda_i) > 0 &\rightarrow \text{exploding mode (exp. increase)}\\ \Im(\lambda_i) \neq 0 &\rightarrow \text{oscillating mode}\end{aligned}$$

The transformed representation of the system dynamics in terms of modes x_i instead of concentration variables c_i provides a systematic and straightforward basis for the time–scale analysis and decomposition of the reaction network (modal analysis)[25]. Neglecting those modes that are considered to be sufficiently relaxed on the time–scale τ of interest results in a reduced *slow*, active state space spanned by the eigenvectors of $\underline{J}$ with $\Re(\lambda_i) \geq 0$ and those with $\Re(\lambda_i) < 0$ and $1/|\Re(\lambda_i)| \geq \tau$.

So far, the discussion is based on a local analysis of the system dynamics at some reference point $\boldsymbol{c}_r$ in phase space only. For linear systems the Jacobian matrix is constant, i.e. it does not depend on the concentration vector $\boldsymbol{c}$. In this specific situation, the above basis transformation and time–scale analysis are valid over the full dynamic range of the system. However, biochemical reaction networks are generally highly nonlinear in nature. Therefore, the structure of the eigenvalue spectrum of $\underline{J}$ may strongly depend on the position of the reference point $\boldsymbol{c}_r$ in phase space. As a consequence, the specific partitioning of the state space into slow/fast subspaces obtained at reference point $\boldsymbol{c}_r$ has to be considered as a local property of the system. In order to obtain a meaningful characterization of the overall dynamics of the nonlinear system the above basis transformation has to be applied repeatedly when propagating on a trajectory in phase space.

One additional problem which has to be considered for (bio)chemical reaction systems of realistic size is that the Jacobian matrix exhibits close-lying or even quasi-degenerate eigenvalues. In this situation the full decomposition of the ODE system using the basis of eigenvectors of $\underline{J}$ is often ill-conditioned [29]. However, in such cases a full decomposition of the system is not necessary and even not reasonable, because the modes associated with eigenvalues of similar size also contribute to the systems dynamics on a similar time–scale. Such modes may be grouped together without losing a significant amount of information.

An alternative approach, that allows for the stable partitioning of the dynamic system into a *slow*, active, and a *fast*, relaxed, subspace has been introduced by Maas and Pope [15]. For our investigation of the time–scale decomposition in biochemical reaction networks we have adapted a variant of the Maas/Pope decomposition scheme as presented by Deuflhard and Heroth [27]. This adapted partitioning scheme will be presented in the following.

In contrast to the basis transformation resulting in a full diagonalization of the Jacobian matrix, this decomposition scheme is based on a similarity transformation that transforms $\underline{J}$ into a matrix with block structure representing the partitioning of the full system into dynamically decoupled subspaces. This task is accomplished by a sequence of matrix transformations.

In the first step, an orthogonal similarity transformation is applied to the Jacobian matrix $\underline{J}$. The resulting matrix $\underline{S}$ has real Schur form, i.e. it is a block upper triangular matrix [28]. In the real Schur matrix $\underline{S}$ the eigenvalues of $\underline{J}$ are obtained as diagonal entries $\underline{S}_{ii}$ $(i = 1, \dots, m)$. These may be either (1×1)-blocks (real eigenvalues) or (2×2)-blocks (complex conjugate pair of eigenvalues).

$$\underline{Q}^T \cdot \underline{J} \cdot \underline{Q} = \underline{S} = \begin{pmatrix} \underline{S}_{11} & \underline{S}_{12} & \cdots & \underline{S}_{1m} \\ \underline{0} & \underline{S}_{22} & \cdots & \underline{S}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \underline{0} & \underline{0} & \cdots & \underline{S}_{mm} \end{pmatrix} = \begin{pmatrix} \underline{S}_{\text{slow}} & \underline{S}_{\text{coup}} \\ \underline{0} & \underline{S}_{\text{fast}} \end{pmatrix} \tag{6}$$

In $\underline{S}$ the eigenvalues have been reordered by a sequence of Givens rotations [28] in such that $|\Re(\lambda_1)| \leq \dots \leq |\Re(\lambda_n)|$. The entries of $\underline{S}$ are grouped together into diagonal blocks $\underline{S}_{\text{slow}}$ and $\underline{S}_{\text{fast}}$ both of which are upper-quasi-triangular submatrices. $\underline{Q}^T, \underline{Q}$ is the orthogonal basis of Schur vectors.

For a given partitioning, $\underline{S}_{\text{slow}}/\underline{S}_{\text{fast}}$ with dimension of the slow subspace $r = n_{\text{slow}}$, the non–zero entries of the coupling matrix $\underline{S}_{\text{coup}}$ can be eliminated in a second step by solving the Sylvester equation [28]

$$\underline{S}_{\text{slow}} \cdot \underline{Z}_r - \underline{Z}_r \cdot \underline{S}_{\text{fast}} = -\underline{S}_{\text{coup}} \tag{7}$$

The solution of the Sylvester equation provides the transformation matrices $\underline{T}_r^{-1}, \underline{T}_r$

$$\underline{T}_r^{-1} = \left(\underline{1} - \begin{pmatrix} \underline{0} & \underline{Z}_r \\ \underline{0} & \underline{0} \end{pmatrix} \right) \cdot \underline{Q}^T, \quad \underline{T}_r = \underline{Q} \cdot \left(\underline{1} + \begin{pmatrix} \underline{0} & \underline{Z}_r \\ \underline{0} & \underline{0} \end{pmatrix} \right) \tag{8}$$

which are then used to perform a non-orthogonal similarity transformation on $\underline{J}$. The resulting matrix $\underline{\tilde{S}}$ has the desired block structure with fully decoupled slow/fast submatrices $\underline{\tilde{S}}_{\text{slow}}$ and $\underline{\tilde{S}}_{\text{fast}}$.

$$\underline{T}_r^{-1} \cdot \underline{J} \cdot \underline{T}_r = \underline{\tilde{S}} = \begin{pmatrix} \underline{\tilde{S}}_{\text{slow}} & \underline{0} \\ \underline{0} & \underline{\tilde{S}}_{\text{fast}} \end{pmatrix} \tag{9}$$

Finally, the application of $\underline{T}_r^{-1}$ on the state vector $\boldsymbol{c}$ and reaction rate vector $\boldsymbol{f}$ results in a decoupled representation of the system dynamics.

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_{\text{slow}} \\ \boldsymbol{x}_{\text{fast}} \end{pmatrix} = \underline{T}_r^{-1} \cdot \boldsymbol{c}, \qquad \boldsymbol{g} = \begin{pmatrix} \boldsymbol{g}_{\text{slow}} \\ \boldsymbol{g}_{\text{fast}} \end{pmatrix} = \underline{T}_r^{-1} \cdot \boldsymbol{f} \tag{10}$$

The partitioning of the reaction system into slow/fast contributions is related to a singular perturbation description of the ODE system in which

$$\begin{aligned} \frac{d\boldsymbol{x}_{\text{slow}}}{dt} &= \boldsymbol{g}_{\text{slow}}(\boldsymbol{x}_{\text{slow}}, \boldsymbol{x}_{\text{fast}}) \\ \epsilon \cdot \frac{d\boldsymbol{x}_{\text{fast}}}{dt} &= \boldsymbol{g}_{\text{fast}}(\boldsymbol{x}_{\text{slow}}, \boldsymbol{x}_{\text{fast}}) \end{aligned} \tag{11}$$

Here, $\epsilon = \tau_{r+1}$ is a singular perturbation parameter.

Assuming $\epsilon dx_{fast}/dt = 0$ ($\epsilon = 0$ corresponds to infinitely fast time scales) it follows $\boldsymbol{g}_{\text{fast}} = \mathbf{0}$ and the fast modes of the reaction system are fully relaxed. The corresponding slow subspace spanned by the first n_{slow} rows of $\underline{T}_r^{-1}$ defines the so–called *intrinsic low-dimensional manifold* (ILDM) at the reference point $\boldsymbol{c}_r$ of the decomposition. The dynamics of the system on the slow time–scale takes place on the ILDM. The relaxed fast components $\boldsymbol{x}_{\text{fast}}$— in general linear combinations of contributions from reactive species— can be interpreted as generalization of the concept of *reactive intermediates* that have to be identified when applying the quasi–steady state approximation (QSSA).

However, in practice the fast modes are not fully relaxed ($0 < \epsilon \ll 1$), i.e. $\boldsymbol{g}_{\text{fast}} \neq \mathbf{0}$ and therefore an error would result from neglecting these dynamics completely. In order to determine a suitable partitioning of the modes into slow/fast components, this error has to be determined and the partitioning has to be chosen such that the error does not exceed a user-specified tolerance.

For this purpose, we adapted an error-criterion established by Deuflhard and Heroth [27]. Here, the error when propagating the separated slow modes starting from a point $\boldsymbol{c} = (\boldsymbol{x}_{\text{slow}}, \boldsymbol{x}_{\text{fast}})^T$ which results from a calculation with $\boldsymbol{g}_{\text{fast}} \neq \mathbf{0}$ compared to the propagation of the same slow modes starting from the corresponding point $\tilde{\boldsymbol{c}} = (\boldsymbol{x}_{\text{slow}}, \boldsymbol{x}_{\text{fast,0}})^T$ with $\boldsymbol{g}_{\text{fast}} = \mathbf{0}$ is determined. This latter point called consistent initial value is situated on the ILDM defined by $\boldsymbol{g}_{\text{fast}} = \mathbf{0}$ with fixed slow modes. It can be computed from $(\boldsymbol{x}_{\text{slow}}, \boldsymbol{x}_{\text{fast}})^T$ by applying a simplified Newton method which relaxes the point to the ILDM. A criterion for the error in the integration of the slow modes generated by the (only approximately correct) assumption that the fast modes are relaxed ($\boldsymbol{g}_{\text{fast}} = \mathbf{0}$) can be obtained according to [27]

$$\tau_{r+1} |\boldsymbol{g}_{\text{slow}}(\boldsymbol{x}_{\text{slow}}, \boldsymbol{x}_{\text{fast}}) - \boldsymbol{g}_{\text{slow}}(\boldsymbol{x}_{\text{slow}}, \boldsymbol{x}_{\text{fast,0}})| \leq tol, \qquad (12)$$

where τ_{r+1} is the time–scale of the fastest slow mode and *tol* a user specified error tolerance.

We evaluate Equation 12 using an iterative procedure in which the number of active slow modes $r = n_{\text{slow}}$, determining the partitioning of the state space, is decreased until the user-defined tolerance *tol* is reached or the Newton iteration fails to converge. In that case the number of active modes has to be increased again by one.

In order to be suitable for the task of analyzing highly nonlinear biochemical reaction systems, the time–scale decomposition scheme has to be coupled to an appropriate integration routine which performs the time propagation between the decomposition steps, i.e. the overall procedure consists of an alternating sequence of decomposition and integration steps. In this sequence the information obtained in the partitioning of the last point on the trajectory, e.g. the number of active modes, can be used as starting point for the partitioning at the actual point. For the purpose of probing the time–scale decomposition along the phase space trajectory of the highly nonlinear PO reaction system, we used the numerical stiff integrator LIMEX with adaptive control of step-size [30] to perform the time propagation of the full ODE system between successive decomposition steps. The discretization of LIMEX is based on a linearly implicit Euler method combined with extrapolation. The reordered Schur decomposition and the evaluation of the Sylvester equation were performed using LAPACK library routines [31].

The above described time–scale decomposition procedure results in a separation of the modes in slow, active and fast, relaxed modes. This partitioning allows the automated reduction of the dynamical system representation to the subspace of slow modes only. Accordingly, the system dynamics of the full reaction system comprising n ODEs is reduced to a differential–algebraic equation (DAE) system consisting of n_{slow} ODEs and $n - n_{\mathrm{slow}}$ algebraic equations. In addition to reducing the number of ODEs necessary to describe the systems dynamics with sufficient accuracy, this process also results in ODEs with a relatively small span of time–scales. This means that the stiffness of the equation system is strongly decreased resulting in larger integration step-sizes and an additional speed–up of the simulation.

As discussed so far, time–scale decomposition results in a reduced representation of the system dynamics reflected in a reduced number of ODEs describing the time–evolution of the active modes. However, since each active mode of the reduced system presentation corresponds to a linear combination of species / concentration variables (see Equation 10), this kind of complexity reduction only offers a potential computational advantage. In order to also gain insight into the composition and interplay of subsystems, we have to evaluate the contribution of each concentration variable to the set of active modes. For this purpose we analyze the transformation matrix T_r^{-1}, the matrix element $T_{r,ij}^{-1}$ determining the contribution of concentration variable c_j to mode x_i. Because we do not carry out a full partitioning of modes in our approach, but only separate the set of active, slow modes from the relaxed fast modes, it makes sense to analyze the sum of contributions of a selected concentration variable c_j to the whole set of active modes. Thus, it is possible to determine which variable(s) are mainly responsible for a certain dynamic behavior at a specific point in time. In the specific case that a subset of species / concentration variables does not contribute to the set of active modes, the time–scale decomposition of the system dynamics results in a dissection of the reaction network itself. Of course, this network decomposition may sensitively depend on the dynamic regime of the reaction system, i.e. it may change in the course of the simulation.

3 Application and Results

In order to probe our adapted ILDM method, we computed and analyzed the time–scale decomposition in the PO reaction system displaying different kinds of dynamic behavior. A simple time–scale decomposition of very simplified models of the PO reaction has been studied before, however not in a time–dependent way [33]. The reaction mechanism of the PO reaction, the oxidation of NAD(P)H catalyzed by peroxidase consists of a number of elementary reaction steps [10]. These steps are well characterized and kinetic parameters are known for most of these steps and for several different peroxidases [10, 11, 34]. Therefore, it is possi-

Table 1. Detailed model of the peroxidase–oxidase reaction coupled to the activation of an enzyme Enz (^{a} Per^{3+} and Per^{2+} indicate iron(III) and iron(II) peroxidase respectively. coI, coII and coIII indicate the enzyme intermediates compound I, compound II and compound III. ^{b} In M^{-1} s^{-1}. ^{c} In M. ^{d} In s^{-1}. ^{e} The value of $[O_2]_{eq}$ is 1.2×10^{-5} M. ^{f} The amount of Enz_{inact} is assumed to be large compared to Enz_{act} and therefore considered constant)

reaction^a	rate expression	constant
(1) $NADH + O_2 + H^+ \longrightarrow NAD^+ + H_2O_2$	$k_1[NADH][O_2]$	3.0 ^{b}
(2) $H_2O_2 + Per^{3+} \longrightarrow coI$	$k_2[H_2O_2][Per^{3+}]$	1.8×10^7 ^{b}
(3) $coI + NADH \longrightarrow coII + NAD^{\cdot}$	$k_3[coI][NADH]$	4.0×10^5 ^{b}
(4) $coII + NADH \longrightarrow Per^{3+} + NAD^{\cdot}$	$k_4[coII][NADH]$	2.6×10^5 ^{b}
(5) $NAD^{\cdot} + O_2 \longrightarrow NAD^+ + O_2^-$	$k_5[NAD^{\cdot}][O_2]$	2.0×10^7 ^{b}
(6) $O_2^- + Per^{3+} \longrightarrow coIII$	$k_6[O_2^-][Per^{3+}]$	1.7×10^6 ^{b}
(7) $2O_2^- + 2H^+ \longrightarrow H_2O_2 + O_2$	$k_7[O_2^-]^2$	2.0×10^7 ^{b}
(8) $coIII + NAD^{\cdot} \longrightarrow coI + NAD^+$	$k_8[coIII][NAD^{\cdot}]$	11.0×10^7 ^{b}
(9) $2NAD^{\cdot} \longrightarrow NAD_2$	$k_9[NAD^{\cdot}]^2$	5.6×10^7 ^{b}
(10) $Per^{3+} + NAD^{\cdot} \longrightarrow Per^{2+} + NAD^+$	$k_{10}[Per^{3+}][NAD^{\cdot}]$	1.8×10^6 ^{b}
(11) $Per^{2+} + O_2 \longrightarrow coIII$	$k_{11}[Per^{2+}][O_2]$	1.0×10^5 ^{b}
(12) $\longrightarrow NADH$	k_{12}	*variable*
(13) $O_2(gas) \longrightarrow O_2(liquid)$	$k_{13}[O_2]_{eq}$	$4.4 \times 10^{-3 d,e}$
(−13) $O_2(liquid) \longrightarrow O_2(gas)$	$k_{-13}[O_2]$	4.4×10^{-3} ^{d}
(14) $Enz_{inact} + O_2^- \longrightarrow Enz_{act}$	$\frac{k_{14}[O_2^-]^5}{(K_f^5+[O_2^-]^5)}$	0.005 ^{b} (k_{14})
		0.4 $^{cf}(K_f)$
(15) $Enz_{act} \longrightarrow Enz_{inact}$	$k_{15}[Enz_{act}]$	1.6 ^{d}

ble to model the reaction using a detailed description of the reaction mechanism as displayed in Table 1 [34]. The rate expressions for each reaction are listed as well as the kinetic parameters used in this study. For the full and realistic system no complexity reduction method has been applied so far.

In addition to analyzing the PO reaction, we coupled the activation of an enzyme to the concentration of superoxide radicals in order to gain a simple case study for the dynamic coupling of different subsystems. Superoxide radicals are known to play the role of second messengers in the cell [35, 36]. The binding characteristics of these radicals to their target molecules is not yet know. Since many messenger molecules (e.g. calcium [37]) bind cooperatively to their respective targets, we assumed very general kinetics taking a potential cooperativity and a simple linear deactivation step into account (see Table 1).

Using these reaction steps to model the whole system leads to a set of ordinary differential equations as described in [34] plus the equation for the coupled enzyme activity. A stoichiometric network analysis of the reaction system detects one conservation relation, namely the total number of peroxidase distributed over the five different active species coI, coII, coIII, Per^{2+} and Per^{3+}. This finding, together with the fact that the species NAD(P)$_2$ and NAD(P)$^+$ are only produced and not consumed in the PO reaction system, reduces the dimension of the state space from 13 to 10.

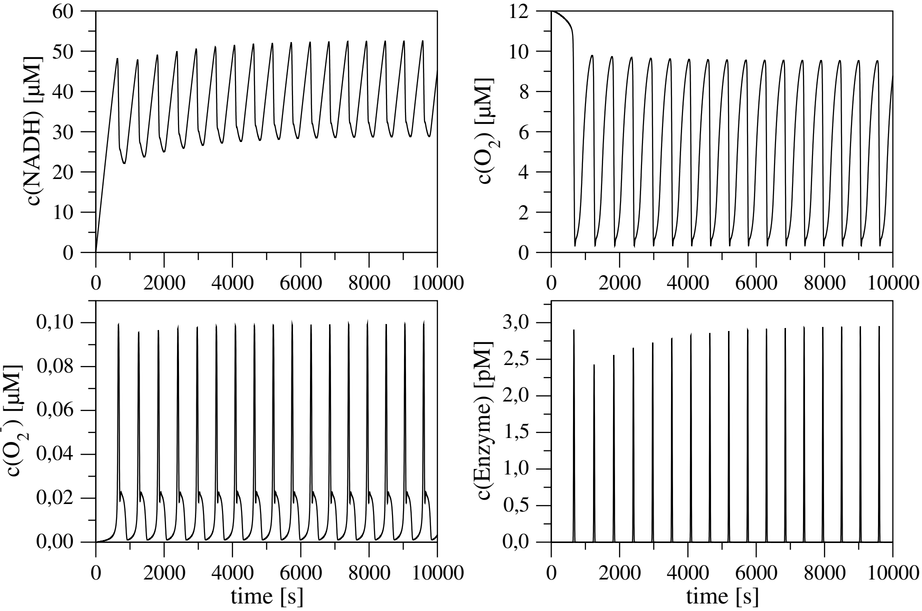


Fig. 1. Simulated time series of selected species concentrations of the PO reaction system model as specified in Table 1. For the chosen NAD(P)H inflow rate of $k_{12} = 0.082[\mu M/s]$ the systems dynamics is characterized by sustained large amplitude relaxation oscillations. The initial concentrations of O$_2$ and Per^{3+} were 12.0 and 1.5 μM, respectively; all other initial concentrations were zero

Varying the rate of NAD(P)H inflow (k_{12}) into the open system leads to different kinds of nonlinear behavior which mimics the experimental observations very well [10]. We explicitly studied the behavior at $k_{12} = 0.082$, 0.129 and $0.132[\mu M/s]$ corresponding to a dynamics characterized by high-amplitude relaxation oscillations, small amplitude regular oscillations and steady state behavior (see Figure 1-3). In the case of $k_{12} = 0.129[\mu M/s]$ (Figure 2) the system displays transient relaxation oscillations first before it settles into regular sustained oscillations. We chose this specific situation as main focus of our analysis because it offers the possibility to observe qualitatively differing behavior in complexity reduction on a single run.

Studying the activity of the enzyme *Enz* coupled to the PO reaction system, we observe that it is only activated in a pulse-like fashion when the system displays large amplitude relaxation oscillations whereas it remains constant (within error-tolerance) during regular small amplitude oscillations with the chosen parameters (see Figure 2). Once again, we want to underline that the equation for the enzyme activation does not correspond to one particular enzyme, but rather represents some general properties occurring frequently in biochemical systems and serves as a very simple, but characteristic prototype case in our investigation.

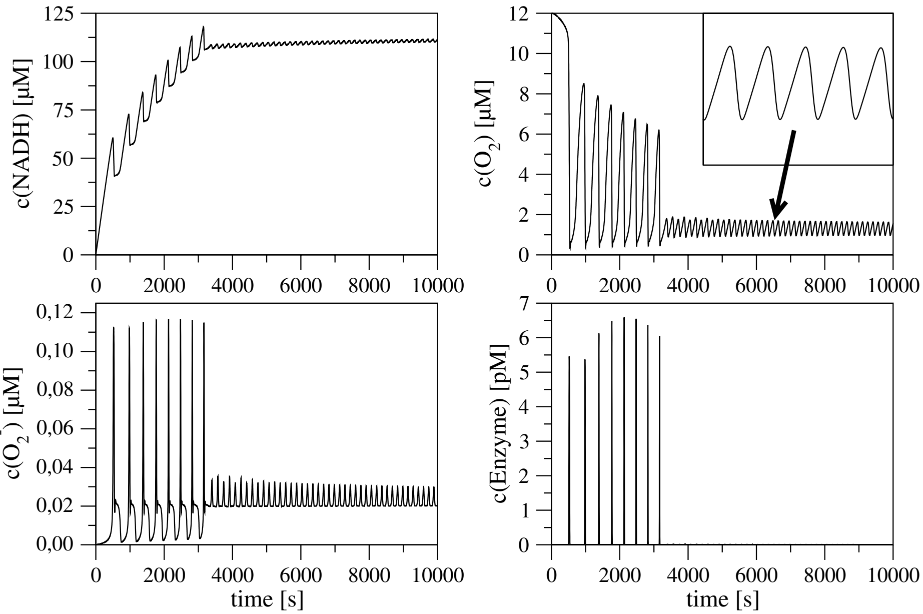


Fig. 2. Simulated time series of selected species concentrations of the PO reaction system model as specified in Table 1. For the chosen NAD(P)H inflow rate of $k_{12} = 0.129[\mu M/s]$ the systems undergoes a transient phase characterized by large amplitude relaxation oscillations, followed by a dynamics regime showing sustained small amplitude regular oscillations (starting at $\approx 3000s$). Interestingly, the active form of the enzyme *Enzyme*, driven by the periodic activation from Superoxide (O_2^-) in the transient large amplitude phase is dynamically switched off in the regular oscillation regime. The initial concentrations of O_2 and Per^{3+} were 12.0 and 1.5 μM, respectively; all other initial concentrations were zero

The results of the complexity reduction analysis of this reaction system with NAD(P)H inflow $k_{12} = 0.129[\mu M/s]$ are shown in Figure 4. The dynamic system represented by a set of 10 ODEs has been analyzed with our adapted ILDM method while propagating along the phase space trajectory obtained by integrating the full ODE system (see Figure 2) starting from the initial state $t = 0s$ up to $t = 4500s$. The user-specified error tolerance has been set to $tol = 1.0e^{-4}$.

As can be seen in the lower panel, the original ODE system of dimension 10 is representable to a very good approximation by a transformed, reduced system consisting of maximally five to six active modes while displaying large amplitude relaxation oscillations. Within each oscillation period, the maximum number of active modes is required in the short relaxation phase, i.e. in the time interval, where the concentrations of the reacting species change most rapidly. In the intervals between the rapid relaxation phases the active state space could be further reduced to three to four active modes.

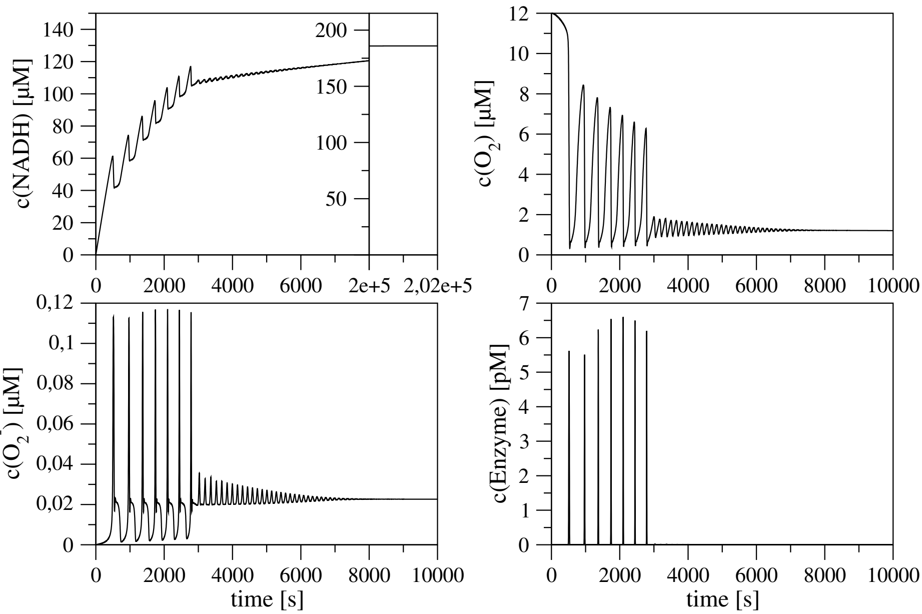


Fig. 3. Simulated time series of selected species concentrations of the PO reaction system model as specified in Table 1. For the chosen NAD(P)H inflow rate of $k_{12} = 0.132[\mu M/s]$ the systems shows a transient phase characterized by large amplitude relaxation oscillations, followed by a transient regime showing damped small amplitude regular oscillations before approaching a steady state. Like in the case of sustained small amplitude regular oscillations, the activation of the enzyme *Enzyme* is dynamically switched off. The initial concentrations of O_2 and Per^{3+} were 12.0 and 1.5 μM, respectively; all other initial concentrations were zero

In contrast to the situation observed in the relaxation oscillation regime, the system can be represented by an even lower average number of active modes while propagating in the regime of sustained regular oscillations starting at $t \approx 3200s$. The actual number of modes is oscillating between five and three. Again, the maximum number of active modes is required in the time interval where the concentrations of the reaction species change most rapidly.

In order to further rationalize the observed pattern of active modes we also inspected the time-dependent eigenvalue spectrum of the Jacobian matrix along the phase space trajectory. The real part of the eigenvalues $\Re(\lambda_i)$ is displayed in the two center panels of Figure 4. Very much like the pattern of active modes, the overall structure of the eigenvalue spectrum nicely reflects the observed oscillatory structure of the time series of the reacting species. By far the most eye-catching feature of the eigenvalue spectrum is the qualitatively differing structure when changing from the transient relaxation oscillation phase to the regular oscillation regime. The eigenvalues $\Re(\lambda_i)$ of the Jacobian matrix are directly related to the characteristic time–scales $\tau_i = 1/|\Re(\lambda_i)|$ of the modes.

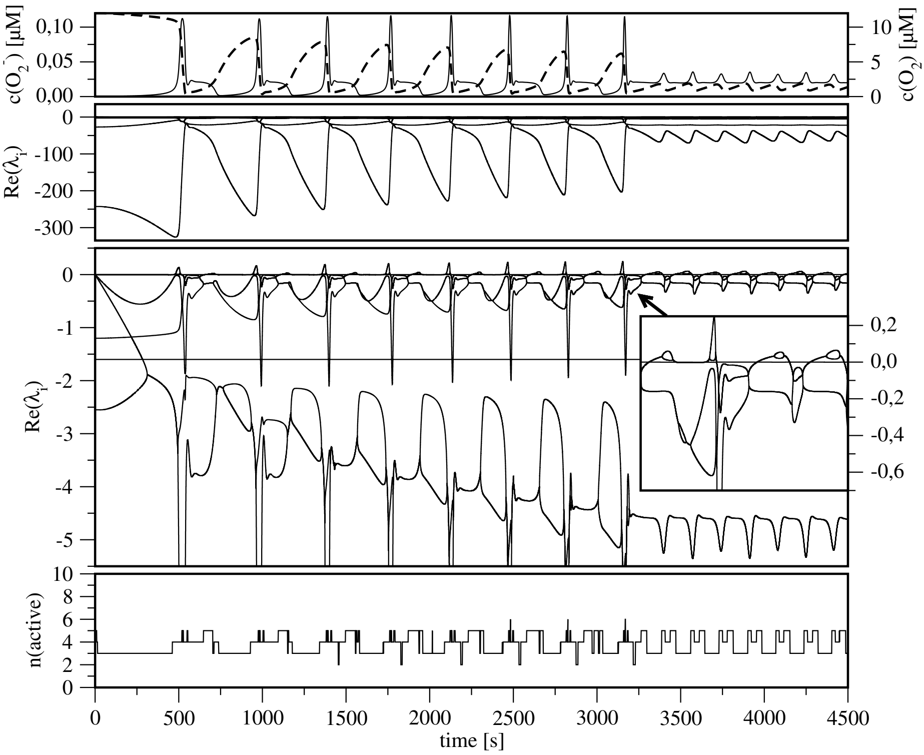


Fig. 4. Time–scale decomposition of the PO reaction system with an NAD(P)H inflow rate of $k_{12} = 0.129[\mu M/s]$. The analysis is performed along the phase space trajectory indicated by the time series of species O_2 (- - -) and O_2^- (—) in the upper panel (see also Figure 2). The two center panels show the structure of the time–dependent eigenvalue spectrum of the Jacobian matrix along the trajectory. Plotted are the real parts $\Re(\lambda_i)$ of the eigenvalues. The lower panel shows the number of active, 'slow' modes resulting from the time–scale decomposition of the phase space along the trajectory. Both the eigenvalue structure and the time dependent pattern in the number of active modes clearly reflect the qualitative change in the systems dynamics in the transition from transient relaxation oscillations to sustained regular oscillations

Therefore, the structural difference in the eigenvalue spectrum is the main cause of the change in behavior observed in the number of active modes. In the transient phase of relaxation oscillations the structure of the eigenvalue spectrum exhibits a rather complex pattern. Repeatedly, relatively strong positive eigenvalues occur whenever the relaxation phase of the oscillations approaches. This is the main destabilizing factor in the system which indicates the phase of the oscillations where dramatical changes occur. The occurrence of distinct positive eigenvalues may be interpreted in terms of a mode causing a periodic strong perturbation that drives the whole system into an oscillatory pattern. Each perturbation, appearing on a relatively short time–scale, is followed by a relaxation

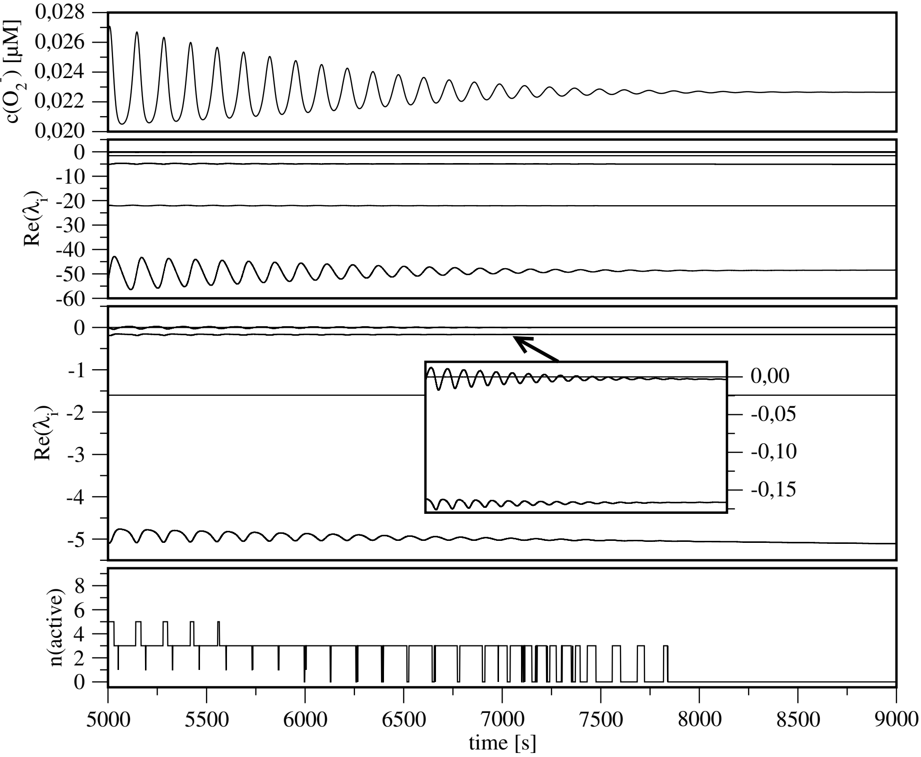


Fig. 5. Time–scale decomposition of the PO reaction system with an NAD(P)H inflow rate of $k_{12} = 0.132[\mu M/s]$. The analysis is performed in the interval of the trajectory where the transient damped regular oscillatory behavior approaches the steady state. This is indicated by the time series of species O_2^- in the upper panel (see also Figure 3). The two center panels show the structure of the time–dependent eigenvalue spectrum of the Jacobian matrix along the trajectory. Plotted are the real parts $\Re(\lambda_i)$ of the eigenvalues. The lower panel shows the number of active, 'slow' modes resulting from the time–scale decomposition of the phase space along the trajectory

phase in which all eigenvalues are negative. The latter relaxation takes place on a comparatively long time–scale.

The situation in the regular oscillation regime is much simpler. The eigenvalues in the spectrum of the Jacobian matrix, and therefore also the characteristic time–scales observable in the systems dynamics are clustered together in groups, the individual groups being separated by large gaps. In sharp contrast to the relaxation oscillation regime, in which the appearance of positive eigenvalues (*exploding* modes) causes distinct mode-mixing effects, the clustered eigenvalue structure is only slightly influenced by the periodic appearance of relatively small positive eigenvalues. Therefore, the number of active modes changes to a much fewer extent as compared to the situation in the relaxation oscillation regime.

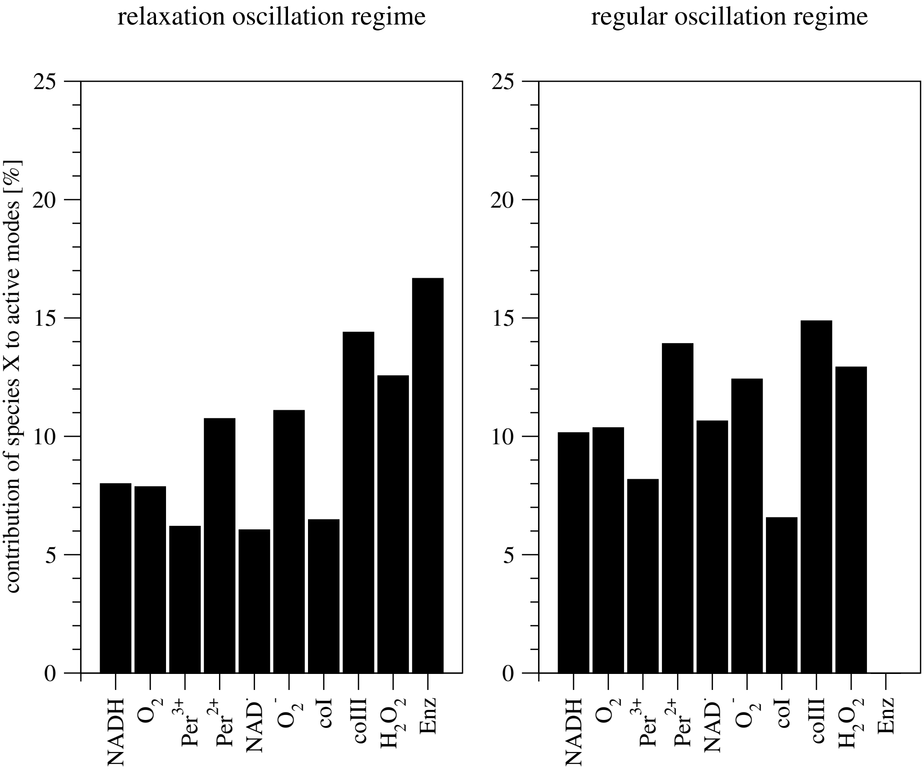


Fig. 6. Analysis of the active modes with respect to the contributions of the species present in the PO reaction system. The results of the modal analysis are shown at two characteristic points on the phase space trajectory for a NAD(P)H inflow rate of $k_{12} = 0.129[\mu M/s]$. On the left panel the species contributions to the active modes are shown at an arbitrary peak position in the relaxation oscillation regime; the right panel shows the result of the same analysis performed at a peak position of the regular oscillation regime. An eye-catching difference in the distribution is observed for the *Enzyme* contribution which is the largest contribution at the chosen peak position in the relaxation regime, but is approximately zero in the regular oscillation regime. This observation indicates the dynamic decoupling of the *Enzyme* subsystem in the regular oscillation regime. In both cases the contributions from species being part of the PO reaction system are of comparable size, indicating strong dynamic coupling of the reaction processes in that system

In order to complete our investigation of the PO reaction system, we have applied the time–scale decomposition scheme to a dynamic regime of damped regular oscillations approaching the steady state. This dynamic regime is observed when slightly increasing the NAD(P)H inflow from $k_{12} = 0.129$ to $k_{12} = 0.132[\mu M/s]$ (see Figure 3). The results of this time–scale decomposition analysis are shown in Figure 5. Obviously, the eigenvalue structure of this regime is much simpler compared to the relaxation oscillations. The number of active modes starts with a pattern oscillating between one and five modes which gradually decrease in number until the steady state with zero active modes is sufficiently approached. To that end, the periodic appearance of positive eigenvalues has vanished and the Jacobian matrix is approximately constant.

So far we have restricted the discussion on the analysis of the reduced number of active modes resulting from the time–scale decomposition. This analysis offers a potential computational advantage, namely the decrease in computation time when integrating the system with reduced active state space. However, the other important goal of complexity reduction, namely the dissection of the biochemical system into different subsystems which can be treated separately cannot be achieved by a systematic reduction of the number of active modes only. Since every concentration variable can, in principle, contribute more or less to every active mode, the time–scale decomposition scheme has to be extended by an analysis that allows the automated study of the contributions of each variable to the active modes. As already discussed, this information is obtained in a straightforward way by analyzing the entries of the transformation matrix $\underline{T}_r^{-1}$ (see Equation 10). According to this equation, the elements of T_r^{-1} correspond to coefficients in an active mode vector representation as a linear combination of real species variables (basis vectors). Therefore, they can be interpreted as weighting factors measuring the relative contribution of each species to the active modes. Figure 6 shows the results of the analysis of the active modes in the system with a NAD(P)H inflow rate of $k_{12} = 0.129[\mu M/s]$ in terms of contributions from all the species concentrations. We compare the results of this analysis being performed at two characteristic, selected states of the phase space trajectory, namely at a peak position in the relaxation oscillation regime as well as in the regular oscillation regime. Looking at this contributions, it is easy to see that in both cases, the contributions from all species being part of the original PO system are of comparable size. Therefore, no separation of subsystems in terms of time–scales is possible within the PO system itself; the system has to be considered to be strongly coupled on all time–scales of observation. However, the analysis allows to observe that compound III, an enzyme intermediate together with H_2O_2 contributes most to relaxation oscillation at this crucial point in time. This is in accordance to experimental investigations which suggested these two species to be the most important ones for the oscillations of the PO reaction [38]. Interestingly, during regular oscillation ferrous peroxidase seems also to play a more dominant role.

Inspection of the full studied system, i.e. taking the enzyme coupled to the PO reaction system into account, results in an obvious difference between the two dynamic regimes. The variable representing the enzyme activity contributes significantly to the active modes at the chosen peak position in the relaxation oscillation regime, but has approximately zero contribution to the active modes in the regular oscillation regime. This observation obviously indicates the strong dynamic coupling of the enzyme activity to the PO reaction system in the case of relaxation oscillations whereas it fully decouples from the PO system in the case of regular oscillations. This holds for the full regular oscillation regime and makes a dissection of the biochemical network possible. Thus, this dissection into subsystems depends strongly on the displayed dynamics. In this case study, the PO reaction system therefore acts as a dynamic switch for the reactions catalyzed by the enzyme activated by O_2^-.

4 Discussion

We have introduced an adapted ILDM method for the use of time–dependent complexity reduction in the context of systems biology. We applied the method to the PO reaction coupled to a simple enzyme activity while the system displayed different kinds of behavior (relaxation oscillations, regular oscillations, steady state). We studied both the mathematical complexity reduction in terms of reduced numbers of equations and the possibility to dissect the biochemical network in dependence on the systems dynamics. We observed that the number of active modes depends crucially on this dynamics, small amplitude regular oscillations having a lower complexity (lower number of active modes) than relaxation oscillations. In addition, the minimum number of active modes necessary to represent the systems dynamics with a given accuracy also changes during each phase of the oscillations. This result emphasizes the need to pursue complexity reduction in a time–dependent manner.

Since the computation of the number of active modes is done repeatedly, i.e. at each integration step, it is computationally expensive. Therefore, a real computational advantage in terms of decreased simulation time is only to be expected when the algorithm is further improved and applied to high-dimensional systems which allow a substantial dimension reduction using this method. Moreover, the computational advantage will multiply, once spatio-temporal systems are simulated compared to homogeneous, temporal ones. In this case, time–scale decomposition determined with this method and the subsequent simplification of the system is applicable to each spatial element that is calculated. The computation of the whole systems dynamics, involving 'spatial' (transport) processes as well as 'local' chemical reactions can be drastically simplified by reducing the representation of the chemical processes in a first step using time–scale decomposition methods, and then using the reduced system as starting point for the simulation of the full system.

An important additional aspect of our methodology is an increased understanding of the dynamic interaction of subsystems. This is achieved by analyzing the contribution of each biochemical species (variable) to the active modes in the system. We used a small and rather simple case study corresponding to the PO reaction this time coupled to a single enzyme activity. Of course, this activity would influence other variables in a real system, the particular enzyme activity modeled here only representing the interface between the two systems. We observed that variables contribute differently to different dynamic behavior verifying the importance of the enzyme intermediate compound III during oscillations. However, all species within the PO reaction are too strongly coupled to be able to fully dissect this system. Nevertheless, the subsystem consisting of the coupled enzyme can be detached depending on the type of oscillatory behavior. We conclude that the possibility to dissect a system depends crucially on the respective dynamics. Of course, this result could have been achieved by simulating the time series and a simple visual inspection showing that in one case, oscillations in the enzyme activity are observed whereas in the other case, the enzyme activity remains constant. However, this case study is an extremely simple case and chosen to point out the major aspects presented in this article. Examining much larger systems in the future, we will need to rely on automated methods providing this kind of analysis.

Our results show that both aims pursued by complexity reduction algorithms, namely computational advantages as well as dissection of systems into subsystem can only be achieved by taking into account the dynamic nature of biochemical processes. Obviously, an analysis of the steady state of a system will not be sufficient to allow a global decoupling of subsystems which holds for all states of the system.

Acknowledgments

J. Zobeley and U. Kummer would like to thank the BMBF and the Klaus Tschira Foundation for funding.

References

1. Kitano, H., 'Computational systems biology', Nature 420, 2002, 206–210 and references therein.
2. Kauffman, K.J., Pajerowski, J.D., Jamshidi, N., Palsson, B.O., and Edwards, J.E., 'Description and analysis of metabolic connectivity and dynamics in the human red blood cell', Biophys. J. 83, 2002, 646–662.
3. Price, N.D., Reed, J.L., Papin, J.A., Famili, I., and Palsson, B.O., 'Analysis of metabolic capabilities using singular value decomposition of extreme pathway matrices', Biophys. J. 84, 2003, 794–804.
4. Schuster, S., Pfeiffer, T., Moldenhauer, F., Koch, I., and Dandekar, T., 'Exploring the pathway structure of metabolism: decomposition into subnetworks and application to Mycoplasma pneumoniae', Bioinformatics 18, 2002, 351–361.
5. Holme, P., Huss, M., and Jeong, H., 'Subnetwork hierarchies of biochemical pathways' Bioinformatics 19, 2003, 532–538.
6. Berridge, M.J., Bootman, M.D., and Lipp, P., 'Calcium - a life and death signal' , Nature 395, 1998, 645–648.
7. Petty, H.R., Worth, R.G., and Kindzelskii, A.L., 'Imaging sustained dissipative patterns in the metabolism of individual cells' , Phys. Rev. Lett. 84, 2000, 2754–2757.
8. Duysens, L.N.M., and Amesz, J. 'Fluorescence sprectrophotometry of reduced phosphopyridine nucleotide in intact cells in the near-ultraviolet and visible region', Biochim. Biophys. Acta 24, 1957, 19–26.
9. Frenkel, R., 'Control of reduced diphosphopyridine nucleotide oscillations in beef heart extracts.I. Effect of modifiers of phosphofructokinase activity', Arch. Biochem. Biophys. 125, 1968, 151–156.

10. Scheeline, A., Olson, D.L., Williksen, E.P., Horras, G.A. Klein, M.L., and Larter, R., 'The peroxidase-oxidase oscillator and its constituent chemistries', Chem. Rev. 97, 1997, 739–756.
11. Bronnikova, T.V. , Fed'kina, V.R., Schaffer, W.M., and Olsen, L.F., 'Period-doubling bifurcations and chaos in a detailed model of the peroxidase-oxidase reaction', J. Phys. Chem. 99, 1995, 9309–9312.
12. Okino, M.S., and Mavrovouniotis, M.L., 'Simplification of mathematical models of chemical reaction systems', Chem. Rev. 98, 1998, 391–408.
13. Tomlin, A.S., Turanyi, T., and Pilling, M.J., 'Mathematical tools for the construction, investigation and reduction of combustion mechanisms', in *Low Temperature Combustion and Autoignition*, Ed. M.J. Pilling, Elsevier, Amsterdam, 1997, 293–437
14. Lam, S.H., and Goussis, D.A., 'The CSP method for simplifying kinetics', Int. J. Chem. Kinet. 26, 1994, 461–486 and references therein.
15. Maas, U., and Pope, S.B., 'Simplifying chemical reaction kinetics: Intrinsic low-dimensional manifolds in composition space', Combustion and Flame 88, 1992, 239–264.
16. Davis, M.J., and Skodje, R.T., 'Geometric investigation of low–dimensional manifolds in systems approaching equilibrium', J. Chem. Phys. 111, 1999, 859–847.
17. Skodje, R.T., and Davis, M.J., 'Geometrical simplification of complex kinetic systems', J. Phys. Chem. A, 105, 2001, 10356–10365.
18. Roussel, M.R., and Fraser, S.J., 'Invariant manifold methods for metabolic model reduction', Chaos, 11, 2001, 196–206.
19. Valorani, M., and Goussis, D.A., 'Explicit time–scale splitting algorithm for stiff problems: auto-ignition of gaseous mixtures behind a steady shock', J. Comput. Phys. 169, 2001, 44–79.
20. Schmidt, D., Blasenbrey, T., Maas, U., 'Intrinsic low-dimensional manifolds of strained and unstrained flames', Combustion Theory and Modelling 2, 1998, 135–152.
21. Correa, C., Niemann, H., Schramm, B., and Warnatz, J., 'Reaction mechanisms reduction for higher hydrocarbons by the ILDM method', Proc. Comb. Inst. 28, 2001, 1607–1614.
22. Voet, D., and Voet, J.G.,*Biochemistry*, Wiley, New York, 1990.
23. Agarwal, P.K., Billeter, S.R., Ravi Rajagopalan, P.T., Benkovic, S.J., and Hammes-Schiffer, S., 'Network of coupled promoting motions in enzyme catalysis', Proc. Natl. Acad. Sci. 99, 2002, 2794–2799.
24. Segel, L.A., and Slemrod, M.,'The Quasi-steady state assumption: a case study in perturbation', SIAM Review 31, 1989, 446–477.
25. Heinrich, R., and Schuster, S., 'The regulation of cellular systems', Chapman and Hall, New York, 1996.
26. Reder, C., 'Metabolic control theory: a structural approach', J. Theor. Biol. 135, 1988, 175–201.
27. Deuflhard, P., and Heroth, J., 'Dynamic dimension reduction in ODE models' in *Scientific Computing in Chemical Engineering*, Springer, Berlin 1996, 29–43.
28. Golub, G.H., and van Loan, C.F., '*Matrix computations*',3d edition, Johns Hopkins University Press, Baltimore, 1996.
29. Golub, G.H., and Wilkinson, J.H., 'Ill-conditioned eigensystems and computation of the Jordan canonical form', SIAM review 18, 1976, 578–619.
30. Deuflhard, P., and Nowak, U., 'Extrapolation Integrators for Quasilinear Implicit ODEs' in *Large Scale Scientific Computing. Progress in Scientific Computing 7*, Birkh"auser, Boston, 1987, 37–50.

31. Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., and Sorensen, D., 'LAPACK Users' Guide', 3d edition, SIAM, Philadelphia, 1999.
32. Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T., '*Numerical recipes in FORTRAN 77*, 2nd edition, Cambridge University Press 1993.
33. Thompson, D.R., and Larter, R., 'Multiple time–scale analysis of two models for the peroxidase-oxidase reaction', Chaos 5, 1995, 448–457.
34. Hauser, M.J.B., Kummer, U., Larsen, A.Z., and Olsen, L.F., 'Oscillatory dynamics protect enzymes and possibly cells against toxic intermediates', Faraday Discuss. 120, 2001, 215–227.
35. Amit, A., Kindzelskii, A.L., Zanoni, J., Jarvis, J.N., and Petty, H.R., 'Complement deposition on immune complexes reduces the frequencies of metabolic, proteolytic, and superoxide oscillations of migrating neutrophils', Cell. Immunol. 194, 1999, 47–53.
36. Klann, E., Robertson, E.D., Knapp, L.T., and Sweat, J.D., 'A role for superoxide in protein kinase C activation and long-term potentiation', J. Biol. Chem. 273, 1998, 4516–4522.
37. Carafoli, E., Santella, L., Brance, D. and Brini, M. 'Generation, control, and processing of cellular calcium signals', Crit. Rev. Biochem. Mol. Biol. 36, 2001, 107–260.
38. Olson, D.L., Williksen, E.P., and Scheeline, A. 'An experimentally based model of the Peroxidase–NADH biochemical oscillator: an enzyme–mediated chemical switch', J. Am. Chem. Soc. 117, 1995, 2–15. Biol. 36, 2001, 107-260.

Author Index