Katsuhisa Horimoto   Georg Regensburger
Markus Rosenkranz   Hiroshi Yoshida (Eds.)

Algebraic Biology

Invited Talks and Short Communications

Third International Conference, AB 2008
Castle of Hagenberg, Austria, July/August 2008

Preface

This booklet contains the invited talk by Kiyoshi Asai and the extended abstracts of the short communications session of the Third International Conference on Algebraic Biology (AB 2008). Jointly organized by the National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, and the Research Institute for Symbolic Computation (RISC), Hagenberg, Austria, it was held from July 31 to August 2, 2008 in the Castle of Hagenberg.

Algebraic biology is an interdisciplinary forum for research on all aspects of applying symbolic computation in biology. The first conference on algebraic biology (AB 2005) was held during November 28–30, 2005 in Tokyo, the second during July 2–4, 2007 in Hagenberg. The AB conference series is intended as a bridge between life sciences and symbolic computation: On the one hand, new insights in biology are found by powerful symbolic methods; on the other hand, biological problems suggest new algebraic structures and algorithms. While this profile has been established in the previous proceedings, the papers in the present volume demonstrate the continuous growth of algebraic biology.

The short communications session was introduced this year for encouraging the presentation of interesting but "not-yet-polished" ideas, in particular unconventional proposals carrying the potential of creating new links between biology and symbolic computation. Despite the late announcement, six communications were submitted from five countries (Austria, France, Japan, Ukraine, USA), and five communications were accepted through a peer-viewing procedure by Program Committee members.

We hope the reader will enjoy the stimulating presentation of Kiyoshi Asai as well as the diversity of innovative ideas communicated in the extended abstracts.

July 2008

Bruno Buchberger
Katsuhisa Horimoto
Reinhard Laubenbacher
Bud Mishra
Georg Regensburger
Markus Rosenkranz
Hiroshi Yoshida

Conference Organization

Conference Chairs

Bruno Buchberger	Johannes Kepler University of Linz, Austria
Katsuhisa Horimoto	National Institute of Advanced Industrial Science and Technology, Japan
Reinhard Laubenbacher	Virginia Bioinformatics Institute, USA
Bud Mishra	New York University, USA

Program Chairs

Katsuhisa Horimoto	National Institute of Advanced Industrial Science and Technology, Japan
Georg Regensburger	Johann Radon Institute for Computational and Applied Mathematics, Austria
Markus Rosenkranz	Johann Radon Institute for Computational and Applied Mathematics, Austria
Hiroshi Yoshida	Kyushu University, Japan

Program Committee

Sachiyo Aburatani	National Institute of Advanced Industrial Science and Technology, Japan
Tatsuya Akutsu	Kyoto University, Japan
Hirokazu Anai	Fujitsu Laboratories Ltd., Japan
Niko Beerenwinkel	ETH Zurich, Switzerland
Armin Biere	Johannes Kepler University of Linz, Austria
Bruno Buchberger	Johannes Kepler University of Linz, Austria
Luca Cardelli	Microsoft Research, Cambridge, UK
Gautam Dasgupta	Columbia University, USA
François Fages	INRIA Rocquencourt, France
Hoon Hong	North Carolina State University, USA
Katsuhisa Horimoto	National Institute of Advanced Industrial Science and Technology, Japan
Abdul Jarrah	Virginia Bioinformatics Institute, USA
Erich Kaltofen	North Carolina State University, USA
Veikko Keränen	Rovaniemi University of Applied Sciences, Finland
Hans A. Kestler	University of Ulm, Germany
Reinhard Laubenbacher	Virginia Bioinformatics Institute, USA
Pierre Lescanne	Ecole Normale Supérieure of Lyon, France
James F. Lynch	Clarkson University, USA
Manfred Minimair	Seton Hall University, USA
Bud Mishra	New York University, USA
Eugenio Omodeo	University of Trieste, Italy

Georg Regensburger	Johann Radon Institute for Computational and Applied Mathematics, Austria
Markus Rosenkranz	Johann Radon Institute for Computational and Applied Mathematics, Austria
Stanly Steinberg	University of New Mexico, USA
Seth Sullivant	Harvard University, USA
Carolyn L. Talcott	SRI International, USA
Francis Thackeray	Transvaal Museum, Northern Flagship Institution, South Africa
Ashish Tiwari	SRI International, USA
Hiroyuki Toh	Kyushu University, Japan
Dongming Wang	Beihang University, China and UPMC-CNRS, France
Bridget S. Wilson	University of New Mexico, USA
Limsoon Wong	National University of Singapore
Kazuhiro Yokoyama	Rikkyo University, Japan
Hiroshi Yoshida	Kyushu University, Japan
Ruriko Yoshida	University of Kentucky, USA

Invited Speakers

Kiyoshi Asai	National Institute of Advanced Industrial Science and Technology, Japan
Charles Cantor	Sequenom, Inc., USA

Tutorial Speakers

Tatsuya Akutsu	Kyoto University, Japan
Armin Biere	Johannes Kepler University of Linz, Austria
François Boulier	University Lille I, France
Ken Fukuda	National Institute of Advanced Industrial Science and Technology, Japan
Tohru Natsume	Biomedicinal Information Research Center, Japan
Ashish Tiwari	SRI International, USA

Local Organization

Betina Curtis	Johannes Kepler University of Linz, Austria
Georg Regensburger	Johann Radon Institute for Computational and Applied Mathematics, Austria
Markus Rosenkranz	Johann Radon Institute for Computational and Applied Mathematics, Austria

External Reviewers

Grégory Batt
Luca Bortolussi
Christopher Brown
Franck Delaplace
Francesco Fabris
Cédric Lhoussaine
Henning Mortveit
Masahiko Nakatsui
Wei Niu
Masayuki Noro
Andrea Sgarro
Yasuhiro Suzuki
Alan Veliz-Cuba
Andreas Weber
Osvaldo Zagordi
Jun Zhang

Sponsors

Austrian Grid
National Institute of Advanced Industrial Science and Technology (AIST)
Johann Radon Institute for Computational and Applied Mathematics (RICAM)
RISC Software GmbH
Special Research Program SFB F013 of the Austrian Science Fund (FWF)
Upper Austrian Government

Table of Contents

Structural Alignments of RNA Sequences (Invited Talk) 1
Kiyoshi Asai, Hisanori Kiryu, Yasuo Tabei, and Toutai Mituyama

A Biochemical Calculus Based on Strategic Graph Rewriting 9
Oana Andrei and Hélène Kirchner

Determining Flexibility of Molecules Using Resultants of Polynomial Systems 14
Robert H. Lewis and Evangelos A. Coutsias

An Algebraic-Numeric Algorithm for the Model Selection in Kinetic Network with Feedback Loop 17
Masahiko Nakatsui, Hiroshi Yoshida, Masahiro Okamoto, and Katsuhisa Horimoto

Recursive Algebraic Modelling of Gene Signalling, Communication and Switching 23
Sergiy Pereverzyev Jr. and Robert S. Anderssen

Analysis of Network Dynamics Including Hidden Variables by Symbolic-Numeric Approach 28
Daisuke Tominaga, Yasuhito Tokumoto, Hiroshi Yoshida, Masahiko Nakatsui, Sachiyo Aburatani, Fuyan Sun, Jun Miyake, and Katsuhisa Horimoto

Structural Alignments of RNA Sequences

Kiyoshi Asai[1 2], Hisanori Kiryu[2], Yasuo Tabei[1 2], and Toutai Mituyama[2]

[1] Department of Computational Biology, Graduate School of Frontier Science,
University of Tokyo, Kashiwa 277-8561, Japan,
[2] Computational Biology Research Center, National Institute of Advanced Science
and Technology, Koto-ku, Tokyo 135-0064, Japan,

Abstract. The information analyses of RNA sequences on the basis of their secondary structures have been rapidly advanced these few years. We briefly review the basic concepts related with RNA sequence analyses and introduce our contributions to recent progress in structural alignments of RNA sequences.

1 Introduction

It has long been believed that protein-coding genes and their regulatory sequences are most important information in genomic DNA sequences. Recent study revealed, however, the existence of a large number of non-protein-coding RNA transcripts in higher eukaryotic cells. Therefore, the needs for information analyses of non-coding RNAs are demanding.

Basic algorithms for nucleotide sequences, similarity searches and multiple alignments, are applicable also to RNA sequences. The evolutional models behind those algorithms assume position-independent occurrences of mutations, insertions and deletions. However, the positions in functional RNA sequences are not independent because of their constraints of the base pairs in the secondary structures. Sequence analysis algorithms that consider their secondary structures are necessary because it is known that non-coding RNAs often conserve their structures rather than their primary sequences.

An obvious approach for analysis of RNA sequences is to use the predicted secondary structures for similarity searches or alignments along with the sequences themselves, but the results are not always reliable because of inaccurate secondary structure predictions. Another approach is to simultaneously predict the common secondary structure in the alignment process. The computational costs of such algorithms, however, were generally expensive. For structural alignment for example, Sankoff's algorithm [1] proposed in 80ś requires $O(L^4)$ in memory and $O(L^6)$ in time for a pair of sequences of length L. The situation drastically has changed because a number of novel algorithms are proposed in recent few years. In this talk, our contributions to recent progress in algorithms of structural alignments of RNA sequences are introduced.

2 Basics on RNA Sequence Analyses

2.1 The energy model and base pairing probabilities

The secondary structure of an RNA sequence comprises the base pairs that are distantly located in the primary sequence and that form hydrogen bonds. They are typically either the pairs of G-C, A-U or G-U.

Let $E(\sigma, x)$ the free energy of the secondary structure σ of a sequence x. It can be calculated as the sum of all the "loops", the areas closed by base pairs, of the secondary structures, using the energy parameters collected by experiments (eg. Mathews et al. [2]): According to the theory of thermodynamics, the probability $P(\sigma|x)$ that a sequence x forms a secondary structure σ can be written as Boltzmann distribution,

$$P(\sigma|x) = \frac{1}{Z(x)} \exp \frac{-E(\sigma, x)}{RT}, \tag{1}$$

where R is the gas constant, T is the temperature, and $Z(x)$ is a partition function, a sum over a set of all possible secondary structures Ω:

$$Z(x) = \sum_{\xi \in \Omega} \exp \frac{-E(\xi, x)}{RT}. \tag{2}$$

By taking the sum over all the secondary structures that include (x_i, x_j) as one of their base pairs, we can define $P^{bp}(i,j)$, the probability that position x_i and x_j form a base pair, as

$$P^{(bp)}_{i,j} = P((i,j) \in \sigma|x) = \sum_{\sigma|(i,j)\in\sigma} P(\sigma|x). \tag{3}$$

The matrix $\{P^{(bp)}_{i,j}\}$ is called the base pairing probability (BPP) matrix. McCaskill [3] introduced an Dynamic Programming (DP) algorithm computing BPP matrix, which requires $O(L^2)$ in space and $O(L^3)$ in time for an RNA sequence of length L.

2.2 Sequence alignments with maximum expected accuracy

The standard procedure to compute the optimal alignment of two sequences is to compute the alignment that maximize the similarity score of the alignment. However, the maximum similarity alignment is not the alignment that maximize the accuracy of the alignment.

The Maximum Expected Accuracy (MEA) alignment was originally proposed for sequence alignment by Miyazawa [4] in 1995 and re-introduced by Holmes and Durbin [5] and a textbook [6]. The idea was implemented in ProbCons [7], one of the most accurate multiple sequence alignment programs.

The substitution matrix for sequence alignments is defined according to substitution rates of residues, typically based on an evolutional model. Therefore, the score of an alignment is interpreted as the log probability that two sequences are obtained from an ancestor sequence, and the probability of the alignment A can be written as:

$$P(A) = \frac{1}{Z(x,y)} \exp \frac{S(A)}{k} \tag{4}$$

where k is a constant and $Z(x,y)$ is a partition function written as

$$Z(x,y) = \sum_{A} \exp \frac{S(A)}{k}. \tag{5}$$

It can be observed that (4) has the same form of (1) and that $S(A)$ behaves as negative of the energy. By taking the sum over all the alignments that include the match of x_i and y_j, we can define the posterior probability

$$P(x_i \sim y_j | x, y) = \sum_{A | x_i \sim y_j} P(A) \tag{6}$$

where $x_i \sim y_j$ means that x_i and y_j are aligned in the same column in the alignment. This posterior probability can be calculated by DP, typically implemented by pair hidden Markov models.

The accuracy of the alignment are defined according to the number of correctly aligned pairs of the characters. The expected number of correctly aligned positions in the alignment A can be written as

$$E^c(A) = \sum_{x_i \sim y_j \in A} P(x_i \sim y_j | x, y) \tag{7}$$

where $x_i \sim y_j$ denote that x_i and y_j are aligned.

The alignment that maximize $E^c(A)$ can be obtained by very simple DP whose recursion is:

$$E^c_{i,j} = \max \begin{cases} E^c_{i-1,j-1} + P(x_i \sim y_j | x, y) \\ E^c_{i,j-1} \\ E^c_{i-1,j-1} \end{cases} \tag{8}$$

with initial conditions $E^c_{0,*} = E^c_{*,0} = 0$.

2.3 Secondary structure prediction of RNAs

The prediction of the secondary structure of an RNA sequence, estimation of all base pairs in its secondary structure, is a classical problem in bioinformatics.

Nussinov and Jacobson [8] proposed a DP algorithm that maximizes the number of base pairs for a given RNA sequence. For an RNA sequence x, the recursion of DP can be written as

$$M_{i,j} = \max \begin{cases} M_{i-1,j} \\ M_{i,j-1} \\ M_{i-1,j-1} + 1 \\ M_{i,k} + M_{k+1,j} \text{ for } i < k < j-1, \end{cases} \tag{9}$$

where $M_{i,j}$ is the maximum number of base pairs in the subsequence $x_i \dots x_j$, and $M_{i,i} = 0$ for $1 \leq i \leq |x|$. The computational complexities are $O(L^2)$ in memory for DP matrix M, and $O(L^3)$ in time for the iteration of the fourth term of max-operator in (9).

Zuker and Stiegler [9] proposed a DP algorithm that finds the Minimum Free Energy (MFE) structure of an RNA sequence and implemented it to mfold program [10]. The MFE secondary structure is interpreted as the maximum likelihood extimator (MLE) because minimum $E(\sigma, x)$ gives maximum $P(\sigma|x)$ in (1). Zuker and Stiegler's algorithm is much more complicated than that of Nussinov and Jacobson, because it uses two DP matrices and handles the stacking energy. The order of the computational complexities, however, is the same $O(L^2)$ in memory for DP matrix M, and $O(L^3)$ in time.

In the evaluation of the secondary structure predictions, the accuracy is evaluated by the number of correctly predicted base pairs. Do et al. [11] proposed the CONTRAfold, a secondary structure prediction program based on MEA principle. They define the accuracy according to the sum of the number of correctly predicted paired positions and γ times of the number of correctly predicted unpaired bases. The maximized expected accuracy is obtained by simple DP with a recursion, which is very similar to Nassinov et al.'s:

$$M_{i,j} = \max \begin{cases} q_i + M_{i+1,j} \\ q_j + M_{i,j-1} \\ 2\gamma P^{(bp)}_{i,j} + M_{i-1,j-1} \\ M_{i,k} + M_{k+1,j} \text{ for } i < k < j, \end{cases} \tag{10}$$

where $P^{(bp)}_{i,j}$ is the BPP matrix and $q_i = 1 - \sum_j P^{(bp)}_{i,j}$. The parameter γ controls the balance of the sensitivity and the specificity of the prediction.

3 Structural alignment of RNA sequences and their common structures

3.1 Previous works

Though there are a number of programs for secondary structure prediction from a single RNA sequence, the predictions are often inaccurate for further analysis. Therefore, structural sequence alignments based on the predicted secondary structure of each sequence are not reliable. In a structural sequence alignment, it is desirable to simultaneously optimize the alignment and the associated common secondary structure. To solve this problem Sankoff [1] proposed an algorithm with $O(L^{2N})$ memory and $O(L^{3N})$ time for N sequences of length L. The computational complexity is hardly acceptable even for a pair of RNA sequences, which require $O(L^4)$ in memory and $O(L^6)$ in time, because the required memory $8.1GB$ for the sequences of length 300nt exceeds the capacity of the standard computers.

Therefore, a number of practical variants of Sankoff's algorithm have been studied. They can be categorized into two groups by the ways of scoring secondary structures in the algorithms.

The first group of the algorithms score the structures using the free energy parameters extracted from experiments [2]. The advantage of those algorithms is that their structure predictions are relatively accurate and the drawback is the difficulty of combining information of sequence similarities. This group includes Foldalign [12], Dynalign [13] and PMMulti [14]. Foldalign restrict the difference of the length of the regions to be compared. PMMulti directly compare the BPP matrices. Dyanalign restricts the DP path of the pairwise alignment within W bases from the original position in the sequence and achieved the computational complexity of $O(L^4)$ memory and $O(L^3W^3)$ time. L is the length of the shorter sequence. The time complexity approaches to (L^6) when the difference of the lengths of the two sequences increases.

The second group stochastically score the structures by the pair SCFG. The advantage is capability to automatically determine the parameters that reflect both the alignments and structures from a training dataset. The drawback is their limited accuracies of structure predictions compared with those in the first group. This group comprises the work of Grate in 1995 [15], Stemloc [16] and Consan [17]. Stemloc combines the constraints in the structure space and those in the alignment space, using suboptimal alignment algorithm. Consan restricts the DP region by anchoring points in the DP matrix that have high posterior probabilities in simple sequence alignment.

Until very recent years, there had been proposed a number of methods for structural alignment, but there were no software with practical computational costs AND reasonable accuracy. In these three years, however, there appeared practical solutions for this problem. Here we introcude our own contributions.

3.2 Murlet: A practical variant of Sankoff's algorithm

Kiryu et al. [18] developed Murlet, a practical program for aligning multiple RNA sequences based on Sankoff's algorithm with marked reduction of computation. The key ideas for the reduction are the restrictions of DP region and the efficient scoring functions. In order to reduce the DP region, Murlet restricts the matches of the residues that have higher posterior probabilities, which is calculated by non-structural alignment, than a threshold ϵ. In the structural alignment, the scoring function should include both the alignment score and the secondary structure probabilities. For the match of two base pairs, for example, we need the joint posterior probability that x_i forms a base pair with x_j (denoted by $x_i \diamond x_j$ hereafter), $y_k \diamond y_\ell$, x_i is aligned to y_k (denoted by $x_i \sim y_k$ hereafter) and $x_j \sim y_\ell$. The computation of this joint posterior probability, however, require full $O(L^6)$ in time. An approximated probatility,

$$\begin{aligned}
&\hat{P}(x_i \diamond x_j, y_k \diamond y_\ell, x_i \sim y_k, x_j \sim y_\ell | x, y) \\
&= P(x_i \diamond x_j | x) P(y_j \diamond y_\ell | y) P(x_i \sim y_k | x, y) P(x_j \sim y_\ell | x, y) \qquad (11)
\end{aligned}$$

is instead used in Murlet. First two factors are BPPs, which can be calculated with $O(L^3)$ in time by McCaskill's Algorithm, and the latter two factors are posterior matching probabilities, which can be calculated with $O(L^2)$.

The alignment quality and the accuracy of the consensus structure prediction from the alignment were the highest among the structural alignment programs. Additionally, it was shown that the algorithm can align relatively long RNA sequences that have not been computable by other Sankoff-based multiple alignment algorithms.

3.3 Scarna: An heuristic approaches to structural alignment

Tabei et al. [19] developed Scarna, a fast program for structural alignment of a pair of RNA sequences based on their potential common secondary structures. In Scarna, the ensemble of the secondary structures are represented by a set of stem candidates, which can mutually overlap, based on the BPP matrix of each sequence. In order to reduce the computational costs, the 5' parts and the 3' parts of the stem candidates are aligned separately. Although it breaks the consistency of the base pairing structures, rough consistency are introduced into the score function and the engineered DP of the alignment. The score function includes BPP, the frequency of substitutions as the base pairs, the stacking energy and the difference of the distance in base pairs. The accuracy of Scarna was better than sequence-based methods and compatible to structure-based methods, while it can align a pair of RNA sequences of 1000nt within a minute. The computational complexities of the DP of Scarna are $O(L^2)$ in memory and in time, but it requires BPP matrix that costs $O(L^3)$ in time.

As an extension of Scarna to multiple alignment, Tabei et al. [20] developed MXSCARNA, a fast program for structural multiple alignments.

MXSCARNA works in the following three steps. First the guide tree for the progressive alignment is built based on the pairwise similarities of the RNA sequences. Second the BPP matrices are calculated for all the RNA sequences by McCaskill's algorithm. Those BPPs are used for extracting the potential stems and for the matching scores in the DP of the alignments. Third the RNA sequences are progressively aligned along the guide tree using SCARNA's pairwise alignment algorithm with the score function (11) of Murlet. At the first stage of the progressive alignment, which corresponds to the bottom level of the guide tree, the pairs of RNA sequences are aligned by engineered DP algorithm of SCARNA's pairwise alignment. In each upper-level step of the progressive alignment according to the guide tree, potential stems for groups of RNA sequences are extracted from the averaged BPP matrices.

MXSCARNA's accuracies were at least comparable to those of current state-of-art aligners. In addition, the accuracies of MXSCARNA were robust over a broad range of sequence similarities, whereas the other aligners showed reductions in SPS or MCC. The computational complexities of MXSCARNA were evaluated as $O(N^3L^3)$ in time and $O(N^2L^2)$ in memory for N sequences of length L. In the comparison of execution time for benchmark datasets, MXSCARNA was by far the fastest among the structural aligners and was fast enough for large-scale analyses. MXSCARNA aligns five 1000-base sequences with about a minute, and even 5000-base RNA sequences with acceptable computational costs.

4 Concluding Remarks

We have reviewed basic concepts related with RNA sequence analyses and introduced our recent contributions in structural alignments. Murlet and MXSCARNA have broken the barrier of computational costs of structural alignments and opened the possibility for genome-wide analyses of non-coding sequences based on their potential secondary structures. The source codes and the web server is available at http://software.ncrna.org with the other developed software.

Acknowledgements

This work was supported in part by a Grant-in-Aid for Scientific Research on Priority Areas "Comparative Genomics" from the Ministry of Education, Culture, Sports, Science and Technology of Japan and by the "Functional RNA Project" funded by the New Energy and Industrial Technology Development Organization (NEDO) of Japan. The authors thank Kengo Sato, Michiaki Hamada, Koji Tsuda and other colleagues in the Computational Biology Research Center (CBRC) for useful discussions.

References

1. Sankoff, D.: Simultaneous solution of the RNA folding, alignment and protosequence problems. Siam J. Appl. Math., 45, 810-825 (1985)
2. Mathews, D.H., Sabina, J., Zuker, M., Turner, D.H.: Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. J. Mol. Biol., 288(5), 911-940 (1999)
3. McCaskill, J.: The equilibrium partition function and base pair binding probabilities for RNA secondary structure. Byopolymers, 29, 1105-1119 (1990)
4. Miyazawa, S.: A reliable sequence alignment method based on probabilities of residue correspondences. Protein Engineering, 8, 999-1009 (1995)
5. Holmes, I., Durbin, R.: Dynamic programming alignment accuracy. J. Comput. Biol., 5, 493-504 (1998)
6. Durbin, R., Eddy, S.R., Krogh, A., Mitchison, G.: Biological Sequence Analysis. Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press (1998)
7. Do, C.B., Mahabhashyam, M.S.P., Brudno, M., Batzoglou, S.: ProbCons: Probabilistic consistency-based multiple sequence alignment. Genome Research, 15, 330-340 (2005)
8. Nussinov, R., Jacobson, A.: Fast algorithm for predicting the secondary structure of single-stranded RNA. Proc. Natl. Acaed. Sci. U.S.A., 77, 6309-6313 (1980)

9. Zuker, M., Stiegler, P.: Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. Nucleic Acids Res., 9, 133-148 (1981)
10. Zuker, M.: Mfold web server for nucleic acid folding and hybridization prediction. Nucleic Acids Res., 31, 3406-3415 (2003)
11. Do, C.B., Woods, D.A., Batzoglou, S.: CONTRAfold: RNA secondary structure prediction without physics-based models. Bioinformatics, 22, e90-e98 (2006)
12. Gorodokin, J., Stricklin, S.L., Stormo, G.D.: Discovering common stem?loop motifs in unaligned RNA sequences. Nucleic Acids Res., 29, 2135-2144 (2001)
13. Mathews, D.H., Turner, D.H.: Dynalign: an Algorithm for Finding the Secondary Structure Common to two RNA Sequences. J. Mol. Biol., 317, 191-203 (2002)
14. Hofacker, I.L., Bernhart, S.H.F, Stadler, P.F: Alignment of RNA base pairing probability matrices. Bioinformatics, 20, 2222-2227 (2004).
15. Grate, L.: Automatic RNA secondary structure determination with stochastic context-free grammars. Proc. Int. Conf. Intell. Syst. Biol. 3, 136-144 (1995)
16. Holmes, I.: Accelerated probabilistic inference of RNA structure evolution. BMC Bioinformatics, 6, 73 (2005)
17. Dowell R.D., Eddy, S.R.: Efficient pairwise RNA structure prediction and alignment using sequence alignment constraints. BMC Bioinformatics, 7, 400 (2006)
18. Kiryu, H., Tabei, Y., Kin, T., Asai, K.: Murlet: A practical multiple alignment tool for structural RNA sequences. Bioinformatics, 23, 1588-1598 (2007)
19. Tabei, Y., Tsuda, K., Kin, T., Asai, K.: SCARNA: fast and accurate structural alignment of RNA sequences by matching fixed-length stem fragments. Bioinformatics, 22, 1723-1729 (2006).
20. Tabei, Y., Kiryu, H., Kin, T., Asai, K.: A fast structural multiple alignment method for long RNA sequences. BMC Bioinformatics, 9, 33 (2008)

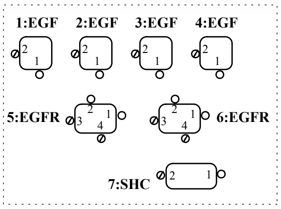
A Biochemical Calculus Based on Strategic Graph Rewriting

Oana Andrei[1] and Hélène Kirchner[2]

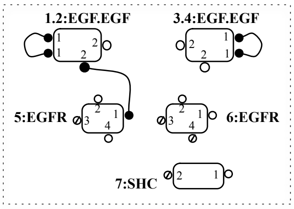
[1] INRIA Nancy Grand-Est & LORIA, France
[2] INRIA Bordeaux Sud-Ouest, France
`First.Last@loria.fr`

When modeling interactions between molecules or proteins, the behaviour of a protein is given by its functional domains that determine which other protein it can bind to or interact with and these domains are usually abstracted as sites that can be bound or free, visible or hidden. Hence a protein is characterized by the collection of interaction sites on its surface and proteins can bind to each other forming molecular complexes. Based on such structures, we considered port graphs [1] which are graphs with ports and with multiple edges and loops attached to ports of nodes. Molecular complexes are port graphs where each port is connected to at most one other port. Such restricted port graphs are called *molecular graphs* and their ports are called *sites*.

We illustrate below a molecular graph G representing the initial state of the system modeling a fragment of the EGFR signaling cascade [2, 3]. The protagonists of this model are three types of proteins: the signal *EGF*, the receptor *EGFR*, and the adapter *SHC*. The molecular graph G' represents a state of the system where two signal proteins are already bound forming a dimer binding in turn a receptor. A node is graphically represent as a box with an unique identifier and a name placed outside the box. A site is represented as a filled, empty, or slashed circle on the surface of the box if its state is respectively bound, free, or hidden.



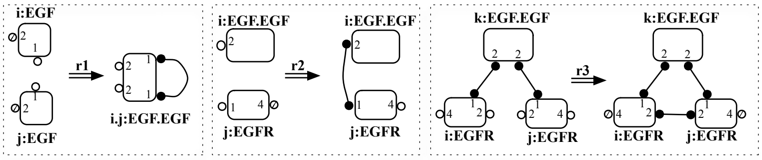
G



G'

A *molecular graph rewrite rule* $L \Rightarrow R$ is a port graph consisting of two molecular graphs L and R called, as usual, the left- and right-hand side respectively, and one special node $\Rightarrow$, called the *arrow node* with ports connected to the sites of L and R such that it embeds the correspondence between elements of L and elements of R. We represent graphically the edges incident to the arrow node only if the correspondence is ambiguous. In consequence, port graphs represent a unifying structure for representing both molecular complexes and the reaction patterns between them.

The five reaction patterns for the EGFR signalling cascade fragment specify the followings: **(r1)** two signaling proteins form a dimer represented as a single node; **(r2)** an EGF dimer and a receptor bind on free sites; **(r3)** two receptors activated by the same EGF dimer bind creating an active dimer RTK; **(r4)** an active dimer RTK activates itself by attaching phosphate groups; **(r5)** an activated RTK binds to an adapter protein activating it as well. They are easily expressible using molecular graph rewrite rules. The first three rules have the following graphical representation:



Let $r : L \Rightarrow R$ be a molecular graph rewrite rule and G_1 a molecular graph such that there is an injective graph morphism g from L to G_1. By replacing the subgraph $g(L)$ for $g(R)$ and connecting it appropriately in the context, we obtain a molecular graph G_2 which represents a result of *one-step rewriting* of G_1 using the rule r, written $G_1 \rightarrow_r G_2$.[3] The formal definition of port graph rewriting is given in [1]. An example of rewriting in the EGFR fragment is the molecular graph G' above obtained from the initial molecular graph G by rewriting it using twice the rule **r1** and once the rule **r2**.

The chemical computation metaphor emerged as a computation paradigm over the last three decades. This metaphor describes computation in terms of a chemical solution in which molecules representing data freely interact according to reaction rules. Chemical solutions are represented by multisets and the computation proceeds by rewritings, which consume and produce new elements according to conditions and transformation rules. The chemical metaphor was proposed as a computational paradigm in the Γ language in [4], then used as a basis for defining the CHemical Abstract Machine (CHAM) [5], and later it was extended to the γ-calculus and HOCL in [6, 7] for modeling self-organizing systems in particular.

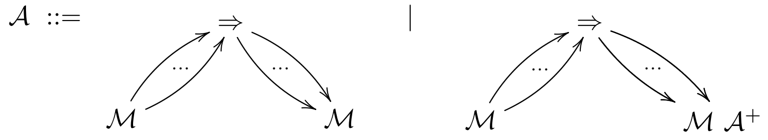
We extend the chemical model with high-level features by considering a port graph structure for the data and the computation rules. The result is a port graph rewriting calculus with higher-order capabilities, called the ρ_{pg}-calculus. The first citizens of the ρ_{pg}-calculus are port graphs, port graph rewrite rules, and rule application. This calculus generalizes the rewriting calculus [8] and the term graph rewriting calculus [9]. The ρ_{pg}-calculus also generalizes the λ-calculus and the γ-calculus through a more powerful abstraction power that considers for matching not only a variable but a port graph with variables.

The ρ_{pg}-calculus is a suitable formalism for modeling systems whose states are port graphs and whose transitions are reductions obtained by applying port graph rewrite rules. Due to the intrinsic parallel nature of rewriting on disjoint

[3] There can be different such morphisms g from L to G_1 leading to different rewrites.

redexes and decentralized rule application, we thus model a kind of *Brownian motion*, a basic principle in the chemical paradigm. In the following we present the main features of the syntax and the semantics of the calculus from a biochemical modeling point of view.

Let $\mathcal{M}$ denote the class of molecular graphs modeling systems states. We denote by $\mathcal{A}$ the class of *abstractions* which are port graph rewrite rules whose left-hand sides are molecular graphs and whose right-hand sides may include other abstractions as well. Both sides are connected through the arrow node. Let $\mathcal{A}^+$ denote a non-empty set of abstractions. Then the abstractions are graphically defined as follows:

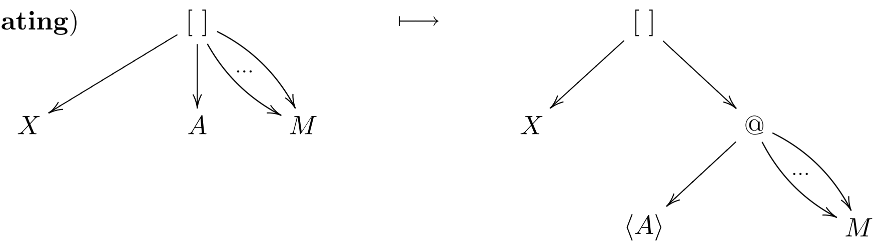


The second type of abstraction enriches the expressivity of the calculus by allowing the application of abstractions to create new molecular graph rewrite rules. This is useful in modeling cellular differentiation: when a particular pattern is found in the system, the application of such an abstraction introduces new rules specializing more the behavior of particular molecular complexes.

The structure modeling the state of the system and the current set of abstractions is itself a port graph built with a node [ ] and distinct auxiliary ports called *handlers* for each node. The handler of the node [ ] is connected to the handlers of all nodes of the molecular graphs and to the handlers of the arrows of all abstractions.

Using a similar mechanism as in the CHAM, an interaction takes place in a system by heating it up. This process isolates an abstraction (or a list of abstractions) and a molecular graph for application and connect them with an application node @. A list of abstractions is defined by a new node ⟨ ⟩ which connects an abstraction and another list of abstractions, possibly empty.

(Heating)



All steps computing the application of abstractions to a molecular graph, including the matching and the replacement operations, are expressible using port graph transformations by considering some more auxiliary nodes and extending the reduction relation. This reduction mechanism is internalized in the calculus.

Instead of having a highly non-deterministic behaviour of molecular graph rewrite rules application, one may want to introduce some control to compose or choose the rules to apply, possibly exploiting failure information. The notion of abstraction is powerful enough to express such control, thanks to the notions of *strategy* and *strategic rewriting* [10]. Strategies are higher-order functions that select rewriting derivations. Various strategy languages have been proposed in ELAN [11], Stratego [12], TOM [13], or Maude [14]. In a strategy language, the basic elements are the rewrite rules and the identity (id) and failure (`fail`) strategies. Based on them, strategies expressing the control can be constructed, like the sequence (`seq`), the left-biased choice (`first`), the application of a strategy only if it is successful (`try`), and the repeating strategy (`repeat`).

In the ρ_{pg}-calculus, strategies are abstractions, hence objects of the calculus. Considering also a failure node stk, we encode for instance the strategies id, `fail`, and `seq` as the following abstractions:

$$\mathsf{id} \triangleq X \Rightarrow X \qquad \mathtt{fail} \triangleq X \Rightarrow \mathsf{stk} \qquad \mathtt{seq}(S_1, S_2) \triangleq X \Rightarrow \langle S_1 \langle S_2 \rangle\rangle @X$$

Then the strategies `first`, `try`, and `repeat` are easily defined using the above strategies and some reduction rules explicitly handling the failure node stk.

Thanks to strategies, the heating rule is reformulated based on a failure catching mechanism as follows: if $\langle S \rangle @M$ reduces to the failure, i.e., to the stk node, then the strategy $\mathtt{try}(\mathsf{stk} \Rightarrow S\ M)$ restores the initial strategy and molecular graph subjects to reduction.

$$(\textbf{Heating}') \qquad [X\ S\ M] \longmapsto [X\ \langle \mathtt{seq}(S, \mathtt{try}(\mathsf{stk} \Rightarrow S\ M))\rangle @M]$$

After the application of a strategy on a molecular graph successfully takes place, a *cooling* rule, the counterpart of the heating rule, is in charge of rebuilding the state of the system by removing the no longer useful application nodes and plugging the result of the (strategic) rewriting in the environment.

The successful application of an abstraction or strategy to a molecular graph produces a new graph, built according to one chosen matching solution.[4]

At this level of definition of the calculus, the strategies are consumed by a non-failing interaction with a molecular graph. One advantage is that, since we work with multisets of port graphs, a strategy can be given a multiplicity, and each interaction between the strategy and the molecular graph consumes one occurrence of the strategy. This permits controlling the maximum number of times an interaction can take place. But sometimes, it may be suitable to have persistence of the information concerning the available abstraction and thus the persistence of a given possible interaction. In this case, the abstraction should not be consumed by the reduction. For that purpose, we define the *persistent* strategy that applies a strategy given as argument and, if successful, replicates itself. Again, we encode this strategy as an abstraction:

$$S! \triangleq X \Rightarrow \langle \mathtt{seq}(S, \mathtt{first}(\mathsf{stk} \Rightarrow \mathsf{stk}, Y \Rightarrow Y\ S!))\rangle @X$$

[4] An alternative would be to consider a structure of all graphs corresponding to the different matching solutions. This would assume a new node for composing possible results with appropriate reduction rules considering such structures. This is not developed here.

Using the capability of strategic rewriting to generate all possible states of a system, the framework can already be used for the verification of some properties (like the presence or the absence of certain molecular graphs) as soon as such properties can be encoded as objects of the calculus. In [15] we showed how the principles of the ρ_{pg}-calculus are expressive enough for modeling systems with self-organizing and emergent properties and illustrated it on a mail delivery system.

For future work, we plan to identify conditions on abstractions for accessibility of stable states of modeled systems, or for imposing fairness on the application of abstractions, and to integrate verification techniques in the calculus. Another interesting feature worth and quite natural to be defined in the calculus represents the possibility of modifying or deleting abstractions as objects of the calculus, with application in modeling cellular dedifferentiation for instance.

References

1. Andrei, O., Kirchner, H.: A Rewriting Calculus for Multigraphs with Ports. In: Proceedings of RULE'07. (2007)
2. Danos, V., Laneve, C.: Formal molecular biology. TCS **325**(1) (2004) 69–110
3. Laneve, C., Tarissan, F.: A simple calculus for proteins and cells. ENTCS **171**(2) (2007) 139–154
4. Banatre, J.P., Metayer, D.L.: A new computational model and its discipline of programming. Technical Report RR-566, INRIA (1986)
5. Berry, G., Boudol, G.: The Chemical Abstract Machine. TCS **96**(1) (1992) 217–248
6. Banâtre, J.P., Fradet, P., Radenac, Y.: A Generalized Higher-Order Chemical Computation Model. ENTCS **135**(3) (2006) 3–13
7. Banâtre, J.P., Fradet, P., Radenac, Y.: Programming Self-Organizing Systems with the Higher-Order Chemical Language. International Journal of Unconventional Computing **3**(3) (2007) 161–177
8. Cirstea, H., Kirchner, C.: The rewriting calculus - Part I and II. Logic Journal of the IGPL **9**(3) (2001) 427—498
9. Bertolissi, C., Baldan, P., Cirstea, H., Kirchner, C.: A Rewriting Calculus for Cyclic Higher-order Term Graphs. ENTCS **127**(5) (2005) 21–41
10. Kirchner, C., Kirchner, F., Kirchner, H.: Strategic computations and deductions. In: Festchrift in honor of Peter Andrews. (2008)
11. Borovanský, P., Kirchner, C., Kirchner, H., Ringeissen, C.: Rewriting with strategies in ELAN: a functional semantics. Int. J. Found. Comput. Sci. **12**(1) (2001) 69–98
12. Visser, E.: Stratego: A Language for Program Transformation based on Rewriting Strategies. System Description of Stratego 0.5. In Middeldorp, A., ed.: RTA 2001. Volume 2051 of Lecture Notes in Computer Science., Springer-Verlag (2001) 357–361
13. Balland, E., Brauner, P., Kopetz, R., Moreau, P.E., Reilles, A.: Tom: Piggybacking rewriting on java. In: RTA 2007. Volume 4533 of Lecture Notes in Computer Science., Springer-Verlag (2007) 36–47
14. Martí-Oliet, N., Meseguer, J., Verdejo, A.: A Rewriting Semantics for Maude Strategies. In: Proc. of WRLA'08. (2008)
15. Andrei, O., Kirchner, H.: Strategic Port Graph Rewriting for Autonomic Computing. In: TFIT. (2008)

Determining Flexibility of Molecules Using Resultants of Polynomial Systems

Robert H. Lewis[1] and Evangelos A. Coutsias[2]

[1] Fordham University, New York, NY 10458, USA
[2] University of New Mexico, Albuquerque, NM 87131, USA

We solve systems of multivariate polynomial equations in order to understand flexibility of three dimensional objects, including molecules.

Generic protein flexibility has been a major research topic in computational chemistry for a number of years and it has a key role for many important functions of proteins as molecular machines [10]. In general, a polypeptide backbone can be modeled as a polygonal line whose edges and angles are fixed while some of the dihedral angles formed by successive triplets of edges can vary freely. It is well known that a segment of backbone both of whose ends are fixed will be (generically) flexible if it includes more than six free torsions. Resultant methods have been applied successfully to this problem, see [3], [4] and the references therein. In this work we focus on non-generically flexible structures that are rigid but become continuously movable if certain symmetries and relations exist. In 1812, Cauchy considered flexibility of three dimensional polyhedra, where each joint can pivot or hinge. He proved that if the polyhedron is convex it must be rigid [2]. But following Bricard's study of flexible non-convex intercrossing octahedra [1], in 1978 Connelly and others found non-convex flexible polyhedra [5] that can be imbedded in 3 dimensions without self-crossing faces.

In our previous work [8], we began a new approach to understanding flexibility, using symbolic computation instead of numerical calculation. We describe the geometry of the object or molecule with a set of multivariate polynomial equations. Solving a system of multivariate polynomial equations is a classic, difficult problem. The approach via resultants was pioneered by Bezout, Sylvester, Dixon [7], and others [6]. The resultant *res* appears as a factor of the determinant *det* of a matrix containing multivariate polynomials. But often *det* is too large to compute or factor, even though *res* is relatively small. We developed a heuristic that overcomes the problem [9]. Given the resultant, we described [8] an algorithm that examines *res* and determines relations for the structure to be flexible.

We discovered in this way the conditions of flexibility for a significant arrangement of quadrilaterals in [1]. The system in our original formulation had six equations in six variables and eleven parameters. The resultant *res*, a function of one variable *ca* and the eleven parameters, has 190981 terms. If the figure is flexible, there are infinitely many possible values for *ca*. That implies that every coefficient relative to *ca* in *res* must vanish. We developed an algorithm *Solve* to search for relations among the parameters that will kill these coefficients and so produce flexibility. As lengths of sides in a geometric figure, the parameters cannot be zero, nor can there be relations using only negative coefficients. These facts simplify the algorithm. *Solve* succeeds in three minutes on a desktop computer.

1 First new result

We have now analyzed Bricard's original formulation of the problem [1] in terms of three equations, with fifteen parameters. This has several advantages, not the least of which is that this system of polynomials also describes the conformational problem of the octahedron, a special case of which describes the cyclohexane molecule [4]. But in contrast to the previous set of equations, some of the parameters can be negative or zero. We have modified algorithm *Solve* to include these cases, with great success. Although the physically meaningful flexible conformations of the cyclohexane are well known, this appears to be the first fully algebraic approach for their derivation, as well as for deriving Bricard's flexible octahedra.

2 Second new result

Next we consider the cylo-octane molecule, pictured in figure 1. Chemically relevant solutions fix the (bond) angles between light lines introducing four constraint equations in the variables τ_i.

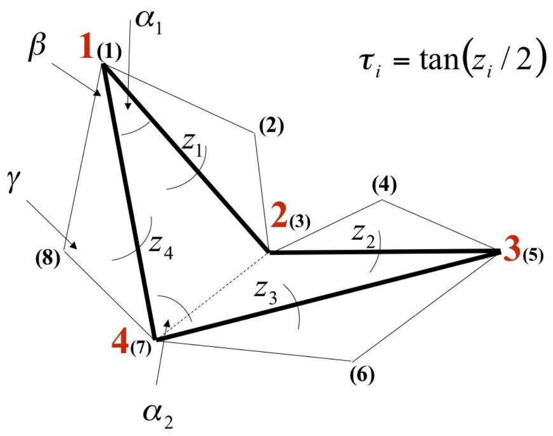


Fig. 1. Geometry of Octane Molecule.

To save space, we show one equation here; the other three are similar:

$$-t\beta^4\,\tau_4{}^2\,\tau_1{}^2-4\,t\alpha_1\,t\beta^3\,\tau_4{}^2\,\tau_1{}^2+6\,t\beta^2\,\tau_4{}^2\,\tau_1{}^2+4\,t\alpha_1\,t\beta\,\tau_4{}^2\,\tau_1{}^2-\tau_4{}^2\,\tau_1{}^2-t\beta^4\,\tau_1{}^2+4\,t\alpha_1{}^2\,t\beta^2\,\tau_1{}^2+2\,t\beta^2\,\tau_1{}^2-\tau_1{}^2-8\,t\alpha_1{}^2\,t\beta^2\,\tau_4\,\tau_1-8\,t\beta^2\,\tau_4\,\tau_1-t\beta^4\,\tau_4{}^2+4\,t\alpha_1{}^2\,t\beta^2\,\tau_4{}^2+2\,t\beta^2\,\tau_4{}^2-\tau_4{}^2-t\beta^4+4\,t\alpha_1\,t\beta^3+6\,t\beta^2-4\,t\alpha_1\,t\beta-1=0$$

Here $\tau_i = \tan(z_i/2)$, $t\beta = \tan(\beta/2)$, and $t\alpha_i = \tan(\alpha_i/2)$.

We use the Dixon resultant to eliminate τ_2, τ_3, and τ_4. An important special case is when the basic quadrilateral (heavy black lines) is planar. The equations simplify quite a bit, and we describe all the solutions of this case.

In the general case we have also made significant progress. The determinant of the Dixon matrix here, were it ever computed, would have many billions of terms. But our techniques [9] discover its hundreds of factors in about 60 hours of CPU time. We verify some known chemical arrangements. We discuss new interesting flexible cases.

References

1. Bricard, Raoul, Mémoire sur la théorie de l'octaèdre articulé, J. Math. Pures Appl. 3 (1897), p. 113 - 150
(English translation: http://www.math.unm.edu/∼vageli/papers/bricard.pdf).
2. Cauchy, A. L. Sur les polygones et les polyhedres. Second Memoire. Journal de l'École Polytechn. **9** (1813), pp. 8.
3. Coutsias, E. A., C. Seok, M. P. Jacobson and K. A. Dill A Kinematic View of Loop Closure, J. Comput. Chem., 25 (2004), no. (4), p. 510 - 528.
4. Coutsias, E. A., C. Seok, M. J. Wester and K. A. Dill, Resultants and loop closure, Int. J. Quantum Chem. 106 (2005), no. (1), p. 176 - 189.
5. Cromwell, P. R. *Polyhedra.* New York: Cambridge Univ. Press, 1997. p. 222 - 224.
6. Cox, D., J. Little, D. O'Shea. Using Algebraic Geometry. Graduate Texts in Mathematics, 185. Springer-Verlag. New York, 1998.
7. Dixon, A. L. The eliminant of three quantics in two independent variables. Proc. London Math. Soc., **6** (1908) p. 468 - 478.
8. Lewis, R. H. and E. A. Coutsias, Algorithmic Search for Flexibility Using Resultants of Polynomial Systems; in Automated Deduction in Geometry, 6th International Workshop, ADG 2006. Springer-Verlag. LNCS **4869** p. 68 - 79 (2007).
9. Lewis, R. H., Heuristics to accelerate the Dixon resultant. Mathematics and Computers in Simulation **77**, Issue 4, p. 400-407, April 2008.
10. Thorpe, M., M. Lei, A. J. Rader, D. J. Jacobs, L. Kuhn. Protein flexibility and dynamics using constraint theory. J. Molecular Graphics and Modelling **19** (2001) p. 60 - 69.

An Algebraic-Numeric Algorithm for the Model Selection in Kinetic Network with Feedback Loop

Masahiko Nakatsui[1], Hiroshi Yoshida[2], Masahiro Okamoto[3], and Katsuhisa Horimoto[1]

[1] Computational Biology Research Centre (CBRC), National Institute of Advanced Industrial Science and Technology (AIST), Aomi 2-42, Koto-ku, Tokyo 135-0064, Japan

[2] Faculty of Mathematics, Organization for the Promotion of Advanced Research, Kyushu University, Hakozaki 6-10-1, Higashi-ku, Fukuoka 812-8581 Japan

[3] Laboratory of Bioinformatics, Graduate School of Systems Life Sciences, Kyushu University, Hakozaki 6-10-1, Higashi-ku, Fukuoka 812-8581 Japan

1 Introduction

Recently, we have proposed a novel algorithm to select a model that is the most consistent with the time series of observed data [4]. In the algorithm, first, a system of differential equations that express the kinetics for a biological phenomenon and a sum of exponentials that are fitted to the observed data are transformed into the corresponding system of algebraic equations, by the Laplace transformation. Then, the two systems of algebraic equations are compared by a algebraic-numeric approach. One of the merits of our algorithm estimates the model's consistency with the observed data and the determined kinetic constants. Furthermore, our algorithm allows a kinetic model with cyclic relationships between variables that cannot be handled by the usual approaches. In this paper, we improve the previous algorithm, especially the part of numerical computation, and address the issue on the selection of model with the feedback loop, which is regarded as one of the difficult issues in the model selection by numerical approach. Two models, a chain graph and a graph with a feedback loop, are examined with the corresponding simulated data, and the plausibility of the improved method is illustrated in comparison with the model selection by numerical approach.

2 Methods

2.1 Overview of Model Selection Algorithm

The procedure for model selection can be summarized as follows:

(i) We fit the observed data as a sum of exponentials in 2.2.

(ii) We perform the Laplace-transformation of both the system of differential equations for the models and the sum of exponentials for the observed data in 2.3.

(iii) By using the least squares method (abbreviated as *LSM*), we calculate the consistency of the model with the observed data.

In what follows, the details of our method will be shown.

2.2 Observed Data Fitting by Genetic Algorithm (GA)

In this paper, we need Laplace-transformed observed data, because we perform the model selection over the Laplace domain. Let $Mo_i(t)$ denote the observed data corresponding to $M_i(t)$ derived theoretically. By genetic-algorithm based numerical fitting, $Mo_i(t)$ is expressed in terms of a sum of exponentials as follows:

$$\beta_b + \sum_{j=1}^{n} \beta_j \exp(-\alpha_j t), \tag{2.1}$$

where n is the number of distinct exponentials determined by $M_i(t)$, and β_b is zero in the case of the non-existence of a constant term within $M_i(t)$. $Mo_i(t)$ thus fitted is changed into the Laplace-transformed data as follows:

$$\frac{\beta_b}{s} + \sum_{j=1}^{n} \frac{\beta_j}{s + \alpha_j}, \tag{2.2}$$

where L denotes the Laplace transformation. In this problem, each set of parameter values α_i, β_i and β_b to be estimated is evaluated using the following procedure: Suppose that $Mo_i(t)$ is the calculated time-course at time t of i and that $Ms_i(t)$ represents sampling data at time t of i. The sum of the square values of the relative error between $Mo_i(t)$ and $Ms_i(t)$ gives the total relative error E_i;

$$E_i = \sum_{t=1}^{T} \left(\frac{Ms_i(t) - Mo_i(t)}{Ms_i(t)} \right)^2, \tag{2.3}$$

where T is the total number of sampling points.

The computational task is to determine a set of parameter values α_i, β_i and β_b that minimizes the objective function E_i. Instead of the use of `NMinimize` command of `Mathematica 5.2` in the previous study [4], here, we use the well-known genetic algorithm (GA). We applied RCGAs with a combination of *unimodal normal distribution crossover* (UNDX) [1] and *minimal generation gap* (MGG) [2] as a nonlinear numerical optimization method for estimating constants.

2.3 Laplace-transformation of Model Formula

Suppose that the model formulae are described over the time domain as follows:

$$\frac{\mathrm{d}M_i(t)}{\mathrm{d}t} = F_i(\vec{M}, \vec{k}), \tag{2.4}$$

where $\overrightarrow{M} = \{M_1, M_2, \ldots, M_n\}$ and $\overrightarrow{k} = \{k_1, k_2, \ldots, k_m\}$. Function $F_i(\overrightarrow{M}, \overrightarrow{k})$ can be determined according to the graph describing the model, and $\overrightarrow{k}$ denotes the kinetic constants between the chemicals. We transform this system of differential equations into a system of algebraic equations over the Laplace domain, and solve the equations in $L[M_i(t)](s)$ $(i = 1, 2, \ldots, n)$.

2.4 Calculation of Consistency Measure

To evaluate the consistency of the model with the observed data, we define *consistency measure*s. If the model is completely consistent with the observed data and the data lack noise and inaccuracies, then $L[M_i(t)](s) = L[Mo_i(t)](s)$ $(i = 1, 2, \ldots, n)$ holds. This fact has led us to the following definitions of consistency measure:

Let *comp* denote the set of polynomials obtained by matching the coefficients of $L[M(t)](s)$ and $L[Mo(t)](s)$ over the Laplace domain, in which every element is zero in the case of $L[M_i(t)](s) = L[Mo_i(t)](s)$ $(i = 1, 2, \ldots, n)$; that is, when Formula $L[M_i(t)](s) = L[Mo_i(t)](s)$ is an identity in s.

The consistency measure of the model is defined as the smallest sum-square value of the elements in *comp* with non-negative kinetic constants. In order to obtain the smallest value, we have utilized the least squares method using the following equations:

$$\frac{\partial}{\partial k_1} g(\overrightarrow{k}) = \frac{\partial}{\partial k_2} g(\overrightarrow{k}) = \cdots = \frac{\partial}{\partial k_m} g(\overrightarrow{k}) = 0, \tag{2.5}$$

where $g(\overrightarrow{k})$ is the sum-square value of the elements in *comp*.

Then, we survey all of the possible candidates of the minimum by calculating *all* of the real positive roots of the system of algebraic equations (2.5). Several methods and tools exist to calculate all real roots of algebraic equations adjoined by a zero-dimensional ideal. In the previous study [4], we simply used a command, `NSolve` in `Mathematica 5.2`.

Using the consistency measures, we performed model selection. We, first, calculated the consistency measures of the candidate models with the observed data. Then, we listed the smallest consistency measures and the corresponding values of kinetic constants of each candidate model for the consistent measures.

3 Results

3.1 Models and Formulations

Fig. 1 shows the two models analyzed in this paper. One is a model of a chain graph with four nodes $(M_1, M_2, M_3$, and $M_4)$ (Fig. 1(a)), and the other is a model of a graph that a feedback loop between M_4 and M_1 is added in the chain graph (Fig. 1(b)). As easily seen from the figure, the former model is a *subgraph* of the latter model.

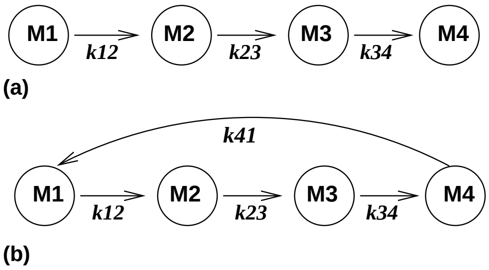


Fig. 1. Models

According to the models in Fig. 1, the kinetics can be expressed by two systems of differential equations as follows:

Model (A)

$$\begin{cases} \mathrm{d}/\mathrm{d}t\ M_1(t) = -k_{12}\ M_1(t), \\ \mathrm{d}/\mathrm{d}t\ M_2(t) = k_{12}\ M_1(t) - k_{23}\ M_2(t), \\ \mathrm{d}/\mathrm{d}t\ M_3(t) = k_{23}\ M_2(t) - k_{34}\ M_3(t), \\ \mathrm{d}/\mathrm{d}t\ M_4(t) = k_{34}\ M_3(t). \end{cases} \tag{3.1}$$

Model (B)

$$\begin{cases} \mathrm{d}/\mathrm{d}t\ M_1(t) = -k_{12}\ M_1(t) + k_{41}\ M_4(t), \\ \mathrm{d}/\mathrm{d}t\ M_2(t) = k_{12}\ M_1(t) - k_{23}\ M_2(t), \\ \mathrm{d}/\mathrm{d}t\ M_3(t) = k_{23}\ M_2(t) - k_{34}\ M_3(t), \\ \mathrm{d}/\mathrm{d}t\ M_4(t) = k_{34}\ M_3(t) - k_{41}\ M_4(t). \end{cases} \tag{3.2}$$

Then the above differential equations are transformed into the corresponding systems of algebraic equations by the Laplace transformation.

3.2 Data Generation and Fitting

We generated the data for the simulation study. The initial conditions for each molecules and the kinetic constants are set as follows: $M_1(0) = 10, M_2(0) = 7, M_3(0) = 3$, $M_4(0) = 1$, $k_{12} = 165/1508$, $k_{23} = 1/29$, $k_{34} = 1/13$ and $k_{41} = 3/1508$. By using these kinetic constants, we sampled the data for examining the models. Since the digits of the constants are different in the above sets of equations, we sampled the data at 100 points when t is in the range from 0 to 10, at 100 points when t is from 10 to 30, and at 70 points when t is from 30 to 100. Furthermore, 5% of fluctuation is added for each data as the noise of data.

Results of fitting by using GA, two sets of generated data are fitted well to two different models.

3.3 Model Selection by Algebraic-Numeric Approach

To examine the performance of our method about the feedback-loop model, we selected one model among the two models with the data generated from one model. Table 1 shows the consistency of the models with the data from the two models by the consistency measure, together with the estimated values of kinetic constants. As for the selection by data from model (A), the smallest ssq's and the kinetic constants show similar values in the two models. However, the values of k_{41} is estimated to be exact zero. The exact zero value of k_{41} indicates that the consistent structure of model (B) is equal to that of model(A). Thus, when the data are observed from model (A) and the question is whether there exists the feedback loop between molecules M_4 and M_1 in the kinetics, our method can select a correct model. Unfortunately, our method dose not operate well, when the data are observed from model (B). Indeed, the ssq is smaller in the case when the model (A) is examined than when the model (B) is examined. As for the estimated kinetic constants, the values of k_{12} and k_{23} are similar to the given values for the data generation in the case when the model (B) is examined, but the value of k_{34} is similar to a given value when the model (A) is examined. Thus, in this case, our method does not discriminate the models with and without the feedback loop.

In summary, our method is useful in the case of the *true* model without the feedback loop, but is not in the case of the *true* model with the feedback loop. In other words, our method can detect the absence of feedback loop but fails in the detection of the existence of feedback loop.

data-generating model	examined model	smallest ssq	k_{12}	k_{23}	k_{34}	k_{41}
(A)	(A)	0.0142	0.104	0.0329	0.0769	–
(A)	(B)	0.0172	0.106	0.0334	0.0748	0^*
(B)	(A)	0.00210	0.0996	0.0276	0.0653	–
(B)	(B)	0.00324	0.109	0.0304	0.0625	0.0135

Table 1. Consistency measure with kinetic constants. The given values of kinetic constants are $k_{12} = 165/1508(\sim 0.109)$, $k_{23} = 1/29(\sim 0.0345)$, $k_{34} = 1/13(\sim 0.0769)$ and $k_{41} = 3/1508(\sim 0.00199)$. The symbol '$0^*$' indicates the exact value of zero.

4 Discussion

We examined the performance of our improved method for selecting the model with the feedback loop by using two models and the corresponding simulated data, we have partly succeeded in selecting the model with the feedback loop.

Note that the present performance is examined by one set of data generated from the given values of kinetic constants. In particular, the present kinetic constants for the feedback effect are relatively small in comparison with the remaining kinetic constants (see in Table 1). This small effect might cause the partial success in the present study. At any rate, we should further test the performance of our method for the generated data by different kinetic constants as well as for actually observed data. Furthermore, we should test the performance of our method for various structures of models, such as the five network motifs classified by Shen-Orr, S. S. *et al*[3].

References

1. Ono, I. and Kobayashi, S.: A real-coded genetic algorithm for function optimization using unimodal distribution crossover, *Proc 7^{th} ICGA*, (1997) 249–253.
2. Satoh, H., Ono, I. and Kobayashi, S.: A new generation alternation model of genetic algorithm and its assessment, *J. of Japanese Society for Artificial Intelligence*, 15(2) (1997) 743-744.
3. Shen-Orr, S. S., Milo, M., Mangan, S and Alon, U.: Network motifs in the transcriptional regulation network of *Escherichia coli*, *Nature Genetics*, Vol. 31 (2002), 64–68.
4. Yoshida, H., Nakagawa, K., Anai, H. and Horimoto, K.: An Algebraic-Numeric Algorithm for the Model Selection in Kinetic Networks. *Proceedings of 10th CASC, Lecture Notes in Computer Science*, Vol. 4770 (2007), 433–447, Springer-Heidelberg.

Recursive Algebraic Modelling of Gene Signalling, Communication and Switching

Sergiy Pereverzyev Jr.[1] and Robert S. Anderssen[2]

[1] Industrial Mathematics Institute, Johannes Kepler University Linz,
Altenbergerstrasse 69, A-4040 Linz, Austria,
pereverzyev@indmath.uni-linz.ac.at,
[2] CSIRO Mathematical and Information Sciences, PO Box 664, Canberra, ACT
2601, Australia,
Bob.Anderssen@csiro.au

Abstract. In the modelling of genetic signalling, communication and switching (GSCS) at the cellular level, there is a need to identify the various mechanistic models, which nature has discovered, in terms of simple positional information rules (Wolpert (1969)) with the fate of cells being determined by their neighbours (Pennell et al. (1999)). The discovery of such simple rules, however, is a highly non-trivial process; in part, because of the complexity of the plethora of organs and organisms that such protocols are able to construct; and, in part, because the rules will not be universal (e.g. in leaves, the genetic dynamics within epidermal cells is different to that within their trichomes).

Based on Young's model (1983) for pea leaf development, we propose a framework for the explorative recursive algebraic mechanistic modelling of the GSCS control of the specific plant development process related to the positioning of trichomes on Arabidopsis leaves. In this framework, the leaf is represented as an array of hexagonal tiles (Figure 1). The tile can be seen as either a single cell or a collection of cells. The quantative characteristic of the tile is, as we will call it, a hexagonal number. As in Young's paper, the relation of this number to a measured quantity is currently not specified. (It can however be surmized that this number is related to the concentration of the expression of the gene GL1 (Larkin et al (2003))). In order to have an effective reference to these numbers, we introduce the concept of the horizontal layer of tiles that is the collection of tiles whose centers lie on the same line. Then, the hexagonal number is denoted by $P(i, j)$, where i is the position of the layer counting from the top and j is the position of the tile within the layer counting from the left.

For the tile (i, j), let $n_l = n_l(i, j)$ and $n_r = n_r(i, j)$ denote the position of the tile on the layer $(i - 1)$ of the left and right of this tile, and $n_c = n_c(i, j)$ denote the position on the layer $(i - 2)$ of the tile immediately above. This is illustrated in Figure 2. We propose the following recursive rules for determining values of $P(i, j)$:

1. $P(1, 1) = P(2, 1) = P(2, 2) = 1$.
2. If the tile (i, j) is located on the periphery of the leaf, then $P(i, j) = 1$.
3. If the hexagonal number in one of the neighbors of the tile (i, j) exceeds some threshold T_{gl1} (i.e. if $P(i - 1, n_l) \geq T_{gl1}$, or $P(i - 1, n_r) \geq T_{gl1}$, or $P(i - 2, n_c) \geq T_{gl1}$), then the hexagonal number in this tile is reset to 1

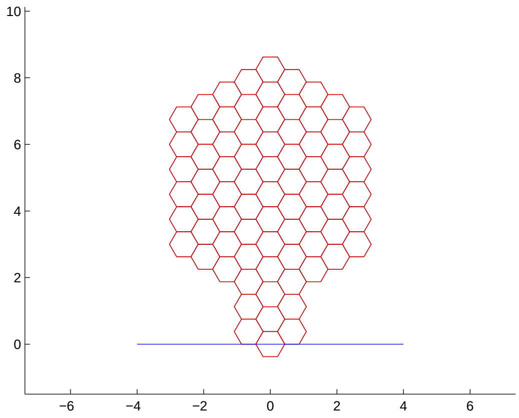


Fig. 1. Representation of the leaf as an array of hexagonal tiles.

(i.e. $P(i,j) = 1$). Otherwise $P(i,j)$ is determined by the following algebraic formula:

$$P(i,j) = P(i-1,n_l) + P(i-2,n_c) + P(i-1,n_r).$$

As shown in Figure 3, the algorithm starts with three hexagonal tiles (Figure 3(a)) and then progressively generates successive layers with the hexagonal numbers determined according to the above rules. Figures 3(b)-(h) show this progression at different stages. The tricomes will initiate in the tiles whose hexagonal number exceeds the threshold.

In Figure 4, it is shown how the regular pattern of trichome initiation, using the above rules, is sensitive to the value of the threshold T_{gl1}. Figure 4(a) shows the hexagonal leaf pattern with trichomes everywhere. Figures 4(b)-(f) show the changing pattern of the trichome initiation when the threshold T_{gl1} varies. As the value of T_{gl1} increases, the number of trichomes decreases (non-monotonically), the positions of the tichomes move down from the leaf tip and the spacing between the various trichomes increases.

The proposed framework has the potential to provide models that simulate not only the development of the wild type but also of known and unknown mutants. This is achieved through small changes in the threshold values, which determine the switching from one cell fate to another. This is fully consistent with the known biology of development.

The overall goal of the above computational modelling is an illustration of how, for an algebraic model of some biological process, to utilize, in an iterative manner, the available biological knowledge in the formulation of a model that captures the essence of the biology being investigated. In particular, it illustrates how, when all factors are taken into account, the resulting model, though quite

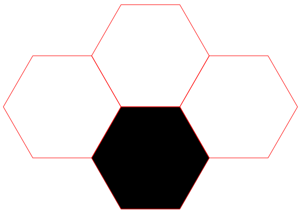


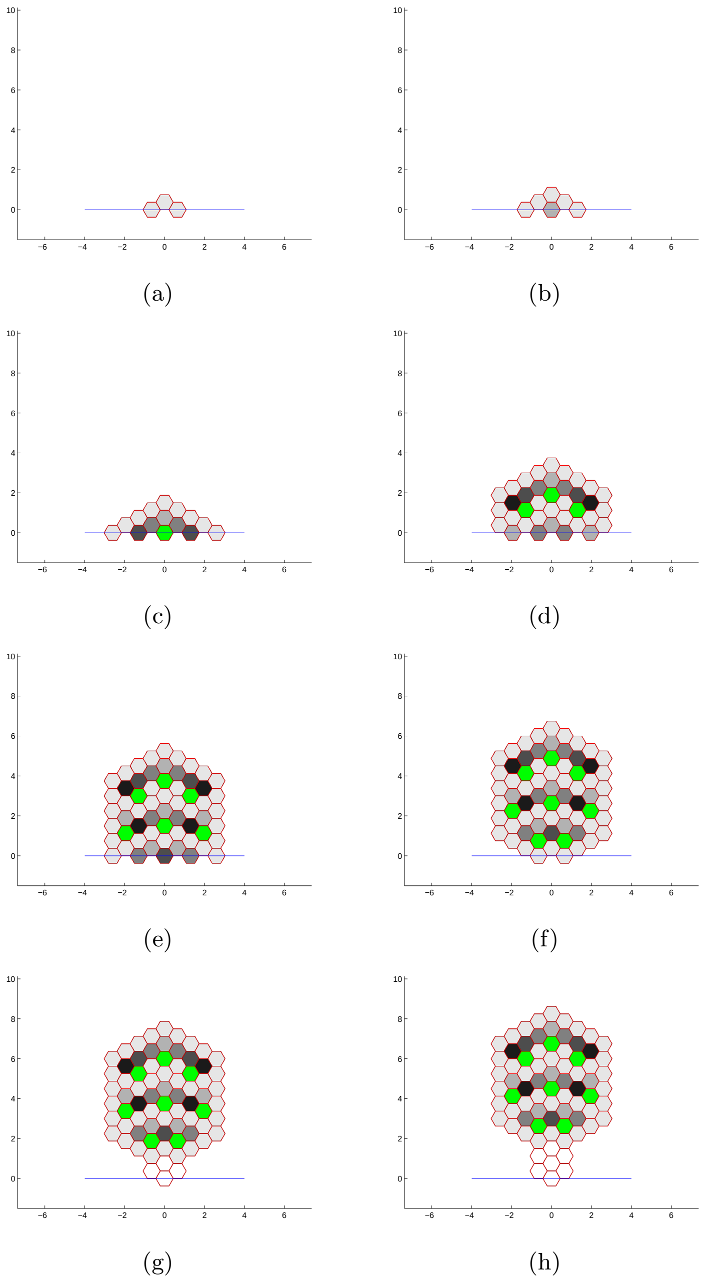
Fig. 2. A tile (i, j) (black one) and its left $(i-1, n_l)$, right $(i-1, n_r)$, and central $(i-2, n_c)$ neighbours.

elementary, is able to generate a plethora of possibilities as occurs in the genetic manipulation of plants.

Consequently, the analysis and interpretation of such a model is not necessary straightforwardly simple. For example, there is a need, on the basis of published results about trichome initiation, positioning, and development, to give a biological interpretation of the threshold(s) and to modify the model (Pereverzyev Jr. and Anderssen (2008)). In addition, the relationship between the number of generated trichomes and the value(s) of the threshold(s) requires further investigation, as does the dependence of this relationship on the total number of cells.

References

1. L. Wolpert (1969) Positional information and spatial pattern of cellular differentiation, J. Theor. Bio. 25, 1-74.
2. R. I. Pennell, Q. C. B. Cronk, L. S. Forsberg, et al. (1995) Cell-context signalling, Phil. Trans. R. Soc. London B 350, 87-93.
3. J. P. W. Young (1983) Pea leaf morphogenesis: A simple model, Annals Bot. 52, 311-316.
4. J. C. Larkin, M. L. Brown and J. Schiefelbein (2003) How do cells know what they want to be when they grow up? Annu. Rev. Plant Biol. 54, 403-430.
5. S. Pereverzyev Jr. and R. S. Anderssen (2008) Recursive algebraic modelling of gene signalling, communication and switching, RICAM Report.



(a)

(b)

(c)

(d)

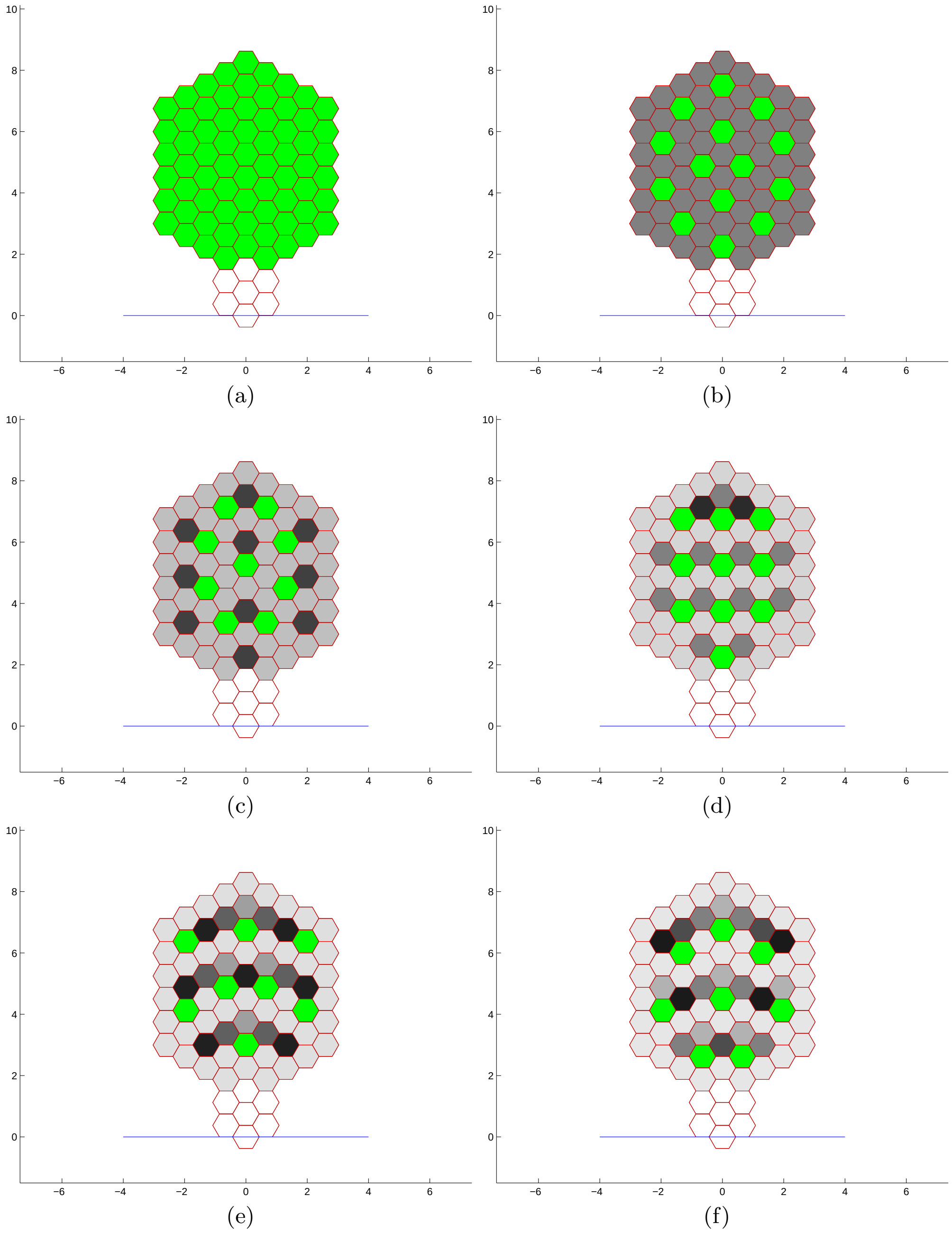
(e)

(f)

(g)

(h)

Fig. 3. An example of the leaf development at different stages. Hexagonal numbers are determined according to the introduced rules. The color of the tile represents the corresponding hexagonal number with white corresponding to zero and black to some maximal value. The tiles where the hexagonal number exceeds the prescribed threshold are marked green. In this example, $T_{gl1} = 10$.



(a) (b) (c) (d) (e) (f)

Fig. 4. Final distribution of hexagonal numbers for different values of the threshold T_{gl1}. (a) $T_{gl1} = 1$; (b) $T_{gl1} \in [2, 3]$; (c) $T_{gl1} \in [4, 5]$; (d) $T_{gl1} \in [6, 7]$; (e) $T_{gl1} \in [8, 9]$; (f) $T_{gl1} \in [10, 13]$.

Analysis of Network Dynamics Including Hidden Variables by Symbolic-Numeric Approach

Daisuke Tominaga[1], Yasuhito Tokumoto[2], Hiroshi Yoshida[3], Masahiko Nakatsui[1], Sachiyo Aburatani[1], Fuyan Sun[1], Jun Miyake[2,4], and Katsuhisa Horimoto[1]**

[1] Computational Biology Research Center (CBRC),
National Institite of Advanced Industrial Science and Technology (AIST),
Tokyo 135-0064, Japan,
tominaga@cbrc.jp,{m.nakatsui,s.aburatani,f-sun,k.horimoto}@aist.go.jp
[2] The University of Tokyo, Tokyo 113-0033, Japan
{y.tokumoto@will.dpc,jmiyake@appchem.t}.u-tokyo.ac.jp
[3] Kyushu University, Fukuoka 812-8581 Japan,
phiroshi@math.kyushu-u.ac.jp
[4] Research Institute for Cell Engineering, National Institite of Advanced Industrial Science and Technology (AIST), Osaka 563-8577, Japan

Abstract. We propose a symbolic-numeric method for estimating the ratio of kinetic constants in a biological network including hidden variables which mean that the behaviors of corresponding molecules cannot be directly measured. In the present method, an algebraic manipulation of the differential equations over the Laplace domain, formulated based on the assumption of linear relationships between the variables, is combined with the numerical fitting of the sampling data. The performance of the method is illustared for a part of MAPK network with the data measured by the transfection cell array in combination of the gene interference by siRNAs.

Key words: Network Dynamics, Hidden Variable, Time-series Data, Laplace Transform, Linear Differential Equation, Bi-fan Structure

1 Introduction

The clarify of the dymanics of a complex network is one of the important issues in systems biology. By the recent advances of the experimental technology in molecular biology, the behaviors of a large numbers of genes such as gene expression levels can be measured simultaneously in different conditions. However, it is still difficult to measure the time series of gene expression levels in living cells. Indeed, the transfection cell array[1] is one of most advanced technology for measuring the time series of gene expressions in a living cell, but even by using these experiments, the gene expressions are measured for only a small number of repoter genes, in which the fluorescence protein is artificially encoded. In usual,

** Corresponding author.

it encounters frequently the difficulty for measuring the molecule behaviors in biological experiments, and for analyzing the network including hidden variables in the biological networks. Thus, it is challenging to clarify the dynamics of whole network only from the measurement of a small fraction of constituent molecules.

In this paper, we propose a symbolic-numeric approach for estimating kinetic constant in the case when the time series of expressions of reporter genes are measured by the transfection cell array in combination of the interference of the remaining genes by siRNAs[2]. In this case, the number of the reporter genes are limited, and thus time-dependent behaviors are not measured in most constituent genes. Here, by using our approach, we present a solution for estimating the network dynamics in a partial model including hidden variables of MAPK pathway.

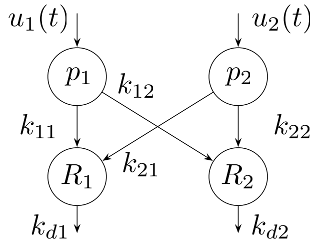


Fig. 1. Network model analyzed in the present study

2 Materials and methods

2.1 Model

We consider a network in Fig. 1 which is called 'bi-fan' structure[3]. In the network, we assume that the expression levels of two molecules, R_1 and R_2, can be measured by their reporter genes. These two molecules degrade with respective known constant rates, k_{d1} and k_{d2}. We also assume that any expression levels can not be measured in two molecules, p_1 and p_2, which change by unknown external forces, $u_1(t)$ and $u_2(t)$. The kinetic constants between them are k_{11}, k_{12}, k_{21}, and k_{22}.

2.2 Formulation over Laplace domain

The dynamics of the molecules in Fig. 1 is expressed by the following ordinary differential equations:

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t}R_1^0(t) &= k_{11}p_1(t) + k_{21}p_2(t) - k_{d1}R_1^0(t) \\ \frac{\mathrm{d}}{\mathrm{d}t}R_2^0(t) &= k_{12}p_1(t) + k_{22}p_2(t) - k_{d2}R_2^0(t) \\ \frac{\mathrm{d}}{\mathrm{d}t}R_1^{-p1}(t) &= k_{21}p_2(t) - k_{d1}R_1^{-p1}(t) \\ \frac{\mathrm{d}}{\mathrm{d}t}R_2^{-p1}(t) &= k_{22}p_2(t) - k_{d2}R_2^{-p1}(t) \\ \frac{\mathrm{d}}{\mathrm{d}t}R_1^{-p2}(t) &= k_{11}p_1(t) - k_{d1}R_1^{-p2}(t) \\ \frac{\mathrm{d}}{\mathrm{d}t}R_2^{-p2}(t) &= k_{12}p_1(t) - k_{d2}R_2^{-p2}(t) \end{cases} \tag{1}$$

where R_i^0 and R_i^{-X} indicate the expression levels when no genes are suppressed and that when gene X is surpressed by the corresponding siRNA, respectively.

Then, Eqns. (1) are also expressed as a system of the corresponding algebraic equations, by Laplace transformation, i.e.,

$$\begin{cases} sL[R_1^0(t)] - R_1^0(0) &= k_{11}L[p_1(t)] + k_{21}L[p_2(t)] - k_{d1}L[R_1^0(t)] \\ sL[R_2^0(t)] - R_2^0(0) &= k_{12}L[p_1(t)] + k_{22}L[p_2(t)] - k_{d2}L[R_2^0(t)] \\ sL[R_1^{-p1}(t)] - R_1^{-p1}(0) &= k_{21}L[p_2(t)] - k_{d1}L[R_1^{-p1}(t)] \\ sL[R_2^{-p1}(t)] - R_2^{-p1}(0) &= k_{22}L[p_2(t)] - k_{d2}L[R_2^{-p1}(t)] \\ sL[R_1^{-p2}(t)] - R_1^{-p2}(0) &= k_{11}L[p_1(t)] - k_{d1}L[R_1^{-p2}(t)] \\ sL[R_2^{-p2}(t)] - R_2^{-p2}(0) &= k_{12}L[p_1(t)] - k_{d2}L[R_2^{-p2}(t)] \end{cases} \tag{2}$$

where $L[R(t)]$ is function in s obtained by Laplace transformation of $R(t)$.

Apart from the network model, we fit the measured data of expression levels by exponential polynomials, i.e.,

$$R(t) = \sum_{i=1}^{n} a_i \exp(-m_i t). \tag{3}$$

Then, Eqn. (3) are expressed as a system of the corresponding algebraic equations by Laplace transformation, i.e.,

$$L[R(t)] = \sum_{i=1}^{n} \frac{a_i}{s + m_i}. \tag{4}$$

2.3 Estimation of kinetic constants over the Laplace domain

We eliminate $L[p_1(t)]$ and $L[p_2(t)]$ from the Eqns. (2), and we obtain the following equations:

$$\left\{\begin{array}{ll}
k_{d1} & = \dfrac{R_1^0(0) - R_1^{-p1}(0) - R_1^{-p2}(0)}{L[R_1^0(t)] - L[R_1^{-p1}(t)] - L[R_1^{-p2}(t)]} - s \\
k_{d2} & = \dfrac{R_2^0(0) - R_2^{-p1}(0) - R_2^{-p2}(0)}{L[R_2^0(t)] - L[R_2^{-p1}(t)] - L[R_2^{-p2}(t)]} - s \\
\dfrac{k_{12}}{k_{11}} & = \dfrac{(s + k_{d2})L[R_2^{-p2}(t)] - R_2^{-p2}(0)}{(s + k_{d1})L[R_1^{-p2}(t)] - R_1^{-p2}(0)} = \\
 & = \dfrac{(\frac{R_2^0(0)-R_2^{-p1}(0)-R_2^{-p2}(0)}{L[R_2^0(t)]-L[R_2^{-p1}(t)]-L[R_2^{-p2}(t)]})L[R_2^{-p2}(t)] - R_2^{-p2}(0)}{(\frac{R_1^0(0)-R_1^{-p1}(0)-R_1^{-p2}(0)}{L[R_1^0(t)]-L[R_1^{-p1}(t)]-L[R_1^{-p2}(t)]})L[R_1^{-p2}(t)] - R_1^{-p2}(0)} \\
\dfrac{k_{21}}{k_{22}} & = \dfrac{(s + k_{d1})L[R_1^{-p1}(t)] - R_1^{-p1}(0)}{(s + k_{d2})L[R_2^{-p1}(t)] - R_2^{-p1}(0)} = \\
 & = \dfrac{(\frac{R_1^0(0)-R_1^{-p1}(0)-R_1^{-p2}(0)}{L[R_1^0(t)]-L[R_1^{-p1}(t)]-L[R_1^{-p2}(t)]})L[R_1^{-p1}(t)] - R_1^{-p1}(0)}{(\frac{R_2^0(0)-R_2^{-p1}(0)-R_2^{-p2}(0)}{L[R_2^0(t)]-L[R_2^{-p1}(t)]-L[R_2^{-p2}(t)]})L[R_2^{-p1}(t)] - R_2^{-p1}(0)}.
\end{array}\right. \tag{5}$$

Note that the right sides of Eqns. (5) are composed of the terms related with the repoter genes. Thus, we substitute Eqn. (4) obtained by fitting of Eqn. (3) into Eqns.(5), and we obtaine the equations in the form as $c = F(s)/G(s)$, where $F(s)$ and $G(s)$ are polynomials in s, and c is a constant value.

In the actual case, however, the equation, $c = F(s)/G(s)$, does not always hold, due to the noise of data. Thus, we estimate c so as to minimize the following formula:

$$M(c) = \int_0^{u_{max}} (cG(s) - F(s))^2 \mathrm{d}s. \tag{6}$$

By solving $\frac{\partial M(c)}{\partial c} = 0$, we obtain the following equation:

$$c = \frac{\int_0^{u_{max}} G(s)F(s)\mathrm{d}s}{\int_0^{u_{max}} G(s)^2 \mathrm{d}s}. \tag{7}$$

The values of k_{d1} and k_{d2} are known as the constant values for each reporter gene, and the value of u_{max} is estimated so as to minimize the following equation:

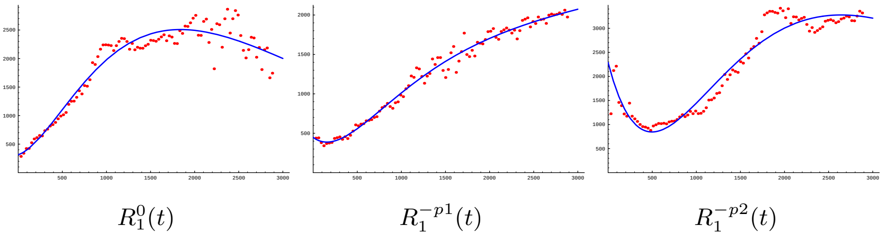
$$N(u_{max}) = (\frac{\int_0^{u_{max}} G_{kd1}(s)F_{kd1}(s)\mathrm{d}s}{\int_0^{u_{max}} G_{kd1}(s)^2 \mathrm{d}s} - k_{d1})^2 + (\frac{\int_0^{u_{max}} G_{kd2}(s)F_{kd2}(s)\mathrm{d}s}{\int_0^{u_{max}} G_{kd2}(s)^2 \mathrm{d}s} - k_{d2})^2. \tag{8}$$

By using the value of u_{max}, all constants, k_{d1}, k_{d2}, k_{12}/k_{11}, and k_{21}/k_{22} are estimated from Eqns. (7). Note that we should check the consistency between estimated and known values of k_{d1} and k_{d2}.

3 Results

We analyzed actual data measured by transfection cell arrays for a part of a network related with apoptosis in mouse[4]. In the actual network, the reporter genes are p53 and jun (R_1 and R_2 in Fig. 1), and are known to be associated with MAPK8 and MAPK14 (p_1 and p_2) by the same way as those in Fig. 1.

In this study, we set n=4 in Eqn. (3), and the examples of curve fitting to the actual data by the differential evolution algorithm which implemented as the *NMinimize* function in Mathematica 6 are shown in Fig. 2.



$R_1^0(t)$ $R_1^{-p1}(t)$ $R_1^{-p2}(t)$

Fig. 2. Examples of curce fitting to actual data

Table 1 shows the estimated values of k_{d1}, k_{d2}, k_{12}/k_{11}, and k_{21}/k_{22}, when the estimated value of u_{max} and ±10% values are used. Note that both values of k_{d1} and k_{d2} are given as 0.00192541. This value shows quite similar to the estimated k_{d1} and k_{d2}. This indicates that kinetic constants are successfully estimated in the present method.

Table 1. Estimation of kinetic constants

u_{max}	k_{d1}	k_{d2}	k_{12}/k_{11}	k_{21}/k_{22}
-10%	0.00182164	0.00250274	0.642616	2.45827
+ 0%	0.00211409	0.00208827	0.476623	3.16587
+10%	0.00237544	0.00182634	0.38048	3.82652

4 Discussion

Our appoach is summarized as follows: *i*) The relationship between the molecules in the analyzed network is modeled by a system of ordinary diffrential equations. *ii*) The time series data of the measurable molecules in the network are numerically fitted by a system of exponential polynomials. *iii*) The kenectic constant values and ratios of kinetic constants are expressed by fractions of fitted polynomials in s by symbolic (algebraic) computation. *iv*) Finally, ratios of kinetic constants are estimated employing the least square method for the known kinetic constants. The present approach will be applied to various issues on the biological networks including hidden variables.

In the present study, only the ratio of the kinetic constants is obtained. In near future, explict values of kinetic constants will be reduced by the symbolic-numeric approach in the similar way to the present study.

References

1. Ziauddin, J., Sabatini, D.M.: Microarray of cells expressing defined cDNAs. Nature 411, 107–110 (2001)
2. Aigner, A.: Applications of RNA interference: current state and prospects for siRNA-based strategies in vivo. Appl. Microbiol. Biotechnol. 76, 9-21 (2007)
3. Alon, U.: An Introduction to Systems Biology: Design Principles of Biological Circuits. Chapman & Hall/CRC (2006)
4. Aza-Blanc, P., Cooper, C.L., Wagner, K., Batalov, S., Deveraux, Q.L., Cooke, M.P.: Identification of Modulators of TRAIL-Induced Apoptosis via RNAi-Based Phenotypic Screening. J. Mol. Cell 12, 627-637 (2003)