View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Kyoto University Research Information Repository

KURENAI
Kyoto University Research Information Repository

KYOTO UNIVERSITY

Title	Kinase Computing : Concepts, Methods and Features (Algebraic Systems, Formal Languages and Conventional and Unconventional Computation Theory)
Author(s)	Liu, Jian-Qin; Shimohara, Katsunori
Citation	数理解析研究所講究録 (2004), 1366: 141-152
Issue Date	2004-04
URL	http://hdl.handle.net/2433/25371
Right	
Type	Departmental Bulletin Paper
Textversion	publisher

Kyoto University

数理解析研究所講究録 1366 巻 2004 年 141-152

Kinase Computing: Concepts, Methods and Features

Jian-Qin LIU and Katsunori SHIMOHARA
ATR Human Information Science Laboratories,
2-2-2 Hikaridai, "Keihanna Science City",
Kyoto, 619-0288, Japan

Extended Abstract

I. Introduction

As one of our efforts on exploring new unconventional computing paradigms, "Kinase Computing" is initiated as a new method of molecular computing with the biological faithfulness of potential implementation [1]. We expect kinase computing can provide us alternative ways to develop practical systems for the real world's applications. Its parallel mechanism of kinase computing can offer us an ability to explore new ideas of designing models, algorithms and software. When we study kinase computing in the view of the biologically inspired information processing paradigms, we feel that some important questions can not be avoided. For example, ***in the view of bio-molecular computers,*** some of them can be listed as:

(1). How to show the method is feasible for biological chemical implementation?
(2). How to demonstrate (1) by simulation?
(3). How to fulfill the computation tasks such as those by ALU, logic gates (AND, OR, NOT, XOR, etc.) and the abstract computation?
(4). Its computability?
(5). Its complexity?
(6). How to apply it into NP problem solving (e.g. 3-SAT)?
(7). How to obtain robustness;
(8). What is its cost in operations and features in biophysics?
(9). How to make the corresponding manipulations to be equipped with (self-) regulation/controllability and feasibility.

In the view of biologically inspired information processing paradigm, the core of the problem becomes what kinds of structure can increase the complexity degree within the algorithm process itself by self-regulation/controlling. Maybe this will lead to the efficiency improvement of programming implementation of the algorithms under the uncertain environment in order to achieve the adaptation mechanism. In our work, we consider the following structure as an example:

Object ::= pathway;
Operation ::= interaction;
(Data) Structure ::= hypergraph
Kinase(enzymes) and signaling pathways in cells in situ.

In the view of theoretical computer science and mathematics, we study on:

(1) Formal systems by graph rewriting (and reactive system theory);
(2) Formal languages derived from the unconventional computation models;
(3) Features related to primitives, words and (possible) codes, and some quantitative measures and limitations.

Of course we also consider the others aspects about the (potential) relationship with other disciplines, e.g.,
(1) graph rewriting in topology (GRiT) to supra-molecular structures based on self-assembly of DNAs in 2D and 3D;
(2) entanglement from the quantization form of the interactions to quantum information processing.
(3) the operations of "table * table" relating to whole memory device of nano-computers.
Besides these points, bio-informatics of genomes and proteoms also is a field, which we are working to apply kinase computing into.

II. Kinase Computing as Bio-molecular Computing

In a biological plausible sense, the whole molecular computing process consists of three biological chemical factors. They are:
(1) the mechanism of signal transduction of G protein (Rho family GTPases) in mammalian cells *in vivo* or *in situ*, which is the adaptive controller for the information flow within the whole molecular computer and works as the wet-CPU;
(2) the program codes(words) based on phorylation-dephorylation schemes in cells for efficiently encoding the molecular computer, where the word designs maybe is related to the programming manner;
(3) kinase that acts as a logic "switch" for the engine of the molecular computer, which cooperates with the photosphases to realize the reversible computing. Here the true values of {T, F} can be regarded to equal to the values of {1,0} under the context of specific application problems.
In order to describe the computation in a biochemical way, the main operations can be summarized as:
The bio-operations(ors) includes:
* The phosphorlation-dephosphorylation for encoding the objects as the binary logic values.
* The activation of the combinatorial forms of the objects coded by phosphorlation-dephosphorylation.
* The regulation of the directed flow of signaling molecules at the inter-and-intra cell communication mode and among the channels.
* The biochemical reaction of the signaling molecules within the domain of the corresponding pathways with kinases.
* The activation of the kinases and corresponding pathways
* The readout.
These are given in the meaning of the direct (potential) implementation of the molecular computers, where we are developing the self-regulation schemes for efficiently controlling the information flow with engineered phosphorylatio-dephosphorylation mechanism.
The high-level operations(ors) includes:
* DISTRIBUTE: to put the population -- the set of the candidates -- into the concerned cells.
* FEED: to push these candidates into the pathways.
* SIEVE: to select the valid candidates through the pathways -- the constraints of the computation.

* EXTRACT: to detect the candidates from different pathways according to the threshold for the common outputs -- to get the final result.

These are used to describe the relative macro-level's operations that are carried out by the bio-operations. The primitives such as "distribute", "feed", "sieve" and "extract" are helpful to optimize the word set and related wet-programming languages.

About the motivation of our work, we must consider seriously the limitation of the number of DNAs that is needed in the existing DNA computing for 3-SAT. The DNA computing has important progresses (Cf.[2~26]). As [14] point out that: "Recent progress on the SAT Problem with DNA computations ... For the SAT problem, the likely upper limit is 70 to 80 Boolean variables". The computation in kinase computing is carried out with the kinases and related functional proteins of cells. So that no that huge number of DNA is needed and kinase computing can beyond that limitation. An individual cell forms of kinase computing can solve the at least 2000 variables' 3-SAT problems. Also the multi-cells can be used for heterogeneous kinase computing for large scale's computation tasks. Beyond that limitation, we have worked out the schemes based on cells and signaling pathways. The first direct advantage derived consequently is the autonomy and self-regulation of phosphorylation-dephosphorylation of cells. The autonomous process of "kinase computers" is emphasized and can be provided in our method/model. In principle, the basic idea is that ATP is feed by sufficiently mechanisms in cells and it guarantee the circles of energy feeding of the whole "computation" we expect using the living cells. The ATP-based mechanism has make the well designed engineered phosphorylation-dephosphorylation for computing become feasible. This mechanism can make the cells working conditionally as molecular machines with efficiency, robustness and controllability. Others merits of kinase computing are the reliability, biological faithfulness, robustness (free of errors), scalability, efficiency (e.g., linear complexity cost in controlling space and time). These are enough to answer the challenges such as in scalability, reliability, efficiency and others from biomolecular computing. One of the features of kinase computing is the "ladders phenomena"[29].

The difference of kinase computing and other bio-molecular computing methods can be summarized as:

(1) The basic difference between kinase computing and DNA/RNA computing: The materials for bio-molecular computing are different. Kinase computing uses the kinases and kinase-regulated signaling pathways in cells, where the information is encoded by phosphorylation-dephosphorylation mechanism. DNA/RNA computing uses the DNAs/RNAs where the information is encoded by DNA/RNA's complementary [2~26]. It is obvious that the principles are totally different.

(2) The difference of kinase computing and other bio-molecular computing based on cells: In kinase computing, the interactions of pathways regulated by kinases and the corresponding functional proteins are constructed to carry out the computer in the view of the information flow. In the work by Laura F. Landweber and Lila Kari [22] and by Lila Kari and Laura Landweber [21], the gene and related operations (e.g., recombination) works for the computing. In kinase computing, graph rewriting is the main formal model for massively parallel computing. In membrane computing (P-system) [25], the string rewriting for membrane structures with multi-set representation is the main form. In the ciliates-based molecular computing method by Andrzej Ehrenfeucht et al[24], the gene assembly mechanism is used and corresponding operations are exerted on strings. In

kinase computing the Turing machine, ALU, problem solving and (basic) logic inference have been successfully fulfilled. In the amorphous computing [26], the logic circuits are made for communication. The physical mechanisms are totally(radically) different.
(3) Kinase computing is different from the surface-based DNA computing[4].
(4) Kinase computing is different from other bio-molecular computing, to the best of our knowledge.
(5) Kinase computing is also different from the other unconventional computing paradigms such as Evolutionary Computation (that includes Genetic Algorithms, Evolutionary Programming, Evolutionary Strategy and etc.), (existing contents of) Artificial Chemistry, Artificial Neural Networks and Multi-Agents.

III. Models of Kinase Computing

By the terms of rewriting, we design some examples to explain the features of kinase computing.
Example 1 (The kinase computing model based on string rewriting representation):
Let V be the alphabeta set, the set of terminals V_T = {a, b, c, d, e, f, g, h, u, v, p, q}, S1~S14 are the strings made by V, we define the following string rules for the pathways as:

(1) a -> S1 (pathway 1),
(2) S1 -> S2 b S3 (pathway 1),
(3) S2 -> c (pathway 1),
(4) S3 -> d (pathway 1),
(5) c -> S4 (pathway 2),
(6) d -> S5 (pathway 3),
(7) S4 e -> S4 p e (pathway 2),
(8) S4 p e f -> S4 e f (pathway 2),
(9) S4 e f q -> S4 (pathway 2),
(10) u S5 g -> u S5 p g (pathway 3),
(11) u S5 p g h -> u S5 g h (pathway 3),
(12) u S5 g h q -> u S5 (pathway 3),
(13) S4 -> u (pathway 2),
(14) v u S5 -> S5 (pathway 3),
(15) c (pathway 1) -> c (pathway 2) (pathway 4),
(16) d (pathway 1) -> d (pathway 2) (pathway 4),
(17) a (pathway 4) -> a (pathway 1) (pathway 4),

Here the pathway 4 refers to the whole pathways that includes the rules for pathways (1), (2), (3) and the intra-cell communication processes by rule (15)~(17). The biochemical reactions are described by rules (1)~(14). From this simplified abstract form, we still feel the hypergraph structure for the whole biochemical processes is necessary for the interactions of different string rewriting systems for different pathways.
If we replace (7) by the form: S4 e -> S4 e p, this will give out a CSL's way. Although the abstract forms are discussed, we have found these rules can be used to model certain kind of signaling pathways in cells in the logic way with coarse-grained forms. This is a good example to explain how peoples work out an unconventional computation model or computing system inspired by natural (e.g. biological) systems, which contribute much to

the theoretical computer science (e.g. parallel computing). In this sense, the significance of our models becomes obvious.
In order to discuss how signaling pathways do work for computing which is different from DNA computing, we give the algorithm as follows:

```
ALGORITHM (an example): m -- the number of clauses; n -- the number of variables; k -- the literals).
(in  a pseudo-programming-language, and by a serial implementation or simulation)
  For ( i  = 0 to m, i++)
      {  /* i */
      For ( t = 0 to Tm; t++);
        {  /* t */
        /* the consuming time in channels -- t */
        For ( j = 0 to n,  j++)
          {    /* j */
          For (p = 0 to 2^n, p++)
              {   /* p */
              Assignment (signaling-molecules[p]);
                          /* assigned with phosphorylation/dephosphoryolations
                    for values of {true, false}  */
               }  /*  p */
          selecting-neighborhood (variable-j, random-0, random-2,  ... ,
                                  random-(n-1));
          /* select the size of neighborhood as normal distribution,
             un-biased, this is for the well-stired global pool */
          sieving-among-local-neighborhood (j, η(t): random);
          /* selected those components/individuals fallen in to the
             neighborhood of radius = η(t) */
                η(t) : = η(t) + Δ η(t);
             reaction-diffusion (variable-j, η(t));
             if (saturation-degree == true)
                {goto: coupling;}
             reaction-of-candidates-with-ρ-units-of-pathways (i);
          }  /* j */
       }  /* t */
coupling: For (t' = 0 to T'm; t'++)
          {   /* t' */
          While (saturation-degree == true)
                do
                {
                 simultaneous-reactions-of-ρ-units (i);
                }
          }   /* t' */
}  /* i */
p" = 0;
For ( p' = 0 to 2^n, p'++)
     {  / * p' */
      detection ( candidates[p'] );
      output[p"] := candidates[p'];
     p"++;
     }  /* p' */
max-p" = p";
For (p" =0 to max-p", p"++);
     {  /* max-p" */
     read-out-by-2D-gel (output[p"]);
```

```
    } /* max-p" */

/* -- end -- */
```

Further we can observe the information flow through the pathways of our models in the following three examples:
Example 2 (Molecules passing through the pathways):
Let the VRs be Xa, Xb, and Xc, the weights of the pathways are Ca', Cb', and Cc', a graph automata with {HR, VR} will work on these as the process:

Xa, Xb, Xc --> Weights: Ca', Cb', Cc' --> Xa", Xb", Xc".

or

Xa" Xb" Xc" <-- Ca' Xa' | Xa . Cb' Xb'| Xb. Cc' Xc'| Xc
with random context

Here the number of the variable = the length of the supra-molecular candidates, they can be bounded. Let X0 X1 Xa X3 Xb X5 Xc X7 ($L=2^3$) and the computing unit has "limited" memory (of course it can be extended into an unlimited device such as those schemes in Turing machines). We can write { Ca' Xa, Cb' Xb', Ca' Xa' Cb' Xb', Ca' Xa' Cb' Xb' Cc' Xc' } and then we can get that

X0 X1 Ca' Xa X3 Xb X5 Xc X7

s.t. certain conditions

---> Ca' .

And we can set the rule as

< Te, Xa > | Xa => Te

w.r.t.

Xa, Xb, Xc.

Example 3 (Probabilistic filtering):
Let the input as X0 (0) , ... , Xn (0) , X0 (1), ... , Xn (1), X0 (L-1), ..., Xn (L-1) where $L = 2^n$. Input is {X0 (q), ... , Xn (q)} with values in {T, F}. The rules are:

```
Ca' Xa' | Xa,     accept Xi,  / reject:  else
Cb' Xb' | Xb        ...   Xj   /  ...
Cc'  Xc' | Xc       ...   Xk   /  ... .
```

Let Xa = Xi, Xb = Xj, Xc = Xk, $0 < i \neq j \neq k < L$.
We can get that:

Ca' Xa' | Xa,
Cb' Xb' | Xb,
Cc' Xc' | Xc,

Ca' Xa' (+) Cb' Xb' | Xa (+) Xb,
Ca' Xa' (+) Cc' Xc' | Xa (+) Xc,
Cb' Xb' (+) Cc' Xc' | Xb (+) Xc,

Ca' Xa' (+) Cb' Xb' | Xa (+) Xb (+) Xc,

...

Then,

X0 X1 Ca' Te X3 Xb X5 Xc X7

-->

* * Ca' Te * * * * *;

X0 X1 Ca' Te X3 Cb' Xb' ... .

Example 4 (Through the coupled pathways):
Assume that we have the different strings in different pathways as:

* * * Ca' Te * * * * *

* * * * * * Cb' Te * *

* * Ca' Te * Cb' Te * *

* * * * * * * * Cc' Te ...

... Ca' Te ... Cc' Te ...

... Cb' Te ... Cc' Te ...

* * Ca' Te * Cb' Te * Cc' Te * * *.

So the confluent features can be observed. The interactions of different pathways give out certain common predicate forms with same truth values remained, e.g. through a matrix that includes the acception element for {Ca, Cb, Te,Cc}. This can produces certain coupled phenomena according to the definition of artificial chemistry, also the templates/building blocks are natural in the processes mentioned above. Further, the recursively enumerate set would be generated and module design could be carried out by simulating the ρ-units in discrete (time)/symbolic (spatial representation) way. We are interested in generating subsets of McNaughton langauges by above model and related issues on regular languages.

Example 5 (Kinase computing model based on graph rewriting):
The model based on the graph rewriting explores the "data structure" of pathways and to define related operators by the logic form (e.g., predicate). This arrangement is helpful to establish a systematic theory with necessary axioms and rigorous logic characteristics.
Among the different equivalent classes $PATs_H$ of pathways, the transductions are exerted on the monographs:

$$G^M_H = (V, ctrl_{H,} \equiv_H),$$

where $\equiv_H$ is the equivalent relationship of $PATs_H$.

The candidates are arranged as the pool consists of the all the combinatorial forms of the binary logical values of all the variables Xi0 Xi1 ... Xi n (i belongs to {0, 1, .., 2^n}. These are input to the model represented by the bigraphs. The rule of the bigraph model is:
(a) Each unit is made to judge the compatibility of the logic situation of each variable, then to exert the HR and VR, to make the linkage to the output.
(b) For all the outputs, only those nodes satisfying all the units are kept and others are deleted with HR and VR.
In terms of the transductions, the rewriting process is represented as the logic forms based on the bigraphs. In the case of applying our model into the NP problem solving (e.g., the 3-SAT), we represent the hypergraph objects by bigraphs. The topographs are used to describe the input-output relations of the hypergraphs (the candidate solutions). The monographs are the connections between the input and the output. In a word, the rewriting acts as:

$$\text{Hygraghs } \Theta \text{ rewriting-rules} \Rightarrow \text{hypergraphs}$$

where the rewriting rules are constructed by pah (.) and HR and VR for pathways.
In the case of the 3-SAT, the hypergraphs are the candidate solutions. At the initial step, these are the following forms:

X1 ------> X1
X2 ------> X2
...

Xn ------> Xn

(not) X1 ------> (not) X1
(not) X2 -------> (not) X2
...
(not) Xn -------> (not) Xn.

The basic rules can be summarized as follows:
(1) The rule for the clause selection:
Delete (Xi) from the hypergraph, i.e., the bigraph,
and delete the related hyperedges, if the Xi is not the
Xj in the clause ($i \neq j$).
To applying this rule by the times of the number of the variables appeared in the clause.
(2) The rule for the interactions of the different clauses:
To delete those variables not appeared in all the hypergraphs.
This rule can be carried out in one step, owing to the parallelism of our model.
Here we explain the first macro-rule that is further realized by the micro-rules constructed by HR and VR (therefore that lead to the pathway replacement).
This rule is equivalent to the transduction operators in our model. This corresponds to the hypergraphs in graph rewriting systems.

$$F(G) = G' \text{ s.t. the condition/constraints of rewriting } \phi.$$

where ϕ refers to the requirement that the remained hypergraphs should satisfies the clauses in the 3-SAT. This implies that the three components of the 3-SAT clauses can be divided into three independent sub-pathways that correspond to the variables and select the complex going through them. E.g, the sub-pathway that corresponds to the variable X1, only allows the chemicals contain the signaling molecule that encodes X1 to attend the reactions. The logic form of the only allowing the hypergraph that contain X1 to remains, equals to the operations that delete the pah(.) which do not contain/infer X1 in the graph rewriting.

IV. The Relationship between the Formal Methods of Kinase Computing and Artificial Chemistry with Self-assembly

In order to study the inter-(proto)cell proto-communications with molecular operations, we consider the object of "nano-life" (e.g., nanobes, nano-organisms, nanoplankton, etc) where the self-assembly is necessary to guarantee the artificial chemistry system capable of emerged/evolved by *in silico* simulation for systems biology and bio-informatics (proteomics). The reversible phosphorylation which very important for most aspects of cells (ATP, metabolic and others). Rho family GTPases [27] play a key role in the skeleton of mammalian cells, which significance can be understood if we think about the brain-like systems in artificial life evolved for AI under necessary conditions. It is significant to emerge proto-cells in the community of artificial chemistry. The further step we expect is to study how to emerge out the proto-communications. Also in the condition of the "nano-organism", we need the self-assembly operations for constructing this kind of artificial life systems/processes. The internal maintenance of the proto-cells is modeled as automata with certain kinds of languages. These proto-cells are regarded as equivalence classes, with defined algebraic relation. The reactions (interactions) of the AlCh is constructed as the "interactions of the rewriting processes" of the different automata, which "rules" are the one that can derive/generate the classes of McNaughton languages and this makes the necessary condition for the emerging the proto-communications among the proto-cells with the condition of the probabilistic constraints among the neighborhoods. The symbolic objects as the "names' chemicals involving in the reactions by chemical evolutions and real valued objects as the "concentrations" are "coupled in the formal modeling.

For example: the main points may include the follows:

(a) Proto-cells: automata with languages (e.g., regular languages),

(b) The reactions among pro-cells: "pathway rewriting" by designed rules:

heterogeous pathways,
regular languages,
Q1,Q2,Q3,Q4 of graph rewriting (Cf.[30]),
this can give out the subsets of McNaughton langauges
so communications are available.

(c) The global concentration distribution has been affected by the reaction-diffusion, and the threshold make the refreshing of the molecules for rewriting in different proto-cells, in which "auto-rewriting systems" or "rewritable rewriting systems" may be embedded. The result of the symbolic communications of inter-cells and the two-dimensional intra-communications within proto-cells' pathways are expected.

V. The Relationship between Kinase Computing and Artificial Life

The concept of "self-reproduction" is important and basic for artificial life systems where the study is on the form of program. The artificial life system must be evolvable. So the evolvability (e.g. the concept in [28]) is important. Looking at the topics of "meta-evolutionary emergence of artificial life based on genomic dynamics", we try to discuss the concepts and then give an example for studying it. In the view of the engineering (e.g., software programs), the meta-evolution (we use this term in the noun form, "meta-evolutionary" in the adjective form) refers to the evolutionary process in the definition of artificial life can generate spontaneously out certain expected (nonlinear) phenomena without explicitly forces from the outsides. Of course, the interfacing with the external environment is necessary. The engineering way to define this more concretely is the phenotype (function) can be generated in a autonomous way from the artificial life system. For example, the proto-cells can be emerged (evolved out) in an artificial chemistry system. In the case of "meta-evolutioanry emergence", the term "generate" equals to the term "emerge". Here this concept mainly refers to the "non-parametric" generation of the phenotype (function) by evolutionary computing (models). This topics may be discussed in following main points:

(A): We consider a one-dimensional artificial chemistry system (denoted as 1dACs) with certain constraints, which is with the Turing computability. When several (e.g., two) 1dALs are being evolved, the interactions can be emerged. in spatial distribution (random in advance for initialization).

(B): Theoretically, this can generate the global proto-cells' communications under certain conditions.

(C): The complexity and efficiency.

(D): The features (merits) of parallel algorithms derived.

(E): Molecular computing system is a good way to study this.

(F): Artificial chemistry + molecular computing.

(G): Emergence owing to the complexity achieved from the evolution --
here we define our "meta-evolutionary emergence" in terms of signaling pathways.

(H): Self-assembly in nano-technology is a good object to study how the collective behavior emerged from the interactions of the "atom" (fundamental) pathways.

(I): The rules are based formal language and the running processes are set in random.

(J): Applications in bio-informatics (models, simulation and others):

The interactions of different pathways in the kinase computing paradigm is beneficial to construct the evolutionary systems with artificial chemistry operations, molecular computing architecture, and other new emerging methodologies.

Acknowledgement

This research was conducted as part of "Research on Human Communication"; with funding from the Telecommunications Advancement Organization of Japan.

The authors are sincerely thankful to Prof. Masami Ito for his advices and suggestions on theoretical computer science, Prof. Kozo Kaibuchi, Dr. Mutsuki Amano and Dr. Shinya Kuroda for their academic helps on signal transduction of molecular biology.

References

[1] Jian-Qin Liu and Katsunori Shimohara, Pathway graph models for molecular computing *in situ*, RIMS Kokyuroku 1222, RIMS, Kyoto University, July 2001; 128-137.
[2] L.M. Adleman, Molecular computation of solutions to combinatorial problems, Science, 266, 1994; 1021-1024.
[3] Winfree, E., Liu, F.R., Wenzler, L.A. & Seeman, N.C., Design and self-assembly of two-dimensional DNA crystals, Nature 394, 1998, pp.539-544.
[4] Qinghua Liu, Liman Wang, Anthony G. Frutos, Anne E. Condon, Robert M. Corn, & Lloyd M. Smith, DNA computing on surfaces, Nature, 13 January 2000, Volume 403, issue no. 6766, pp.175-179.
[5] Mitsunori Ogihara and Animesh Ray, DNA computing on a chip, Nature, 13 January 2000, Volume 403, issue no. 6766, pp.143-144.
[6] C. Mao, LaBean, T.H.Reif, J.H.Seeman, Logic Computation using algorithmic self-assembly of DNA triple-crossover molecules, Nature, vol.407, Sept.28 2000, pp.493-495.
[7] Yaakov Benenson, Tamar Paz-Elizur, Rivka Adar, Ehud Keinan, Zvi Livneh, Ehud Shapiro, Nature, 414, 430-434 (22 Nov 2001).
[8] David K. Gifford, On the path to computation with DNA, Science, 266, 1994; pp.993-994.
[9] Lipton, R.J., DNA solution of hard computational problems, Science, 268, 1995, pp.542-545.
[10] Guarnieri, F., Fliss, M. & Bancroft, C., Making DNA add, Science, 273, 1996, pp.220-223.
[11] Ouyang, Q., Kaplan, P.D., Liu, S. & Libchaber, A., DNA solution of the maximal clique problem, Science, 278, 1997, pp.446-449.
[12] Kensuku Sakamoto, Hidetaka Gouzu, Ken Komiya, Daisuke Kiga, Shigeyuki Yokoyama, Takashi Yokomori and Masami Hagiya, Molecular computation by DNA hairpin formation, Science, 288, 2000; 1223-1226.
[13] Ravinderjit S. Braich, Nickolas Chelyapov, Cliff Johnson, Paul W.K.Rothemund, Leonard Adleman, Solution of a 20-variable 3-SAT problem on a DNA computer, Science, 296, 14 March, 2002;499.
[14] John H. Reif, Successes and challenges, Science, vol.296, April 19, 2002, pp.478-479.
[15] John A. Rose and Russell J. Deaton, The fidelity of annealing-ligation: a theoretical analysis, DNA6 book, pp.231-246.
[16] Sergio Díaz, Juan Luis Esteban and Mitsunori Ogihara, A DNA-based random walk method for solving k-SAT, in Pre-proceedings of DNA6 -- Sixth International Meeting on DNA Based Computers, A. Condon and G. Rozenberg, editors, Leiden Center for Natural Computing, Leiden, The Netherlands, June 13-17, 2000, pp. 181-191.
[17] Richard J. Lipton, Eric B. Baum, Editors, DNA Based Computers, (Proceedings of a DIMACS workshop, April 4, 1995, Princeton University), DIMACS, Series in Discrete Mathematics and Theoretical Computer Science, Volume 27, 1996, American Mathematical Society.
[18] Leonard M. Adleman, On constructing a molecular computer, in [17]: pp.1-21.
[19] Eric B. Baum, A DNA associative memory potentially large than the brain, in [17]: pp. 23-28.
[20] Erik Winfree, David Gifford, Editors, Preliminary Proceedings, Fifth International Meeting on DNA Based Computers, June 14-15, 1999, Massachusetts Institute of Technology.
[21] Lila Kari and Laura F. Landweber, Computational power of gene rearrangement, in [17]: pp. 203-211.
[22] Laura F. Landweber, Lila Kari, The evolution of cellular computing: nature's solution to a computational problem, BioSystem 52, 1999, 3-13.
[23] Thomas Hinze, Uwe Hatnik, and Monika Sturm, An object oriented simulation of real occurring molecular biological processes for DNA computing and its experimental verification, N. Jonoska and N. C. Seeman (Eds.): DNA7, LNCS 2340, Springer-Verlag (Berlin, Heidelberg) pp.1-13, 2002.
[24] Andrzej Ehrenfeucht, Tero Harju, Ion Petre, and Grzegorz Rozenberg, Patterns of micronuclear genes in ciliates, N. Jonoska and N. C. Seeman (Eds.): DNA7, LNCS 2340, Springer-Verlag (Berlin, Heidelberg) pp. 279-289.
[25] Gheorghe Paun, From cells to computers: computing with membranes (P systems), BioSystems 59 (2001); 139-158.
[26] Ron Weiss and Thomas F. Knight, Jr., Engineered communications for microbial robotics, , in Pre-proceedings of DNA6 -- Sixth International Meeting on DNA Based Computers, A. Condon and G. Rozenberg, editors, Leiden Center for Natural Computing, Leiden, The Netherlands, June 13-17, 2000, pp.5-19.
[27] K. Kaibuchi, S. Kuroda and M. Amano, Regulation of the cytoskeleton and cell adhesion by the Rho family GTPases in mammalian cells, Annu.Rev.Biochem. 68, 1999; 459-486.

[28] Christina L. Burch and Lin Chao, Evolvability of an RNA virus is determined by its mutational neighbourhood, Nature, 10 August, 2000, Vol.406, No. 6796, pp.625-628.
[29] Jian-Qin Liu and Katsunori Shimohara, On ladders' complexity mechanism and quantitative regulation of scalable molecular computers by Rho family GTPases, Proceedings of 2002 2nd IEEE Conference on Nanotechnology (IEEE-NANO 2002), Washington D.C., August 26-28 2002, pp.329-332.
[30] Jian-Qin Liu and Katsunori Shimohara, Heteronomous kinase computing: a novel class of optimization algorithms inspired by molecular biology, RIMS Kokyuroku 1241, 2001, pp.1-8.

APPENDIX: A Note on Conditional Upper Limitation of Emerging Communications of Proto-cells in Artificial Chemistry Systems

Considering that

$$P_g = h P_l / Q_l$$

where Q is the size of the local neighborhood.
Provided that the effects of reaction-diffusion on different pathways obey the normal distribution (*condition*), we have that

Conjecture (Lower Bound):

When $P_l \geq \min (p_Q / \lambda^Q_{max})$, the measure $P_g \neq 0$,

where p_Q is the probability of transition from current state to the next state of the individuals of proto-cells within the local neighborhood, and λ^Q_{max} is the maximum value λ of reaction (coefficient) matrix of related signaling pathways within the local neighbborhood/windows.
Currently our work is more focussed on the simulation, and wish this conjecture could be studied further.

Example 1:
Let the initial 3 × 3 state matrix represent the state array of the 9 spatially distributed proto-cells as

0.06 0.12 0.12
0.18 0.25 0.31
0.37 0.37 0.43

The proto-cell communications may occur as he situation that that certain symbol, e.g., certain symbols located in (0,3) and (1,1) in generation (t) can be emerged at other locations in the generation (t+l).