

A Process Calculus for Molecular Interaction Maps

Roberto Barbuti	Andrea Maggiolo–Schettini	Paolo Milazzo
Dipartimento di Informatica Università di Pisa, Italy barbuti@di.unipi.it	Dipartimento di Informatica Università di Pisa, Italy maggiolo@di.unipi.it	Dipartimento di Informatica Università di Pisa, Italy milazzo@di.unipi.it
Giovanni Pardini	Aureliano Rama	
Dipartimento di Informatica Università di Pisa, Italy pardinig@di.unipi.it	Dipartimento di Informatica Università di Pisa, Italy rama@di.unipi.it	

We present the *MIM calculus*, a modeling formalism with a strong biological basis, which provides biologically-meaningful operators for representing the interaction capabilities of molecular species. The operators of the calculus are inspired by the reaction symbols used in Molecular Interaction Maps (MIMs), a diagrammatic notation used by biologists. Models of the calculus can be easily derived from MIM diagrams, for which an unambiguous and executable interpretation is thus obtained. We give a formal definition of the syntax and semantics of the MIM calculus, and we study properties of the formalism. A case study is also presented to show the use of the calculus for modeling biomolecular networks.

1 Introduction

The use of formal methods in Systems Biology provides important advantages in the description and analysis of biological systems, since the structure and behaviour of biological systems can be described unambiguously and different analysis techniques can be applied to their study. In this field, the most influential approach has been proposed by Regev, Shapiro and others in [26, 27, 22, 23], where the π -calculus process algebra [19] is used to formalize biomolecular processes. Afterwards, many other formalisms originally developed by computer scientists to model systems of interacting components have been applied to Biology [25, 12, 22, 28], and extended to allow more precise descriptions of the biological behaviors. Other formalisms have also been developed expressly for being used in Biology [3, 1, 2, 7, 8, 9, 10, 18, 13, 21, 24, 6, 14].

Biologists have introduced graphical languages for describing bioregulatory networks. As an example, we quote *Molecular Interaction Maps* (MIM) [17]. MIM diagrams are composed of nodes, representing molecular species, and edges connecting nodes, which represent the possible reactions among species. Edges can express different kinds of reactions, according to the used reaction symbol. In this paper, we present a formalism which can be used for modeling and analyzing biological processes, called *MIM Calculus* (MIMc), which focusses on modeling the interaction capabilities of the involved elements. MIMc is defined in the style of *process calculi*, where each molecule appearing in the system is described by a term. However, unlike most of the previously proposed calculi for describing biological processes, which model reactions by means of process communication, MIMc provides high-level operators with a direct biological meaning. For example, there are operators for expressing the creation of a bond between two compounds (such as a *complexation*), and other biologically interesting events.

The calculus has a strong relationship with Molecular Interaction Maps. The presented approach has a twofold advantage. On one side, we can exploit the features of process calculi such as incremental

definition of models, techniques for analysis and verification of properties, and easy development of simulators. On the other side, the correspondence of the operators of the calculus with biological interactions allows an immediate translation of Molecular Interaction Maps into MIMc. Less immediate translations of Molecular Interaction Maps into more general formalisms can be found in [4, 5, 11]. Remark that the aim of the paper is to propose a calculus whose operators have a direct correspondence with the ones of Molecular Interaction Maps. Thus all the MIM operators are considered, without any investigation about a minimal set of them able to encode all the others.

The paper is structured as follows. After recalling the Molecular Interaction Maps, in section 2 we introduce the MIM calculus. In section 3 we study the relationship between the MIM calculus and Molecular Interaction Maps, and we establish conditions under which a term of the calculus is a formal representation of a MIM diagram. In section 4 we show an example of modeling with the MIM calculus, and in section 5 we draw some conclusions.

1.1 Molecular Interaction Maps

MIM diagrams provide a static view of the molecular species in a system, and their possible interactions. Interactions are represented by lines connecting nodes representing species, and the meaning of each interaction depends on the symbol used to draw the line. Each molecular species can appear only once. Moreover, since the diagram is static, it does not contain any information about number of molecules (concentration) of the molecular species.

Three classes of molecular species can be represented: *elementary species* (fig. 2a), *complex species* (fig. 2b) and *DNA sites* (fig. 2c). Complex species represent either a combination of elementary species or a modified elementary species. Figure 1 shows a simple MIM diagram, containing the elementary species A and B which can interact. A named elementary species is drawn as a rounded box, containing its name. A complex molecular species, resulting from an interaction, is depicted as a bullet on the corresponding interaction line. For instance, in Figure 1, the complex species obtained by the binding of A and B is represented by the node x on the interaction line.

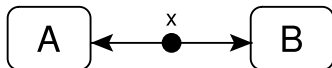


Figure 1: An example of MIM diagram.

MIM diagrams allow representing two kinds of interactions: *reactions*, which act on molecular species, and *contingencies*, which act on reactions or other contingencies. An interaction symbol represents a possible interaction that can happen if certain state conditions hold. Interactions can have a kinetic constant k associated with them, that is used to model its “occurrence” rate. Conceptually, a higher kinetic constant means that the interaction is more likely to happen than an interaction with a lower kinetic constant.

For defining MIM calculus we consider the reaction symbols shown in Figure 3. Note that they are only a subset of the reaction symbols available for use in a MIM diagram.

- *Non-covalent binding* (Figure 3a): denotes the reversible binding of the two pointed species: a molecule of the first species can bind to a molecule of the second species, forming a compound. Two species joined in a non-covalent binding can eventually dissociate again.
- *Covalent modification* (Figure 3b): denotes the covalent modification of the pointed species; the modification type (such as phosphorylation or acetylation) is written at the tail.

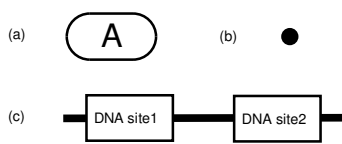


Figure 2: Species in MIMs.

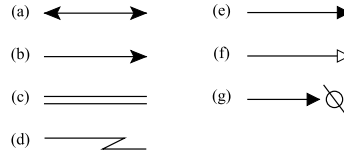


Figure 3: Reaction symbols.

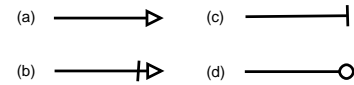


Figure 4: Contingency symbols.

- *Covalent binding* (Figure 3c): denotes a covalent bond of the two connected species.
- *Cleavage of a covalent bond* (Figure 3d): denotes the possibility of a covalent bond at the head (right end) to be broken by the presence of the species at the tail (left end). This symbol points from a species to a reaction symbol representing covalent binding.
- *Stoichiometric conversion* (Figure 3e): denotes the conversion of the species at the tail of the arrow, called reactant, into a corresponding number of product species, i.e. the species written at the tail of the arrow disappears, while the pointed ones appear.
- *Lossless production* (Figure 3f): it is similar to the stoichiometric conversion, but without loss of the reacting species.
- *Degradation* (Figure 3g): means that molecules of the species can disappear.

The following contingency symbols, shown in Figure 4, are provided by MIM diagrams:

- *Stimulation* (Figure 4a): means that the molecule of the species at left end stimulates the pointed reaction;
- *Requirement* (Figure 4b): means that the molecule of the species at left end is required in order for the pointed reaction to happen;
- *Inhibition* (Figure 4c): the presence of the species at the tail (left end) inhibits the possibility for the pointed interaction to happen;
- *Catalysis* (Figure 4d): means that the pointed reaction have a much higher reaction rate if the species is present than if it is not.

Interpretation of MIM diagrams There are three different interpretations for MIM diagrams [17, 16]: *explicit*, *combinatorial* and *heuristic*. Each interpretation is suited to a different purpose, depending on the application. They differ in how interactions between indirectly connected species are considered. Figure 5 shows a small example of MIM, which explicitly shows the bindings between A and B, yielding A:B; the binding between B and C, yielding B:C; and the possible phosphorylation of B, yielding pB. Some questions arise, such as if a complex between A:B and C, yielding (A:B):C, can form. Or, similarly, if pB and A can bind. The three different interpretations address this issue, by stating which interactions are possible.

A MIM diagram, in its explicit interpretation, depicts each possible reaction: an interaction line applies only to the molecular species directly connected to it. In this interpretation the order of bindings can be easily extracted from the diagram. In Figure 5, the explicit interpretation only allows the formation of A:B, B:C and pB. Explicit maps can be built using only a subset of MIM symbols: all contingencies symbols may, on the whole, be represented by a set of reaction symbols ([17]). Explicit maps without contingencies can be readily used for computer simulation.

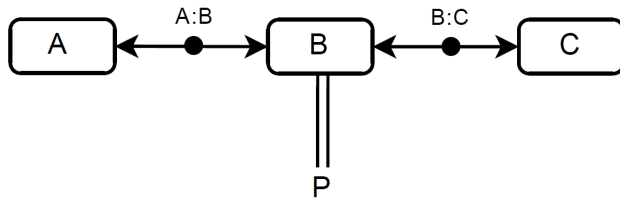


Figure 5: An example of MIM diagram.

Besides the complexes that are allowed by the explicit interpretation, in the combinatorial interpretation of MIM diagrams, an interaction line represents an implicit set of complexes and, hence, of reactions. In particular, each interaction line represents all those reactions between the interacting species, in each possible combination of their binding and modification states. In Figure 5, the combinatorial interpretation always allows A and B to bind, regardless of the fact that B is free, bound to C, or phosphorylated. This “transitivity” means that an interaction symbol applies indirectly to species through other interaction symbols. A main advantage of the combinatorial interpretation is this ability to synthesize with a few symbols a large number of possible complexes and reactions, making MIM a compact notation.

Finally, like the combinatorial interpretation, the heuristic one allows all the complexes that are permitted by the explicit interpretation, with the difference that it does not specify whether each of the combinatorial possibilities may or may not occur, either because of lack of knowledge or because some contingency symbols have been omitted to avoid overcrowding the diagram. Thus, heuristic MIM diagrams are used to depict only what is known, leaving unspecified what still has to be discovered.

2 MIM Calculus

In this section we formally introduce the syntax and semantics of the MIM calculus. MIM calculus is defined in the style of process calculi, where an agent represents a molecule of a certain named species. Names A, B, C, \dots are used to identify the different elementary species, and we denote by \mathcal{E} the set of names of elementary species. We also assume a set \mathcal{E}_c whose elements denote types of covalent modifications (such as phosphorylations).

Definition 2.1 (Syntax). *Processes P , named species S and capabilities μ of the MIM calculus are defined by the following grammar:*

$$\begin{array}{ll}
 P ::= \mathbf{0} \mid S \mid P \mid P & \gamma ::= (v, \iota) \xrightarrow{N} \mu \quad (\text{non-covalent binding}) \\
 S ::= \mu . IS & \mid (v, \iota) \xrightarrow{N} \mu \quad (\text{covalent binding}) \\
 IS ::= A \mid S : S \mid \overline{qS} \mid \overline{SS} & \mid (v, \iota) \xrightarrow{q} \mu \quad (\text{covalent modification}) \\
 \mu ::= \text{rec } x . \mu \mid M \mid x & \mid (v, \iota) \xrightarrow{N} \quad (\text{cleavage}) \\
 M ::= \emptyset \mid M + M \mid \gamma & \mid (v, \iota) \longrightarrow P \quad (\text{conversion}) \\
 & \mid (v, \iota) \longrightarrow P \quad (\text{lossless production})
 \end{array}$$

where $\mathbf{0}$ is the empty process, $A \in \mathcal{E}$ denotes an elementary species name, $q \in \mathcal{E}_c$ denotes the type of modification, $x \in \mathcal{X}$ is a variable, and N, v, ι denote species names, which are elements of the set \mathcal{N} of terms S without capabilities. For the sake of legibility we shall often use round brackets and we shall systematically enclose capabilities in curly brackets.

Terms P of the calculus are made of a composition of molecules S , by means of the parallel operator $- | -$. Each molecule is of the form $\mu.IS$, where IS describes the structure of the molecule, and μ describes its interaction capabilities. In particular, IS denotes either an elementary molecule of species A , or a compound molecule. In the case of compound molecules, IS is made of the single molecules forming the compound, combined by means of different syntactical operators specifying the kind of bond that keeps the molecules together: a *non-covalent bond* $S_1 : S_2$ between the species S_1 and S_2 , a *covalent modification* \overline{qS} of species S , or a *covalent bond* $\overline{S_1 S_2}$ between S_1 and S_2 . Note that the capabilities of each molecule forming a compound are retained in the compound description.

For example, term $\{\gamma\}.A$ models a molecule of species A , having a single interaction capability γ . A complex formed of two simple molecules A and B can instead be represented as $\mu_1.(\mu_2.A : \mu_3.B)$, where μ_1 are the capabilities of the compound, and μ_2, μ_3 are the capabilities of molecules A and B , respectively.

We denote the set of species by \mathcal{S} , and identify $\mathcal{N} \subset \mathcal{S}$ as its subset of named species without capabilities, i.e. where each μ is empty ($\mu = \emptyset$). We assume a function $[\cdot] : \mathcal{S} \rightarrow \mathcal{N}$ that strips all the capabilities from a named species $S \in \mathcal{S}$. For example, $[\mu_1.(\mu_2.A : \mu_3.B)] = \emptyset.(\emptyset.A : \emptyset.B)$. Moreover, we often avoid writing empty capabilities when no ambiguities arise, therefore we simply write $A : B$ instead of $\emptyset.(\emptyset.A : \emptyset.B)$. This function is extended to processes $[\cdot] : \mathcal{P} \rightarrow \mathbf{P}(\mathcal{N})$ as $[S_1 | \dots | S_n] = [S_1] \cup \dots \cup [S_n]$.

The calculus allows expressing different capabilities for molecules. Operator $\xrightarrow{N}\mu$ means that a species can form a non-covalent bond with a molecule of species name N . The result will be a compound of the form $- : -$ made of the two involved species, and with capabilities μ . Similarly, operator $\xrightarrow{N}\mu$ means that a species can form a covalent bond with a N , resulting in a compound of the form $\overline{S_1 S_2}$ with capabilities μ . The operator for covalent modification $\xrightarrow{q}\mu$, similarly produces a compound \overline{qS} with capabilities μ . The operator for cleavage \xrightarrow{N} means that a molecule can break the covalent bond specified by N , where N has to be either of the form of $\overline{N_1 N_2}$ or \overline{qN} . Finally, there are the operators $\longrightarrow P$, for expressing a conversion of a molecule into other molecules, and $\longrightarrow P$ for a lossless production of molecules. In both cases, the resulting molecules are represented by a process P .

We allow recursive definitions of capabilities, by means of the *recursion* operator rec . As always, $rec x.\mu$ binds the free occurrences of the variable name x in μ . We assume a *substitution* function $\mu[\mu'/x]$ for replacing each free occurrence of x in μ with μ' . The substitution function is also extended to processes. We use the notation $rec \tilde{x}.\mu$ with $\tilde{x} = x_1, \dots, x_n \in Var^*$ as an abbreviation for $rec x_1 \dots rec x_n.\mu$.

Names ν, ι are used to express contingencies on the application of an operator, depending on the species appearing in the environment. The former, ν , expresses the species that *must* be present (*promoters*), while the latter, ι , expresses those that must be *absent* (*inhibitors*). We omit writing contingencies when they are empty.

To give an example of a term in which recursive capabilities are used, let us consider a system in which substrate A is transformed into product C by the enzyme E . The MIM diagram in Figure 6 shows that enzyme E binds to A and the complex $E : A$ is subsequently transformed into C and E , thus recreating the enzyme. The enzyme E can be modeled in MIMc by the following term: $rec x.\{\xrightarrow{A}\{\longrightarrow(x.E | \emptyset.C)\}\}.E$.

Definition 2.2 (Structural congruence). The congruence relations \equiv_x on the syntactical categories $x \in \{P, S, IS, \mu, M, \gamma\}$ of the calculus are the least equivalence relations closed under syntactical operators and such that the following laws hold:

1. $P_1 | P_2 \equiv_P P_2 | P_1, \quad P_1 | (P_2 | P_3) \equiv_P (P_1 | P_2) | P_3, \quad P | \mathbf{0} \equiv_P P;$

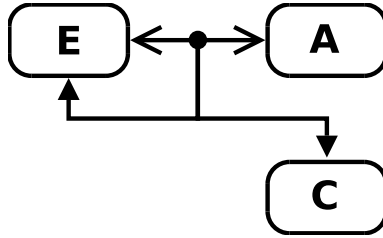


Figure 6: An example of recursive MIM

2. $S_1 : S_2 \equiv_{IS} S_2 : S_1$, $\overline{S_1 S_2} \equiv_{IS} \overline{S_2 S_1}$;
3. $M_1 + M_2 \equiv_M M_2 + M_1$, $M_1 + (M_2 + M_3) \equiv_M (M_1 + M_2) + M_3$, $M + \emptyset \equiv_M M$, $M + M \equiv_M M$;
4. (α -conversion) $\mu_1 \equiv_\mu \mu_2$ if they differ only on bound names;
5. $rec\ x.\mu \equiv_\mu \mu[rec\ x.\mu/x]$.

We omit the indication of x in \equiv_x when no ambiguities arise.

We propose now a reduction semantics for the MIM calculus, given in terms of a Labelled Transition System (LTS) representing the possible evolutions of a term. The labels of the LTS are actions identifying the reactions that each single transition models. All the possible actions Act are of the following forms: (a) $N_1 \leftrightarrow N_2$, (b) $N_1 \leftrightarrow\!\!\!\leftrightarrow N_2$, (c) $N \longrightarrow \{N_1, \dots, N_k\}$, (d) $N \longrightarrow\!\!\!\!> \{N_1, \dots, N_k\}$, (e) $N_1 = N_2$, (f) $N \rightsquigarrow \overline{N_1 N_2}$, (g) $q \Rightarrow N$, (h) $N \rightsquigarrow q\overline{N_1}$, which respectively represent (a) the creation of a non-covalent bond, (b) the breakage of a non-covalent bond, (c) a conversion, (d) a lossless production, (e) the creation of a covalent bond, (f) the breakage of a covalent bond, (g) a covalent modification, (h) the removal of a covalent modification.

Definition 2.3 (Reduction semantics). The *reduction semantics of MIM calculus* is the relation $\xrightarrow{\alpha}$ on processes such that

$$P \xrightarrow{\alpha} P' \quad \text{iff} \quad \exists \iota \in \mathcal{N}. P \xrightarrow{(\emptyset, \iota) \alpha} P' \quad (1)$$

where $\alpha \in Act$ is an action that represents the capability of P used for the reduction step, and $\xrightarrow{(\nu, \iota) \alpha}$, with $\nu, \iota \in \mathcal{N}$, is the least relation on processes, closed under structural congruence \equiv_P , and satisfying the following inference rules:

$$\frac{\mu_1 = \{X + (\nu, \iota) \xrightarrow{[S_2]} \mu\} \quad \alpha = [S_1] \leftrightarrow [S_2] \quad [S_1], [S_2] \notin \iota}{\mu_1.S_1 \mid \mu_2.S_2 \xrightarrow{(\nu, \iota) \alpha} \mu.(\mu_1.S_1 : \mu_2.S_2)} \quad (2)$$

$$\frac{\alpha = [S_1] \leftrightarrow\!\!\!\leftrightarrow [S_2]}{\mu.(S_1 : S_2) \xrightarrow{(\emptyset, \emptyset) \alpha} S_1 \mid S_2} \quad (3)$$

$$\frac{\mu = \{X + (\nu, \iota) \longrightarrow P\} \quad \alpha = [S] \longrightarrow [P] \quad [S] \notin \iota}{\mu.S \xrightarrow{(\nu, \iota) \alpha} P} \quad (4)$$

$$\frac{\mu = \{X + (\nu, \iota) \longrightarrow\!\!\!\!> P\} \quad \alpha = [S] \longrightarrow\!\!\!\!> [P] \quad [S] \notin \iota}{\mu.S \xrightarrow{(\nu, \iota) \alpha} \mu.S \mid P} \quad (5)$$

$$\frac{\mu_1 = \{X + (\nu, \iota) \xrightarrow{[S_2]} \mu\} \quad \alpha = [S_1] = [S_2] \quad [S_1], [S_2] \notin \iota}{\mu_1.S_1 \mid \mu_2.S_2 \xrightarrow{(\nu, \iota) \alpha} \mu.(\overline{\mu_1.S_1})(\mu_2.S_2)} \quad (6)$$

$$\frac{\mu = \{X + (v, \iota) \xrightarrow{[\overline{S_1 S_2}]}\} \quad \alpha = [S] \xrightarrow{[\overline{S_1 S_2}]} \quad [S], [\overline{S_1 S_2}] \notin \iota}{\mu.S \mid \mu'.\overline{S_1 S_2} \xrightarrow{(v, \iota) \alpha} \mu.S \mid S_1 \mid S_2} \quad (7)$$

$$\frac{\mu_1 = \{X + (v, \iota) \xrightarrow{q} \mu\} \quad \alpha = q \Rightarrow [S_1] \quad [S_1] \notin \iota}{\mu_1.S_1 \xrightarrow{(v, \iota) \alpha} \mu.q(\mu_1.S_1)} \quad (8)$$

$$\frac{\mu = \{X + (v, \iota) \xrightarrow{[\overline{qS_1}]}\} \quad \alpha = [S] \xrightarrow{[\overline{qS_1}]} \quad [S], [\overline{qS_1}] \notin \iota}{\mu.S \mid \mu'.\overline{qS_1} \xrightarrow{(v, \iota) \alpha} \mu.S \mid S_1} \quad (9)$$

$$\frac{P \xrightarrow{(v, \iota) \alpha} P' \quad [Q] \cap \iota = \emptyset \quad v' = v \setminus [Q]}{P \mid Q \xrightarrow{(v', \iota) \alpha} P' \mid Q} \quad (10)$$

Rule 2 deals with the creation of a non-covalent binding between molecules $\mu_1.S_1$ and $\mu_2.S_2$, thus giving rise to a complex $\mu.(\mu_1.S_1 : \mu_2.S_2)$. Note that the rule requires that molecule $\mu_1.S_1$ has the capability of binding with a molecule with name $[S_2]$ (the symmetric capability is not required for $\mu_2.S_2$). Rule 3 deals with the break of a non-covalent binding. There are no conditions for the breaking. Rule 4 deals with the conversion of a molecule $\mu.S$ into a number of other molecules when $\mu.S$ has the proper capability. Rule 5 deals with the lossless production, namely with the case in which $\mu.S$ produces a number of molecules without disappearing. Rules 6 and 7 are the analogs of rules 2 and 3 for the case of covalent binding. The unbinding, expressed by rule 7, requires the presence of a molecule $\mu.S$ having the capability of breaking the bond. Rules 8 and 9 deal with molecule covalent modification of a type q and with the removal of the modification, respectively.

All the rules are applicable only if the molecules involved are not inhibitors of the transformation itself. Rule 10 is used to apply a step of the reduction to the parallel composition of processes. The conditions of the rule ensure that the step is not forbidden by any of the molecules present in the composition. Moreover, these conditions, together with the fact that we need that the set v of promotors is empty to actually do the reduction step (defined in 1), ensure that all the promotors of the capability used for that step are present in the parallel composition of processes. As usual, we define \rightarrow^* as the reflexive and transitive closure of relation $\xrightarrow{\alpha}$.

We now define *contexts*, which represent terms with a hole, denoted as \square . The hole corresponds to a capability of a molecule, and therefore the hole occurs in the position of a capability μ . Conceptually, contexts allow identifying the position inside a term in which a molecule of a certain species appears.

Definition 2.4 (Context). Contexts of MIM calculus are defined by the following grammar:

$$\begin{array}{ll} C ::= P \mid S_c & \gamma_c ::= (v, \iota) \xrightarrow{N} \mu_c \quad (\text{non-covalent binding}) \\ S_c ::= \mu_c.IS \mid \mu.IS_c & \mid (v, \iota) \xrightarrow{N} \mu_c \quad (\text{covalent binding}) \\ IS_c ::= S : S_c \mid \overline{qS_c} \mid \overline{SS_c} & \mid (v, \iota) \xrightarrow{q} \mu_c \quad (\text{covalent modification}) \\ \mu_c ::= \square \mid M + \gamma_c & \mid (v, \iota) \longrightarrow C \quad (\text{conversion}) \\ & \mid (v, \iota) \longrightarrow C \quad (\text{lossless production}) \end{array}$$

The syntax ensures that exactly one hole is present in a context. Given S_c , the hole can occur either in the capabilities of S_c itself, when $S_c = \mu_c.IS$, or in the capabilities of one of the molecules of which

the molecule is composed, $S_c = \mu.IS_c$. Given a capability with a hole μ_c , the hole can be either the capability itself \square , or it can occur in one of the capabilities appearing in μ_c . In particular, if $\mu_c = \gamma_c$ (a basic capability) the hole can occur inside the capabilities of the species γ_c allows producing. Given a context C , its hole can be substituted with a capability μ , giving a process denoted $C[\mu]$.

For example, the context $C_1 = \mu_1.(\mu_2.A : \square.B)$ represents a molecule complex $A : B$ in which the hole refers to the capability of B forming the complex. Context C_1 can be applied to a capability μ_3 obtaining $C_1[\mu_3] = \mu_1.(\mu_2.A : \mu_3.B)$. Note that in this case, the hole is relative to a species named B , and this is clearly visible from the syntax of the context. However, in other cases, the name of the species relative to a hole is not directly present in the syntax of the context. For example, the hole in context $C_2 = \{\overset{B}{\dashv}\square + \dashv P\}.A$ is relative to the species obtained as a complexation between A and B , whose name is $A : B$, which is not directly present in the syntax of the context. In order to extract, from a given context, the name of the species relative to the hole, we use a function $name : \mathcal{C} \rightarrow \mathcal{N}$ defined as follows.

Definition 2.5. Function $name$, from contexts to molecular names \mathcal{N} , and $name'(\mu_c, N) = N$, from a context μ_c and a name \mathcal{N} to a name \mathcal{N} , are recursively defined as follows:

$$\begin{aligned}
 name'(\square, N) &= N & (17) \\
 name(P \mid S_c) &= name(S_c) & (11) \\
 name(\mu.IS_c) &= name(IS_c) & (12) \\
 name(S : S_c) &= name(S_c) & (13) \\
 name(\overline{qS_c}) &= name(S_c) & (14) \\
 name(\overline{SS_c}) &= name(S_c) & (15) \\
 name(\mu_c.IS) &= name'(\mu_c, [IS]) & (16) \\
 name'((\nu, \iota) \xrightarrow{N'} \mu_c, N) &= name'(\mu_c, N : N') & (18) \\
 name'((\nu, \iota) \xrightarrow{N} \mu_c, N) &= name'(\mu_c, \overline{NN'}) & (19) \\
 name'((\nu, \iota) \xrightarrow{q} \mu_c, N) &= name'(\mu_c, \overline{qN}) & (20) \\
 name'((\nu, \iota) \dashv C, N) &= name(C) & (21) \\
 name'((\nu, \iota) \dashv C, N) &= name(C) & (22) \\
 name'(M + \gamma_c, N) &= name(\gamma_c, N) & (23)
 \end{aligned}$$

Definition of function $name$ is given by two mutually recursive functions $name$ and $name'$. In particular, $name'$ takes two parameters, a capability context μ_c and a name N , where N is the name of the species with which this capability is associated. The function $name'$ is used in equation 16, where extracting the name, relative to the hole, from a context $\mu_c.IS$ is reduced to extracting the name from μ_c , knowing that the capability μ_c is relative to a species named $[IS]$.

Example Consider an example of a MIM process which represents a molecular system described by the MIM diagram shown in Figure 7.

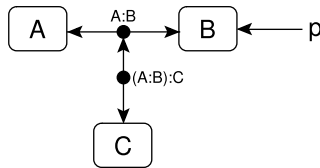


Figure 7: A MIM diagram.

A MIM process, differently from a MIM diagram, represents both the possible interactions among the species and the number of molecules that are present in the system. The following MIM process corresponds to a system with species A , B and C , with the interaction capabilities described by the

diagram in Figure 7, and in which there are two molecules of A , two of B and one of C :

$$\begin{aligned}
 P = & \{ \xrightarrow{B} \{ \xrightarrow{c} \emptyset \} \} . A \mid \\
 & \{ \xrightarrow{B} \{ \xrightarrow{c} \emptyset \} \} . A \mid \\
 & \{ \xrightarrow{A} \{ \xrightarrow{c} \emptyset \} + \xrightarrow{P} \emptyset \} . B \mid \\
 & \{ \xrightarrow{A} \{ \xrightarrow{c} \emptyset \} + \xrightarrow{P} \emptyset \} . B \mid \\
 & \{ \xrightarrow{A:B} \emptyset \} . C
 \end{aligned}$$

In the process P , the species A can complex with B , producing a molecule able to complex with C . Species B is able to perform the same reaction of A , and it can also be phosphorylated. Finally, species C can complex with $A : B$. Recall that the non-covalent bond of $A : B$ and $C : (A : B)$ can dissociate autonomously as described by rule 3 of the reduction semantics.

3 Consistency

In this section we investigate the relationship between Molecular Interaction Maps and MIM Calculus. We propose some *consistency* definitions, with the aim of identifying the terms of the calculus which could be formal representation of a MIM diagram. The first difference between MIM diagrams and the MIM calculus is that diagrams provide a static view of the species of a system, and of the interactions which can occur among the species, while MIM calculus allows representing single molecules and provides a semantics for deriving the evolution of the described system. MIM diagrams are also restricted with regards to the capabilities of species, since each single molecular species can appear only once in a diagram. For this reason, the informal interpretation of MIM diagrams assumes that the capabilities of each molecule depend only on the species of the molecule. Capabilities are irrespective, for example, of the different reactions that might produce a molecule of that species. On the contrary, the MIM calculus allows representing single molecules $\mu.IS$, and different molecules of the same species might have different capabilities. For example, term $P = \{ \xrightarrow{B} \mu_1 \} . A \mid \emptyset . A \mid P'$ contains two molecules of species A with different capabilities: the first one can bind to another molecule of species B , while the second one has no capabilities. Note that these two molecules of species A could have been obtained as a result of other reactions (for example, by transformation of other molecules), hence during any evolution of the system there may be some states in which all the molecules of the same species have the same capabilities while, in other states, this is not true. It appears to be of particular interest to establish which terms of MIMc represent MIM diagrams, in the sense that a MIM diagram can be associated with a term, and the term evolves in accordance with the behaviour intended by the diagram. One may also ask that in a term molecules of a certain species always have the same capabilities. This captures the constraint of uniqueness of species in MIM diagrams.

For this purpose, we present three different definitions of consistency of MIM calculus terms, namely *semantic consistency*, (*weak*) *syntactic consistency*, and *strong syntactic consistency*. Semantic consistency is the weakest form of consistency, and takes into account only terms that can be reached from the initial state. This form of consistency requires that, whenever a molecule of a certain species named N is produced (i.e. a molecule S , with $\lfloor S \rfloor = N$, appears in the top-level parallel composition), it always has the same capabilities.

Definition 3.1 (Semantic Consistency). A term P is *semantically consistent* iff

$$\forall \mu_1.S_1, \mu_2.S_2, P', P''. \quad \lfloor S_1 \rfloor \equiv \lfloor S_2 \rfloor \text{ and } P \rightarrow^* \mu_1.S_1 \mid P' \text{ and } P \rightarrow^* \mu_2.S_2 \mid P'' \text{ implies } \mu_1 \equiv \mu_2$$

The following definitions of syntactic consistency, deal instead with the species that syntactically appear in a term. The weak form requires that the capabilities of each molecule of a species appearing in the term, including those forming compound molecules and those that can be obtained as the result of reactions, always have the same capabilities. The definitions make use of contexts, in order to precisely identify the position inside the term in which molecules of a species (with their capabilities) appear.

Definition 3.2 ((Weak) Syntactic Consistency). A term P is (weakly) syntactic consistent iff

$$\forall C_1, C_2, \mu_1, \mu_2. \text{ name}(C_1) \equiv \text{ name}(C_2) \text{ and } P \equiv C_1[\mu_1] \equiv C_2[\mu_2] \text{ implies } \mu_1 \equiv \mu_2$$

Strong syntactic consistency adds a further constraint, by requiring also that, whenever a non-covalent bond (\longrightarrow) or a covalent bond (\equiv) can be created between two species, then both species have the corresponding capability. In the definition, we write $\gamma \in \mu$ as a shorthand for $\exists M. \mu \equiv \{M + \gamma\}$.

Definition 3.3 (Strong Syntactic Consistency). A term P is strongly syntactic consistent iff

- P is weakly syntactic consistent and
- $\forall C_1, C_2, \mu_1, \mu_2, N_1, N_2.$
 $\text{ name}(C_1) \equiv N_1$ and $\text{ name}(C_2) \equiv N_2$ and $P \equiv C_1[\mu_1] \equiv C_2[\mu_2]$ implies

$$\begin{aligned} \left(\xrightarrow{N_2} \mu \right) \in \mu_1 &\Leftrightarrow \left(\xrightarrow{N_1} \mu \right) \in \mu_2 \\ \left(\equiv^{N_2} \mu \right) \in \mu_1 &\Leftrightarrow \left(\equiv^{N_1} \mu \right) \in \mu_2 \end{aligned}$$

For example, term $P_2 = \{\xrightarrow{B} \mu_1\}.A \mid \emptyset.B$ is weakly syntactic consistent, but not strongly syntactic consistent, since molecule B does not have the capability of binding (with a non-covalent bond) to A . The strongly syntactic consistent term corresponding to P_2 is $P_3 = \{\xrightarrow{B} \mu_1\}.A \mid \{\xrightarrow{A} \mu_1\}.B$.

The following proposition shows that syntactic consistency implies semantic consistency.

Proposition 3.1. Syntactic Consistency entails Semantic Consistency, that is

$$\forall P. P \text{ is syntactically consistent} \Rightarrow P \text{ is semantically consistent.}$$

Proof. It is sufficient to prove that $P \rightarrow^* \mu.S \mid P'$ implies $\exists C. C[\mu] \equiv P$ and $\text{ name}(C) = \lfloor S \rfloor$. This proof is done by induction on the length of the sequence of transitions $P \rightarrow^* \mu.S \mid P'$. Let us assume that such a sequence has the following form $P \equiv P_0 \xrightarrow{\alpha_1} P_1 \xrightarrow{\alpha_2} \dots \xrightarrow{\alpha_n} P_n \equiv \mu.S \mid P'$.

As regards the base case ($n = 0$), we have $P \equiv \mu.S \mid P'$. Then, context $C = \square.S \mid P'$ is such that $C[\mu] \equiv P$ and $\text{ name}(C) = \lfloor S \rfloor$.

As regards the induction step, let $n > 1$ and suppose that the property holds for all $m < n$. We have $P_n \equiv \mu.S \mid P'$ and there are two cases to be considered: either $\mu.S$ already appeared before, i.e. $\exists \bar{n} < n. P_{\bar{n}} \equiv \mu.S \mid P''$ for some P'' , or not. In the first case, by induction hypothesis, there exists a context C such that $C[\mu] \equiv P$ and $\text{ name}(C) = \lfloor S \rfloor$. In the second case, $\mu.S$ has been created in the last execution step $P_{n-1} \xrightarrow{\alpha_n} \mu.S \mid P'$, where $\mu.S$ does not occur in the top-level parallel composition, i.e. $\nexists Q. P_{n-1} \equiv \mu.S \mid Q$.

According to the semantics, $P_{n-1} \xrightarrow{\alpha_n} \mu.S \mid P'$ iff $P_{n-1} \xrightarrow{(\emptyset, \iota)\alpha_n} \mu.S \mid P'$ for some $\iota \in \mathcal{N}$. By rule induction on the rules 2–10 of the semantics, we prove that, for all transitions $Q \xrightarrow{(\nu, \iota)\alpha_n} Q'$, for any $\mu.S$ created in the transition there is a context C such that $C[\mu] \equiv P$ and $\text{ name}(C) = \lfloor S \rfloor$.

- **Rule 2:** $\mu_1.S_1 \mid \mu_2.S_2 \xrightarrow{(\nu, \iota)\alpha} \mu.(\mu_1.S_1 : \mu_2.S_2)$ with $\mu_1 \equiv \{X + (\nu, \iota) \xrightarrow{\lfloor S_2 \rfloor} \mu\}$
 By induction hypothesis, there exists a context \bar{C} such that $\bar{C}[\mu_1] \equiv P$ and $\text{ name}(\bar{C}) = \lfloor S_1 \rfloor$. Therefore, context $C = \bar{C}[\{X + (\nu, \iota) \xrightarrow{\lfloor S_2 \rfloor} \square\}]$ is such that $C[\mu] \equiv P$ and $\text{ name}(C) = \lfloor \mu_1.S_1 : \mu_2.S_2 \rfloor$.

- *Rule 3:* $\mu.(\mu_1.S_1 : \mu_2.S_2) \xrightarrow{(\emptyset, \emptyset) \alpha} \mu_1.S_1 \mid \mu_2.S_2$
Let \bar{C} be the context of $\mu.(\mu_1.S_1 : \mu_2.S_2)$ (by induction hypothesis). Context \bar{C} must contain the portion $\square.(\mu_1.S_1 : \mu_2.S_2)$. Suppose that at least one of $\mu_1.S_1$ and $\mu_2.S_2$ never appeared before in any $P_m, m < n$ (otherwise, by induction hypothesis, their contexts are already known). Hence, term $S = \mu.(\mu_1.S_1 : \mu_2.S_2)$ has not been obtained by applying rule 2, but by one of the rules 3,4,5, 7,9. This means that S appeared literally in the initial term P , thus contexts C_1 can be obtained from \bar{C} by replacing $\square.(\mu_1.S_1 : \mu_2.S_2)$ with $\mu.(\square.S_1 : \mu_2.S_2)$ and contexts C_2 can be obtained from \bar{C} by replacing $\square.(\mu_1.S_1 : \mu_2.S_2)$ with $\mu.(\mu_1.S_1 : \square.S_2)$. Contexts C_1 and C_2 are such that $C_1[\mu_1] \equiv P$ with $name(C_1) = \lfloor S_1 \rfloor$ and $C_2[\mu_2] \equiv P$ with $name(C_2) = \lfloor S_2 \rfloor$.
- *Rule 4:* $\mu_1.S_1 \xrightarrow{(v, \iota) \alpha} \mu.S \mid P'$ with $\mu_1 = \{X + (v, \iota) \longrightarrow (\mu.S \mid P')\}$
Let \bar{C} be the context of $\mu_1.S_1$. Then the context for $\mu.S$ is $C = \bar{C}[\{X + (v, \iota) \longrightarrow (\square.S \mid P')\}]$.
- *Rule 5:* analogous to rule 4.
- *Rule 6:* analogous to rule 2.
- *Rule 7:* $\mu_1.S_1 \mid \mu'.\overline{(\mu_2.S_2)(\mu_3.S_3)} \xrightarrow{(v, \iota) \alpha} \mu_1.S_1 \mid \mu_2.S_2 \mid \mu_3.S_3$
Similarly to rule 3, if either $\mu_2.S_2$ or $\mu_3.S_3$ did not appear before, their contexts can be obtained from context \bar{C} of $\mu'.\overline{(\mu_2.S_2)(\mu_3.S_3)}$. Context \bar{C} must contain $\square.\overline{(\mu_2.S_2)(\mu_3.S_3)}$. We obtain context C by replacing $\square.\overline{(\mu_2.S_2)(\mu_3.S_3)}$ with $\mu'.(\square.S_2)(\mu_3.S_3)$ in \bar{C} for $\mu_2.S_2$. Similarly, we obtain context C by replacing $\square.\overline{(\mu_2.S_2)(\mu_3.S_3)}$ with $\mu'.(\mu_2.S_2)(\square.S_3)$ in \bar{C} for $\mu_3.S_3$.
- *Rule 8:* $\mu_1.S_1 \xrightarrow{(v, \iota) \alpha} \mu.\overline{q(\mu_1.S_1)}$ with $\mu_1 = \{X + (v, \iota) \xrightarrow{q} \mu\}$
Let \bar{C} be the context of $\mu_1.S_1$. Then the context for $\mu.\overline{q(\mu_1.S_1)}$ is $C = \bar{C}[\{X + (v, \iota) \xrightarrow{q} \square\}]$.
- *Rule 9:* analogous to rules 3 and 7.
- *Rule 10:* $P \mid Q \xrightarrow{(v', \iota) \alpha} P' \mid Q$
Since Q is not modified by the transition, only terms $\mu.S$ in P' could have been created by the transition. The contexts of any $\mu.S$ in P' is given by the induction hypothesis on the rule. \square

4 An Example of Modeling

In this section we show an example of a real molecular interaction map, taken from [15], and we show the corresponding term in MIM calculus. Differently from Kohn maps, the MIM calculus can contain multiple copies of the molecular species involved, thus it can describe the evolution of the system starting from an initial configuration.

The example in [15] presents a comprehensive molecular interaction map of regulators of cell cycle and DNA repair processes. The presented map is limited to the events in the mammalian cell nucleus. We consider here only the interaction between a protein of the E2F family and a gene promoter E2. This interaction is an important part of the cell cycle. The transcription of the gene is activated or inhibited by the binding of different complexes with the promoter. The molecular interaction map representing the interactions among stimulatory and inhibitory complexes of E2F1, DP1 and pRb is shown in Figure 8.

The E2F1:DP1 dimer (indicated by node (a) in Figure 8) and the (E2F1:DP1):pRb trimer (node (b)) can be bound to the promoter element E2. When the E2F1:DP1 dimer is bound to E2 the transcription activity is stimulated, while when the (E2F1:DP1):pRb trimer is bound to E2 the transcription is inhibited. We represent each species involved by a MIMc term with the capabilities of the species itself. In

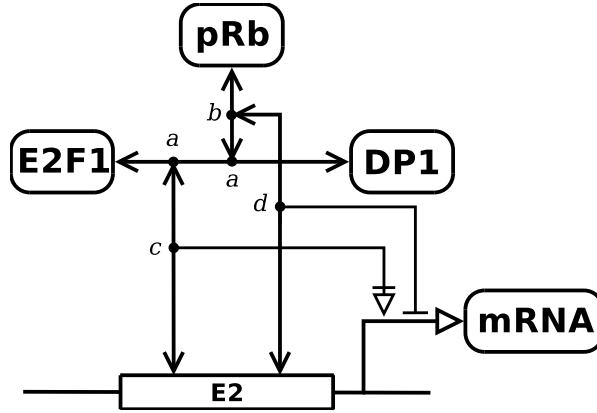


Figure 8: Molecular interaction map representing interactions among E2F1, DP1, and pRb.

particular we use particular terms for representing the promoter E2 and the DNA. In the Kohn map the DNA is implicitly represented, but, in a MIMc term DNA must be represented explicitly, for assigning to it the capability of producing the mRNA. Remark that we can have multiple copies of the species E2F1, DP1, and pRb in a MIMc term representing the system. Coherently with the cell system we can have only one copy of the DNA and of the gene promoter E2. Each basic element is identified by an elementary species name $E2F1, DP1, pRb, E2, DNA, mRNA \in \mathcal{E}$.

The E2F1 species can be represented by the following term:

$$\left\{ \xrightarrow{DP1} \left\{ \xrightarrow{E2} \emptyset + \xrightarrow{pRb} \left\{ \xrightarrow{E2} \emptyset \right\} \right\} \right\}.E2F1 = \mu_1.E2F1$$

which states that $E2F1$ can be bound to $DP1$, and then the dimer can be bound to $E2$. Note that the stimulation of DNA transcription by the trimer is not modeled among the capability of the species, which are just empty. Instead, this behaviour is captured by the DNA process, shown in the following. Moreover, the $E2F1 : DP1$ dimer can be bound to pRb and then to promoter $E2$ to inhibit the transcription. As for stimulation, inhibition is captured inside the definition of the DNA. Species $DP1$ and pRb are represented by the terms:

$$\begin{aligned} \left\{ \xrightarrow{E2F1} \left\{ \xrightarrow{E2} \emptyset + \xrightarrow{pRb} \left\{ \xrightarrow{E2} \emptyset \right\} \right\} \right\}.DP1 &= \mu_2.DP1 \\ \left\{ \xrightarrow{E2F1:DP1} \left\{ \xrightarrow{E2} \emptyset \right\} \right\}.pRb &= \mu_3.pRb \end{aligned}$$

Finally, the promoter E2 and the DNA can be represented by the terms:

$$\begin{aligned} \left\{ \xrightarrow{E2F1:DP1} \emptyset + \xrightarrow{(E2F1:DP1):pRb} \emptyset \right\}.E2 &= \mu_4.E2 \\ \left\{ (v_{DNA}, \iota_{DNA}) \longrightarrow mRNA \right\}.DNA &\quad \text{with} \quad \begin{cases} v_{DNA} = \{(E1F1 : DP1) : E2\} \\ \iota_{DNA} = \{((E2F1 : DP1) : pRb) : E2\} \end{cases} \end{aligned}$$

The lossless production of $mRNA$ by the DNA is regulated by the presence/absence of the two complexes $(E1F1 : DP1) : E2$ and $((E2F1 : DP1) : pRb) : E2$. In particular, the former complex represents a promoter (triggering the reaction), while the latter represents an inhibitor for the reaction.

An initial configuration in which two copies of species E2F1, DP1 and pRb are present is represented

by the following MIMc term

$$P_1 = \mu_1.E2F1 \mid \mu_1.E2F1 \mid \mu_2.DP1 \mid \mu_2.DP1 \mid \mu_3.pRb \mid \mu_3.pRb \mid \mu_4.E2 \\ \mid \{(v_{DNA}, t_{DNA}) \multimap mRNA\}.DNA$$

The term can evolve towards different configurations. For example, after a complexation between $E2F1$ and $DP1$ occurs, the processes $\mu_1.E2F1$ and $\mu_2.DP1$ are replaced by the following term, representing a complex with name $E2F1 : DP1$:

$$\{\xrightarrow{E2}\emptyset + \xrightarrow{pRb}\{\xrightarrow{E2}\emptyset\}\}.(\mu_1.E2F1 : \mu_2.DP1)$$

Thus the whole term becomes:

$$P_2 = \mu_1.E2F1 \mid \mu_2.DP1 \mid \mu_3.pRb \mid \mu_3.pRb \mid \mu_4.E2 \mid \{(v_{DNA}, t_{DNA}) \multimap mRNA\}.DNA \\ \mid \{\xrightarrow{E2}\emptyset + \xrightarrow{pRb}\{\xrightarrow{E2}\emptyset\}\}.(\mu_1.E2F1 : \mu_2.DP1)$$

As a further evolution step we may have the binding of the dimer $E2F1 : DP1$ to the promotor $E2$. The resulting term is:

$$P_3 = \mu_1.E2F1 \mid \mu_2.DP1 \mid \mu_3.pRb \mid \mu_3.pRb \mid \{(v_{DNA}, t_{DNA}) \multimap mRNA\}.DNA \\ \mid \emptyset. \left((\{\xrightarrow{E2}\emptyset + \xrightarrow{pRb}\{\xrightarrow{E2}\emptyset\}\}.(\mu_1.E2F1 : \mu_2.DP1)) : \mu_4.E2 \right)$$

As possible example of derivation, we show how the semantics is applied to the term P_1 above obtaining the term P_2 in a single reduction step. For the sake of readability, we write the terms P_1, P_2 as:

$$P_1 = \mu_1.E2F1 \mid \mu_2.DP1 \mid R \\ P_2 = \mu.(\mu_1.E2F1 : \mu_2.DP1) \mid R$$

where

$$R = \mu_1.E2F1 \mid \mu_2.DP1 \mid \mu_3.pRb \mid \mu_3.pRb \mid \mu_4.E2 \mid \{(v_{DNA}, t_{DNA}) \multimap mRNA\}.DNA \\ \mu = \{\xrightarrow{E2}\emptyset + \xrightarrow{pRb}\{\xrightarrow{E2}\emptyset\}\}.$$

The transition $P_1 \xrightarrow{(v', t)} P_2$, with $v' = t = \emptyset$, is obtained with the following derivation, by using the rules of the semantics:

$$\frac{\frac{\mu_1 = \{X + (v, t) \xrightarrow{DP1} \mu\} \quad v = t = \emptyset \quad E2F1, DP1 \notin t}{\mu_1.E2F1 \mid \mu_2.DP1 \xrightarrow{(v, t) \quad E2F1 \leftrightarrow DP1} \mu.(\mu_1.E2F1 : \mu_2.DP1)} \quad [R] \cap t = \emptyset \quad v' = v \setminus [R] = \emptyset}{\mu_1.E2F1 \mid \mu_2.DP1 \mid R \xrightarrow{(v', t) \quad E2F1 \leftrightarrow DP1} \mu.(\mu_1.E2F1 : \mu_2.DP1) \mid R}$$

Finally, according to property (1), we have the transition $P_1 \xrightarrow{E2F1 \leftrightarrow DP1} P_2$.

5 Conclusions

Using formal methods for studying biological systems is an interesting approach that allows using many different analysis techniques. We have defined the MIM calculus, a new calculus with high-level operators directly inspired by Molecular Interaction Maps (MIM), a graphical notation used in biology. This approach allows exploiting the features of process calculi, such as incremental definition of models, techniques for analysis and verification of properties, and the development of simulators. Moreover, the correspondence of the operators of the calculus with biological reactions allows an easy translation of Molecular Interaction Maps into terms of the MIM calculus. We have studied conditions under which a term of the MIM calculus is a formal representation of a MIM diagram, and we have provided different consistency definitions for the terms of the MIM calculus.

In the future, we plan to investigate the properties of calculus, such as its expressiveness, and to develop different extensions for a better description of biological systems.

References

- [1] Barbuti, R., Maggiolo-Schettini, A., Milazzo, P.: Extending the calculus of looping sequences to model protein interaction at the domain level. In: *Int. Symposium on Bioinformatics Research and Applications (ISBRA'07)*. Volume 4463 of LNBI., Springer (2007) 638–649
- [2] Barbuti, R., Maggiolo-Schettini, A., Milazzo, P., Tiberi, P., Troina, A.: Stochastic Calculus of Looping Sequences for the Modeling and Simulation of Cellular Pathways *Transactions on Computational Systems Biology IX* (2008) 86–113.
- [3] Barbuti, R., Maggiolo-Schettini, A., Milazzo, P., Troina, A.: A calculus of looping sequences for modelling microbiological systems. *Fundamenta Informaticae* **72** (2006) 21–35
- [4] Barbuti, R., Lepri, D., Maggiolo-Schettini, A., Milazzo, P., Pardini, G., Rama, A.: Simulation of Kohn's Molecular Interaction Maps through translation into Stochastic CLS. In: *Perspectives of System Informatics (PSI'09)*. Springer LNCS, in press.
- [5] Bortolussi, L., Fonda, S., Policriti, A.: Constraint based simulation of biological systems described by Molecular Interaction Maps. In: *IEEE conference on Bioinformatics and Biomedicine (BIBM 2007)*, IEEE Computer Society, 288–293
- [6] Brodo, L., Degano, P., Priami, C.: A stochastic semantics for bioambients. In: *Parallel Computing Technologies (PaCT'07)*. Volume 4671 of LNCS., Springer (2007) 22–34
- [7] Cardelli, L.: Brane calculi. interactions of biological membranes. In: *Computational Methods in Systems Biology (CMSB'04)*. Volume 3082 of LNCS., Springer (2005) 257–280
- [8] Chabrier-Rivier, N., Chiaverini, M., Danos, V., Fages, F., Schachter, V.: Modeling and querying biomolecular interaction networks. *Theoretical Computer Science* **325**(1) (2004) 25–44
- [9] Chiaverini, M., Danos, V.: A core modeling language for the working molecular biologist. In: *Computational Methods in Systems Biology (CMSB'03)*. Volume 2602 of LNCS., Springer (2003) 166
- [10] Ciocchetta, F., Hillston, J.: Bio-PEPA: an extension of the process algebra PEPA for biochemical networks. In: *Proc. of FBTC2007*. ENTCS Volume 194(3) (2008) 103–117
- [11] Ciocchetta, F., Priami, C., Quaglia, P.: Modeling Kohn Interaction Maps with Beta-Binders: An Example. *Transactions on Computational Systems Biology* **3** (2005) 33–48
- [12] Curti, M., Degano, P., Priami, C., Baldari, C.: Modelling biochemical pathways through enhanced pi-calculus. *Theoretical Computer Science* **325**(1) (2004) 111–140
- [13] Danos, V., Laneve, C.: Formal molecular biology. *Theoretical Computer Science* **325**(1) (2004) 69–110
- [14] Kitano, H.: Graphical Notation for Biochemical Networks. *BIOSILICO* **1** Number 5 (2003)

- [15] Kohn, K. W.: Molecular Interaction Map of the Mammalian Cell Cycle Control and DNA Repair Systems. *Molecular Biology of the Cell* **10** 2703–2734 (1999)
- [16] Kohn, K. W., Aladjem, M. I., Kim, S., Weinstein, J. N., Pommier, Y.: Depicting combinatorial complexity with the molecular interaction map notation. *Molecular Systems Biology* **2** 1–12 (2006)
- [17] Kohn, K. W., Aladjem, M. I., Weinstein, J. N., Pommier, Y.: Molecular Interaction Maps of Bioregulatory Networks: A General Rubric for Systems Biology. *Molecular Biology of the Cell* **17** (2005) 1–13
- [18] Laneve, C., Tarissan, F.: A simple calculus for proteins and cells. In: *Membrane Computing and Biological Inspired Process Calculi (MeCBIC'06)*. Volume 171 of ENTCS., Elsevier (2007) 139–154
- [19] Milner, R.: *Communicating and Mobile Systems: the π -Calculus*. Cambridge University Press (1999)
- [20] Priami, C.: Stochastic π -calculus. *The Computer Journal* **38**(7) (1995) 578–589
- [21] Priami, C., Quaglia, P.: Beta binders for biological interactions. In: *Computational Methods in Systems Biology (CMSB'04)*. Volume 3082 of LNCS., Springer (2005) 20–33
- [22] Priami, C., Regev, A., Silverman, W., Shapiro, E.: Application of a stochastic name-passing calculus to representation and simulation of molecular processes. *Information Processing Letters* **80** (2001) 25–31
- [23] Regev, A.: *Computational Systems Biology: a Calculus for Biomolecular Knowledge*. PhD thesis, Tel Aviv University, 2002
- [24] Regev, A., Panina, E., Silverman, W., Cardelli, L., Shapiro, E.: Bioambients: An abstraction for biological compartments. *Theoretical Computer Science* **325**(1) (2004) 141–167
- [25] Regev, A., Shapiro, E.: Cells as computation. *Nature* **419** (2002) 343
- [26] Regev, A., Silverman, W., Shapiro, E.: Representing Biomolecular Processes with Computer Process Algebra: π -Calculus programs of Signal Transduction pathways. (2000)
- [27] Regev, A., Silverman, W., Shapiro, E.: Representation and simulation of biochemical processes using the π -calculus process algebra. In: *Pacific Symposium on Biocomputing*, World Scientific Press (2001) 459–470
- [28] Versari, C., Gorrieri, R.: $\pi@$: a π -Based Process Calculus for the Implementation of Compartmentalised Bio-inspired Calculi. In: *Formal Methods for Computational Systems Biology*. LNCS 5016 (2008) 449–506