Brane Calculi
Interactions of Biological Membranes

Luca Cardelli

Microsoft Research

Abstract. We introduce a family of process calculi with dynamic nested membranes. In contrast to related calculi, including some developed for biological applications, active entities here are tightly coupled to membranes, and can perform interactions on both sides of a membrane. That is, computation happens *on* the membrane, not inside of it.

1 Introduction

A biological cellular membrane is an oriented closed surface that can perform various molecular functions. Membranes are not just containers: they are coordinators and active sites of major activity[1]. Large functional molecules (proteins) are embedded in membranes, with consistent orientation, and can act on both sides of the membrane simultaneously. The consistent orientation of such proteins induces an orientation on the membrane. Freely floating molecules interact with membrane proteins, and can be sensed, manipulated, and pushed across by active molecular channels. Membranes come in different kinds, distinguished mostly by the proteins embedded in them, and typically consume energy to perform their functions.

One of the most remarkable properties of biological membranes is that they form a two-dimensional fluid (a lipid bilayer) embedded in a three-dimensional fluid (water). That is, both the structural components and the embedded proteins freely diffuse on the two-dimensional plane of the membrane (unless they are held together by specific mechanisms). Moreover, membranes float in water, which may contain other molecules that freely diffuse in that three-dimensional fluid. Membrane themselves are impermeable to most substances, such as water and protons, so that they partition the three-dimensional fluid.

Many membranes are highly dynamic: they constantly shift, merge, break apart, and are replenished. But the transformations that they can support are rather limited, partially because orientation must be preserved, and partially because membrane transformations need to be fairly continuous. For example, it is possible for a membrane to gradually buckle and create a bubble that then detaches, or for such a bubble to merge back with a membrane, but it is not possible for a bubble to "jump across" a membrane (only small molecules can do that).

The fluid-within-fluid structure inspires the basic organization of our Brane Calculi[2], which is characterized by two commutative monoids, each representing a kind of fluid. The specific transformations that we have selected are further inspired by (some of) the biological

[1] "For a cell to function properly, each of its numerous proteins must be localized to the correct cellular membrane or aqueous compartment." [9] p.675.

[2] "Brane" is a common abbreviation for "membrane" in physics.

constraints. However, within the general structure of Brane Calculi there is scope for refining or ignoring such constraints.

One of the constraints one may adopt is the preservation of orientation (e.g., membranes of different orientation should not merge). A related constraint is *bitonality*, which requires nested membranes to have opposite orientations, so that the orientations can be coded by coloring systems in two tones, as in Figure 1, where P and Q represent arbitrary subsystems. Preservation of bitonality means that reactions must preserve the even/odd parity with which components are nested inside membranes: note that P and Q remain on the same color background in each reaction. This means, in particular, that in a sequence of bitonal reactions there is never any actual mixing of fluids from inside and outside any given membrane, although external fluids can be brought inside if safely wrapped in another membrane. Bitonality is common in cellular-scale living systems. Although not universal, it inspires a collection of basic reactions that are biologically implementable, and that are different from those of calculi that are not biologically inspired.[3]

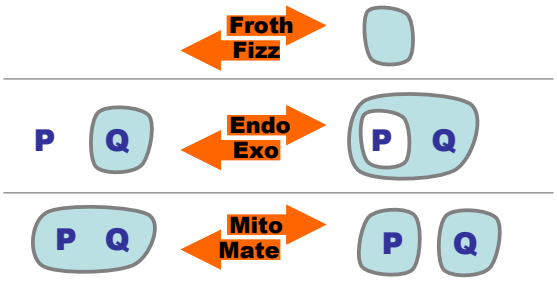

Figure 1 Examples of Bitonal Reactions

The reactions illustrated in Figure 1 can be formalized and studied on their own [2]. However, in this paper we use them only as informal guides for more detailed calculi, where the reasons "why" those reactions happen are made more apparent.

2 Basic Framework

2.1 Syntax and Reactions

The basic structure of Brane Calculi consists of two commutative monoids with replication: we use $\circ$ for composition of *systems*, with unit $\diamond$, and $|$ for composition of *membranes*, with unit 0. Replication (!) is used to model the notion of a "multitude" of components of the same kind, which is in fact a standard situation in biology. Quantitative refinements are possible [12] and certainly desirable.

Systems consist of nested membranes, and membranes consists of collection of *actions*. Actions are left unspecified at the moment, and are detailed in the following sections. The familiar notion of structural congruence of processes [11] is applied to systems and membranes, characterizing their fluidity properties. Reactions happen only at the level of systems, and are caused only by actions on membranes.

[3] The framework in which Brane Calculi are formalized originates in the study of calculi for mobile agents [3]. In that context, *sandboxing* an applet on its arrival at a site is, in fact, a bitonal operation: it maintains the separation between safe regions (of internal origin) and unsafe regions (of external origin). We are not aware of proposals to use sandboxing as a basic operations in that context; here, it corresponds to phagocytosis.

Syntax

Systems	$P,Q ::= \diamond \mid P \circ Q \mid !P \mid \sigma\langle P \rangle$	nests of membranes
Branes	$\sigma,\tau ::= 0 \mid \sigma \mid \tau \mid !\sigma \mid a.\sigma$	combinations of actions
Actions	$a,b ::= \ldots$	(detailed later)

A membrane $\sigma\langle P \rangle$
with actions σ and contents P

Figure 2 Brane Graphical Notation

We abbreviate $a.0$ as a, and $0\langle P \rangle$ as $\langle P \rangle$, and $\sigma\langle \diamond \rangle$ as $\sigma\langle\rangle$.

Structural Congruence

$P \circ Q \equiv Q \circ P$	$\sigma \mid \tau \equiv \tau \mid \sigma$
$P \circ (Q \circ R) \equiv (P \circ Q) \circ R$	$\sigma \mid (\tau \mid \rho) \equiv (\sigma \mid \tau) \mid \rho$
$P \circ \diamond \equiv P$	$\sigma \mid 0 \equiv \sigma$
$!\diamond \equiv \diamond$	$!0 \equiv 0$
$!(P \circ Q) \equiv !P \circ !Q$	$!(\sigma \mid \tau) \equiv !\sigma \mid !\tau$
$!!P \equiv !P$	$!!\sigma \equiv !\sigma$
$!P \equiv P \circ !P$	$!\sigma \equiv \sigma \mid !\sigma$
$0\langle \diamond \rangle \equiv \diamond$	
$P \equiv Q \Rightarrow P \circ R \equiv Q \circ R$	$\sigma \equiv \tau \Rightarrow \sigma \mid \rho \equiv \tau \mid \rho$
$P \equiv Q \Rightarrow !P \equiv !Q$	$\sigma \equiv \tau \Rightarrow !\sigma \equiv !\tau$
$P \equiv Q \wedge \sigma \equiv \tau \Rightarrow \sigma\langle P \rangle \equiv \tau\langle Q \rangle$	$\sigma \equiv \tau \Rightarrow a.\sigma \equiv a.\tau$

Basic Reactions

$P \rightarrow Q \Rightarrow P \circ R \rightarrow Q \circ R$
$P \rightarrow Q \Rightarrow \sigma\langle P \rangle \rightarrow \sigma\langle Q \rangle$
$P \equiv P' \wedge P' \rightarrow Q' \wedge Q' \equiv Q \Rightarrow P \rightarrow Q$

We write $\rightarrow^*$ for the reflexive and transitive closure of $\rightarrow$.

Within this framework, our Basic Brane Calculus is the one gradually introduced in Sections 3 and 4. Possible extensions are discussed in Section 5. Orthogonally, one could add restriction operators to both systems and membranes, in the style of π-calculus [11], with extrusion rules such as $((\nu n)\sigma)\langle P \rangle \equiv (\nu n)(\sigma\langle P \rangle)$ if $n \notin fn(P)$. The bound names n would be the ones used in the following sections to identify pairs of related actions and co-actions.

3 Bitonal Interactions

Bitonal interactions [2] are inspired by endocytosis/exocytosis (the second reversible reaction in Figure 1). Endocytosis is the process of incorporating external material into a cell by

"engulfing" it with the cell membrane (without breaking the membrane or letting the material cross it). Exocytosis is the reverse process.

3.1 Definitions

Endocytosis, thus described, is an uncontrollable process that can engulf an arbitrary amount of material. We are interested in more controllable interactions, therefore we specialize endocytosis into two basic operations: *phagocytosis*, engulfing just one external membrane, and *pinocytosis*, engulfing zero external membranes. In addition we have *exocytosis*, which is itself sufficiently controllable. Each action usually comes with a co-action that it is intended to interact with, indicated by the symbol $^{\perp}$ (pinocytosis does not have a co-action).

Bitonal Actions

Actions	$a ::= \ldots \mid \circlearrowright_n \mid \circlearrowright^{\perp}_n(\sigma) \mid \circlearrowleft_n \mid \circlearrowleft^{\perp}_n \mid \circledcirc(\sigma)$	phago $\circlearrowright$, exo $\circlearrowleft$, pino $\circledcirc$

Precedence: $a.\sigma|\tau$ stands for $(a.\sigma)|\tau$, and $!\sigma|\tau$ stands for $(!\sigma)|\tau$. The subscripted names n are used to pair-up related actions an co-actions; we omit them when there is no ambiguity. Co-phago is indexed by a membrane σ; this σ becomes the new membrane that engulfs the outside material: conceptually it is related to a piece of the old membrane. Exo causes irreversible mixing of membranes: since membranes are fluids, there is in general no way to untangle two membranes once they have merged. Incidentally, this implies that merging is often not a desirable operation.

Bitonal Reactions

Phago	$\circlearrowright_n.\sigma\|\sigma_0\langle P\rangle \circ \circlearrowright^{\perp}_n(\rho).\tau\|\tau_0\langle Q\rangle \rightarrow \tau\|\tau_0\langle\rho\langle\sigma\|\sigma_0\langle P\rangle\rangle\circ Q\rangle$
Exo	$\circlearrowleft^{\perp}_n.\tau\|\tau_0\langle\circlearrowleft_n.\sigma\|\sigma_0\langle P\rangle\circ Q\rangle \rightarrow P \circ \sigma\|\sigma_0\|\tau\|\tau_0\langle Q\rangle$
Pino	$\circledcirc(\rho).\sigma\|\sigma_0\langle P\rangle \rightarrow \sigma\|\sigma_0\langle\rho\langle\diamond\rangle\circ P\rangle$

One can see that the parity of nesting of P and Q is preserved in all these reactions, hence they preserve the bitonal coloring of those subsystems.

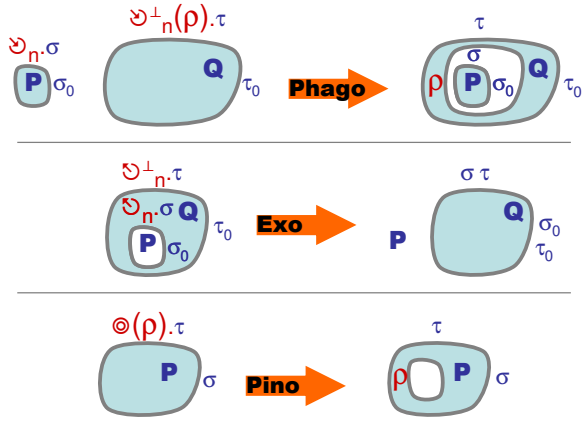

Figure 3 Phago, Exo, Pino (shaded for emphasis)

3.2 Derived Bitonal Interactions

The Mito reaction, as illustrated in Figure 1 is another uncontrollable process that can split a membrane at an arbitrary place. To make it more controllable, we specialize it into two basic operations: *budding*, splitting off one internal membrane, and *dripping*, splitting off zero internal membranes. In addition we have *mating* (a.k.a. merging or fusion), the obvious merging of membranes, which is itself sufficiently controllable.

These three bitonal operations, mating, budding, and dripping, can be derived from the previous three. The derivations are not meant to be biologically significant: they are just a test of expressive power. In practice one would want to consider these as primitives at the same level as Phago, Exo, and Pino, since they all have direct implementations in cellular mechanisms.

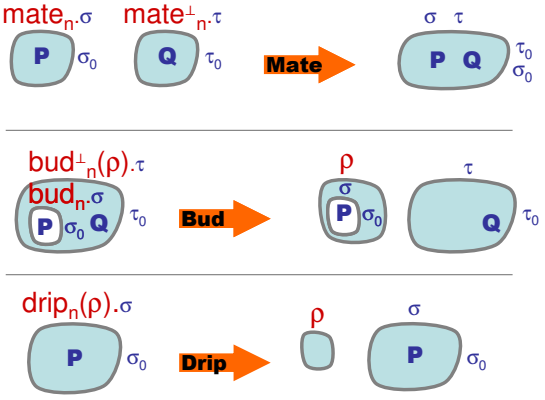

Figure 4 Mate, Bud, Drip (shaded for emphasis)

Mate causes irreversible membrane mixing, as in Exo. In Bud, the fresh membrane ρ that surrounds the bud is a parameter of the co-action, similarly to the situation with Phago. Drip is similar to Pino, but towards the outside.

The encodings of Mate, Bud, and Drip follow the single basic idea that Mito/Mate in Figure 1 can be encoded with a sequence of three Endo/Exo operations.

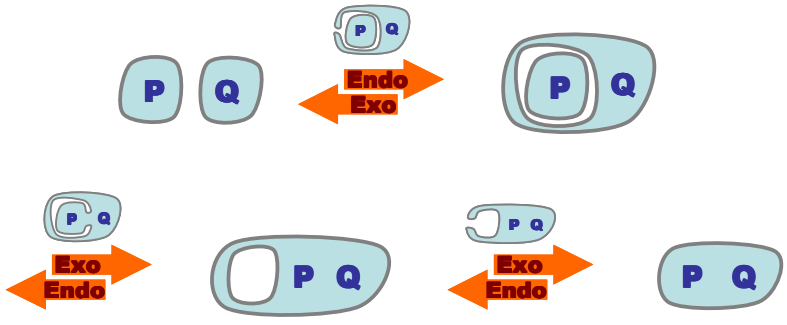

Figure 5 Mito/Mate by 3 Endo/Exo (basic technique)

Mate

$\text{mate}_n.\sigma \triangleq \mathcal{Y}_n.\mathcal{Y}_{n'}.\sigma$
$\text{mate}^{\perp}_n.\tau \triangleq \mathcal{Y}^{\perp}_n(\mathcal{Y}^{\perp}_{n'}.\mathcal{Y}_{n''}).\mathcal{Y}^{\perp}_{n''}.\tau$

$\text{mate}_n.\sigma|\sigma_0\langle P\rangle \circ \text{mate}^{\perp}_n.\tau|\tau_0\langle Q\rangle \rightarrow^* \sigma|\sigma_0|\tau|\tau_0\langle P \circ Q\rangle$

$\text{mate}_n.\sigma|\sigma_0\langle P\rangle \circ \text{mate}^{\perp}_n.\tau|\tau_0\langle Q\rangle =$
$\mathcal{Y}_n.\mathcal{Y}_{n'}.\sigma|\sigma_0\langle P\rangle \circ \mathcal{Y}^{\perp}_n(\mathcal{Y}^{\perp}_{n'}.\mathcal{Y}_{n''}).\mathcal{Y}^{\perp}_{n''}.\tau|\tau_0\langle Q\rangle \rightarrow_{\text{Phago n}}$
$\mathcal{Y}^{\perp}_{n''}.\tau|\tau_0\langle\mathcal{Y}^{\perp}_{n'}.\mathcal{Y}_{n''}\langle\mathcal{Y}_{n'}.\sigma|\sigma_0\langle P\rangle\rangle \circ Q\rangle \rightarrow_{\text{Exo n'}}$
$\mathcal{Y}^{\perp}_{n''}.\tau|\tau_0\langle\mathcal{Y}_{n''}|\sigma|\sigma_0\langle\diamond\rangle\rangle P \circ Q\rangle \rightarrow_{\text{Exo n''}}$
$\sigma|\sigma_0|\tau|\tau_0\langle P \circ Q\rangle$

Bud

$\text{bud}_n.\sigma \triangleq \mathcal{Y}_n.\sigma$
$\text{bud}^{\perp}_n(\rho).\tau \triangleq \circledcirc(\mathcal{Y}^{\perp}_n(\rho).\mathcal{Y}_{n'}).\mathcal{Y}^{\perp}_{n'}.\tau$

$\text{bud}^{\perp}_n(\rho).\tau|\tau_0\langle\text{bud}_n.\sigma|\sigma_0\langle P\rangle \circ Q\rangle \rightarrow^* \rho\langle\sigma|\sigma_0\langle P\rangle\rangle \circ \tau|\tau_0\langle Q\rangle$

$\text{bud}^{\perp}_n(\rho).\tau|\tau_0\langle\text{bud}_n.\sigma|\sigma_0\langle P\rangle \circ Q\rangle =$
$\circledcirc(\mathcal{Y}^{\perp}_n(\rho).\mathcal{Y}_{n'}).\mathcal{Y}^{\perp}_{n'}.\tau|\tau_0\langle\mathcal{Y}_n.\sigma|\sigma_0\langle P\rangle \circ Q\rangle \rightarrow_{\text{Pino}}$
$\mathcal{Y}^{\perp}_{n'}.\tau|\tau_0\langle\mathcal{Y}^{\perp}_n(\rho).\mathcal{Y}_{n'}\langle\diamond\rangle \circ \mathcal{Y}_n.\sigma|\sigma_0\langle P\rangle \circ Q\rangle \rightarrow_{\text{Phago n}}$
$\mathcal{Y}^{\perp}_{n'}.\tau|\tau_0\langle\mathcal{Y}_{n'}\langle\rho\langle\sigma|\sigma_0\langle P\rangle\rangle\rangle \circ Q\rangle \rightarrow_{\text{Exo n'}}$
$\rho\langle\sigma|\sigma_0\langle P\rangle \circ \tau|\tau_0\langle Q\rangle$

Drip

$\text{drip}_n(\rho).\sigma \triangleq \circledcirc(\circledcirc(\rho).\mathcal{Y}_n)).\mathcal{Y}^{\perp}_n.\sigma$

$\text{drip}_n(\rho).\sigma|\sigma_0\langle P\rangle \rightarrow^* \rho\langle\rangle \circ \sigma|\sigma_0\langle P\rangle$

$\text{drip}_n(\rho).\sigma|\sigma_0\langle P\rangle =$
$\circledcirc(\circledcirc(\rho).\mathcal{Y}_n)).\mathcal{Y}^{\perp}_n.\sigma|\sigma_0\langle P\rangle \rightarrow_{\text{Pino}}$
$\mathcal{Y}^{\perp}_n.\sigma|\sigma_0\langle\circledcirc(\rho).\mathcal{Y}_n\langle\diamond\rangle\rangle \circ P\rangle \rightarrow_{\text{Pino}}$
$\mathcal{Y}^{\perp}_n.\sigma|\sigma_0\langle\mathcal{Y}_n\langle\rho\langle\diamond\rangle\rangle\rangle \circ P\rangle \rightarrow_{\text{Exo n}}$
$\rho\langle\rangle \circ \sigma|\sigma_0\langle P\rangle$

3.3 Example: Viral Infection, Part 1

Certain kinds of viral infection mechanisms represent an ideal example of bitonality in action. A virus is too big to just cross a cellular membrane. It can either punch its DNA or RNA through the membrane, essentially performing a Mate, or it can enter by utilizing standard cellular endocytosis pathways, as shown in Figure 6.

The Semliki Forest virus consists of a capsid containing the viral RNA (the nucleocapsid). The nucleocapsid is surrounded by a membrane that is similar to the cellular membrane (in fact, it is obtained from it "on the way out"). This membrane is however enriched with a special protein that plays a crucial trick on the cellular machinery, as we shall see shortly. The virus is brought into the cell by phagocytosis, thus wrapped by an additional

membrane layer; this is part of a standard transport pathway into the cell. As part of that pathway, an endosome compartment merges with the wrapped-up virus. At this point, usually, the endosome causes some reaction to happen in the material brought into the cell. In this case, though, the virus uses its special membrane protein to trigger an exocytosis step that deposits the naked nucleocapsid into the cytosol. The careful separation of internal and external substances that the cell usually maintains has now been subverted. The nucleocapsid is in direct contact with the inner workings of the cell, and can begin doing damage. First, the nucleocapsid disassembles itself, depositing the viral RNA into the cytosol. This vRNA then follows three distinct paths. First it is replicated (either by cellular proteins, or by proteins that came with the capsid), to provide the vRNA for more copies of the virus. The vRNA is also translated into proteins, again by standard cellular machinery. Some proteins are synthesized in the cytosol, and form the building blocks of the capsid: these self-assemble and incorporate a copy of the vRNA to form a nucleocapsid. The virus envelope protein is instead synthesized in the Endoplasmic Reticulum, and through various steps (through the Golgi apparatus) ends up lining transport vesicles that merge with the cellular membrane, along another standard transport pathway. Finally, the newly assembled nucleocapsid makes contact with sections of the cellular membrane that are now lined with the viral envelope protein, and buds out to recreate the initial virus structure outside the cell.

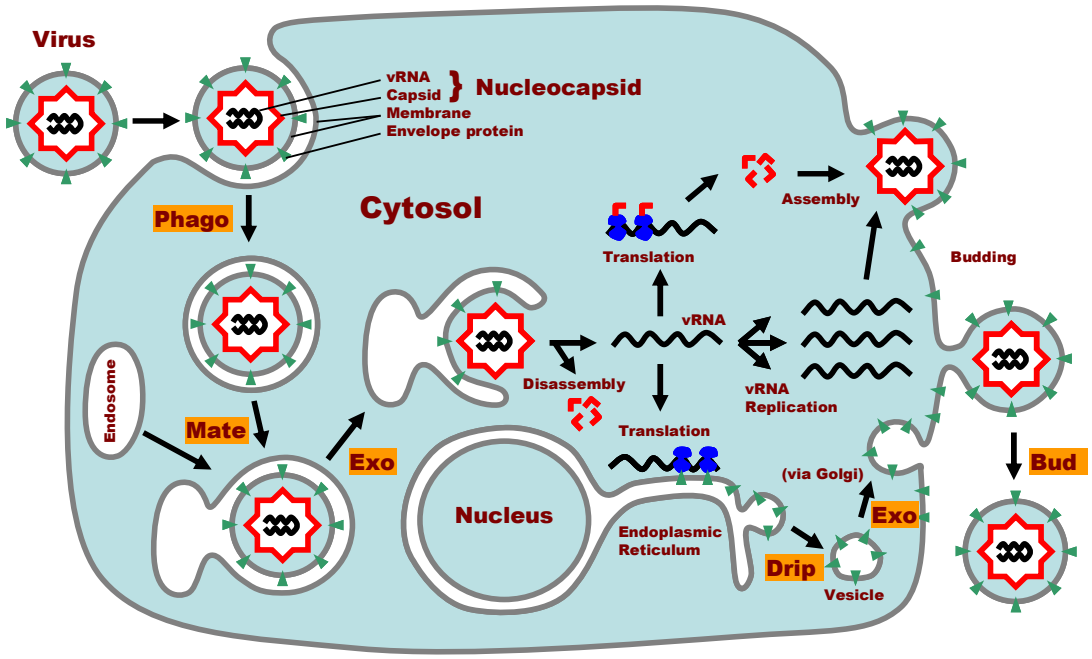

Figure 6 Viral Infection and Reproduction ([1] p.279)

The initial and final stages of the virus lifecycle can be coded up as follows.

virus	≙	ʚ.ʚ❨nucapʗ
nucap	≙	!budlX❨vRNAʗ
cell	≙	membrane❨cytosolʗ
membrane	≙	!ʚ⊥(mate)l!ʚ⊥

cytosol	≜	endosome ∘ Z
endosome	≜	!mate$^{\perp}$|!ↄ$^{\perp}$◖◗
viral-envelope	≜	bud$^{\perp}$(ↄ.ↄ)
envelope-vesicle	≜	ↄ.viral-envelope◖◗

In the first phase (infection, Figure 7), the nucleocapsid (i.e., the capsid with the viral RNA inside, abbreviated "nucap") places itself in the cytosol:

virus ∘ cell ➡* membrane◖nucap∘cytosol◗

We next assume that, by interaction with the available cellular machinery in the cytosol, the nucap causes the production of some number of copies n and m of envelope-vesicles and nucaps, leaving some modified cytosol'. (In section 4.6 we detail the mechanisms involved, including the unspecified cytosol', X, Z, Z'.)

nucap ∘ cytosol ➡* nucap^{n} ∘ envelope-vesicle^{m} ∘ cytosol'

In the final phase (reproduction, Figure 8), the virus reassembles itself outside the cell:

membrane◖nucap ∘ envelope-vesicle ∘ Z'◗ ➡* membrane◖Z'◗ ∘ virus

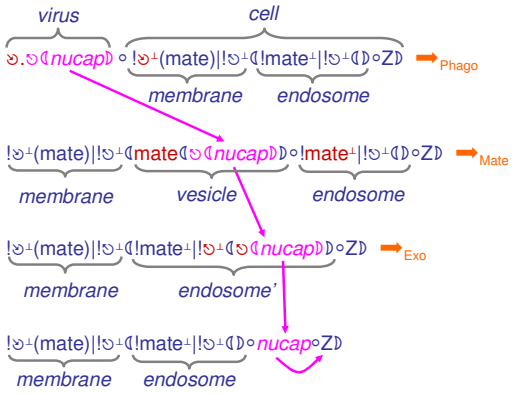

Figure 7 Viral Infection

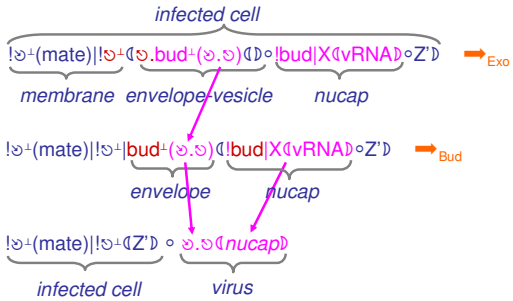

Figure 8 Viral Reproduction

The level of abstraction in this code has been chosen to be as close as possible to the one in the picture. This is important, because we rarely understand the finest details of biological

processes, and even if we did, we still would not want to model every molecule individually. The reality of virus infection is of course much more complex, and the modeling could be correspondingly refined. But one has to be able to choose an appropriate level of abstraction: Brane Calculi aim to provide such a level of abstraction for dynamic membrane transformations.

4 Molecules

We have not discussed free-floating molecules so far, to emphasize membrane interactions. Still, a primary function of membranes and of their embedded proteins is to shuttle molecules across, and it is important to include this ability in our models. In this section we discuss only *small* molecules, the ones that can easily cross or be transported across membranes. See sections 4.7 and 5.4 for a discussion of *large* molecules.

Membranes may let certain small molecules through by simple diffusion. Usually, however, they shuttle specific molecules through molecular channels that are implemented by sophisticated membrane-bound proteins (represented by our *actions*). Membranes are also a favorite mooring point of catalysts that cause free-floating molecules to interact with each other without crossing the membrane (e.g. in processes as basic as protein synthesis). Moreover, free-floating molecules can act as communication tokens between different membranes. A simplifying assumption for now is that small molecules do not change, do not have internal structure, and do not interact among themselves. All interactions between small molecules are mediated by membranes.

4.1 Definitions

Membranes can bind molecules on either sides of their surface, and can release molecules on either sides of their surface. Usually, coordinated bindings and releases happen completely or not at all, as in the antiporter in Figure 10. Because of this, we integrate in a single new action the ability to bind and release multiple molecules simultaneously.

Molecules and Molecular Actions

Systems	$P,Q ::= \ldots \mid m$	systems extended with molecules $m \in M$
	$p,q ::= m_1 \circ \ldots \circ m_k$	multisets of molecules
Actions	$a,b ::= \ldots \mid p_1(p_2) \rightleftarrows q_1(q_2)$	bind&release of molecules
B&R	$p_1 \circ p_1(p_2) \rightleftarrows q_1(q_2).\sigma \mid \sigma_0 \{ p_2 \circ P \} \rightarrow q_1 \circ \sigma \mid \sigma_0 \{ q_2 \circ P \}$	

A set of molecules M is added to the syntax of systems. A bind&release action is added to the set of actions. This action (Figure 9) binds, in general, a multiset of molecules outside the membrane (p_1) and a multiset of molecules inside the membrane (p_2); if that is possible, it instantly releases a multiset of molecules outside the membrane (q_1) and a multiset of molecules inside the membrane (q_2). (Conservation of mass or energy is not enforced, and must be designed in.)

Figure 9 Bind and Release

Obvious special cases are the separate binding and release on a single side; we omit ⋄(⋄):

$p_1(\diamond)\rightrightarrows$	bind outside	$\rightrightarrows q_1(\diamond)$	release outside
$\diamond(p_2)\rightrightarrows$	bind inside	$\rightrightarrows\diamond(q_2)$	release inside

4.2 Example: Chemical Reactions

A chemical reaction between molecules can be represented as a *catalyst*: an always empty membrane that enables a reaction via an appropriate bind-outside&release-outside action. Therefore, an explicit catalyst has to be present for a certain reaction to happen. This may be a bit artificial for simple chemistry, but most biological reactions are actively controlled or enhanced by catalysts.

$p \longrightarrow q$	$\triangleq$	$!\, p(\diamond) \rightrightarrows q(\diamond)\ ⦇⦈$	Chemical reaction
$p \Longleftrightarrow q$	$\triangleq$	$p \longrightarrow q \circ q \longrightarrow p$	Reversible reaction

For example, the reaction forming a peptide bond between two amino acids (with residues R^1 and R^2) can be written:

$$R^1R^2\text{PeptideBonding} \triangleq R^1\text{-COOH} \circ H_2\text{N-}R^2 \longrightarrow R^1\text{-CO-HN-}R^2 \circ H_2O$$

4.3 Example: Compartment Conditions

We can use an appropriate bind-inside&release-inside action to model chemical reactions that are specific to a given compartment; we call these *conditions* of the compartment. For example, certain chemical reactions happen only at a certain acidity, which is a compartment-wide property. An appropriate condition on the membrane of a compartment can represent acidity, and the evolution of conditions and compartments can represent changes of acidity. For example, the merging of a vesicle carrying some reagents with an endosome having a certain acidity condition, can cause the reagents to react after the merge because they find themselves in a compartment with the right acidity condition.

$p \dashrightarrow q$	$\triangleq$	$!\, \diamond(p) \rightrightarrows \diamond(q)$	Condition causing p to change into q
$p \dashleftarrow\!\dashrightarrow q$	$\triangleq$	$p \dashrightarrow q \circ q \dashrightarrow p$	Reversible condition
$p \dashrightarrow q \mid \sigma ⦇P⦈$			Compartment-wide condition affecting P
$p \dashrightarrow q \mid \sigma ⦇p⦈ \Rightarrow p \dashrightarrow q \mid \sigma ⦇q⦈$			A condition-driven reaction

4.4 Example: Molecular Pumps and Channels

A plant vacuole is a specialized membrane that stores nutrients, e.g. salt. The breakdown of ATP on the external surface of the vacuole, via a *proton pump*, is used to charge the interior

of the vacuole with protons (H^+). In general, several other specialized pumps and channels can be powered by such a charge. In a plant vacuole, a passive (but selective) *ion channel* can let chlorine ions (Cl^-) in, attracted by the excess electric charge of H^+. Transporting sodium ions (Na^+) inside is more difficult, because those are naturally repelled by the excess charge of H^+. A *proton antiporter*, however, can swap an Na^+ outside with an H^+ inside.

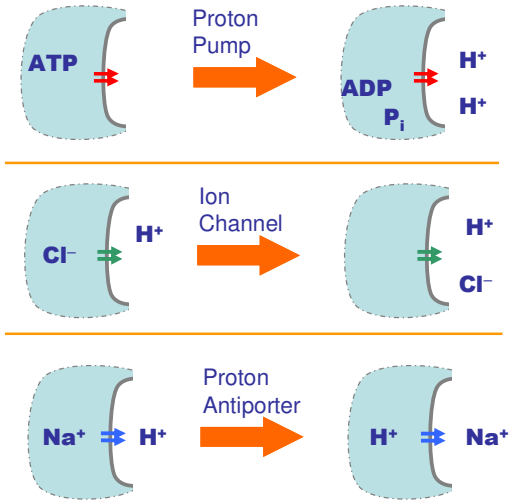

Figure 10 Molecular Channels

Each pump and channel is represented by a replicated bind&release action. These actions are then assembled as the membrane of an initially empty vacuole.

Plant Vacuole

ProtonPump	≙	$! \, ATP(\diamond) \rightrightarrows ADP \circ P_i(H^+ \circ H^+)$
IonChannel	≙	$! \, Cl^-(H^+) \rightrightarrows \diamond(H^+ \circ Cl^-)$
ProtonAntiporter	≙	$! \, Na^+(H^+) \rightrightarrows H^+(Na^+)$
PlantVacuole	≙	ProtonPump ∣ IonChannel ∣ ProtonAntiporter ↺

This is of course a qualitative representation of the process. Attaching reaction rates to the actions, as in Stochastic π-calculus [12] should yield quantitative modeling. Accurately modeling this situation should be quite interesting, because the reaction rates depend on the concentrations on both sides of the membrane.

4.5 Examples: Molecularly-Triggered Membrane Interactions

Molecular interactions can trigger membrane interactions, simply by sequencing the two kinds of actions on a membrane. In the following example, membrane A produces a molecule that stimulates membrane B to eat A:

Eat Me

A	≙	$\rightrightarrows n(\diamond).\, ⍥(P)$

$$B \triangleq n(\diamond)\rightrightarrows.\ \circlearrowleft^{\perp}(\rho)\langle Q\rangle$$

$$A \circ B \rightarrow \circlearrowleft\langle P\rangle \circ n \circ n(\diamond)\rightrightarrows.\ \circlearrowleft^{\perp}(\rho)\langle Q\rangle$$
$$\rightarrow \circlearrowleft\langle P\rangle \circ \circlearrowleft^{\perp}(\rho)\langle Q\rangle$$
$$\rightarrow \langle \rho\langle\langle P\rangle\rangle \circ Q\rangle$$

Pinocytosis, in reality, may incorporate molecular nutrients into the cell. Our basic pinocytosis operation does not do that, but it can be used as follows to recognize and incorporate external nutrients. Here n is a nutrient molecule, and C is a cell that recognizes it, transports it, and stores it in an internal vesicle.

Seek and Store

$$seek_n \triangleq !n(\diamond)\rightrightarrows.\ \circledcirc(\rightrightarrows\diamond(n).mate_{store})$$
$$store \triangleq !mate^{\perp}_{store}$$
$$C \triangleq seek_n\langle store\langle\rangle\rangle$$

$$n \circ C \rightarrow\rightarrow seek_n\langle \rightrightarrows\diamond(n).mate_{store}\langle\rangle \circ store\langle\rangle\rangle$$
$$\rightarrow seek_n\langle mate_{store}\langle n\rangle \circ store\langle\rangle\rangle$$
$$\rightarrow seek_n\langle store\langle n\rangle\rangle$$

4.6 Example: Viral Infection, Part 2

We can now complete the central part of the virus reproduction cycle, as shown in Figure 11. In section 3.3, we still had to provide a mechanism for the following reaction:

nucap ∘ cytosol ➡* nucap^{n} ∘ envelope-vesicle^{m} ∘ cytosol'

This can be obtained by the following definitions.

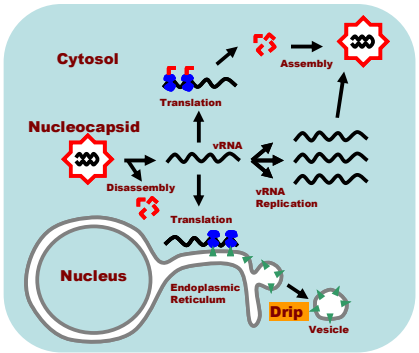

Figure 11 Nucleocapsid Replication (detail of Figure 6)

Nucleocapsid structure

$$nucap \triangleq capsid\langle vRNA\rangle$$
$$capsid \triangleq !bud \mid disasm$$
$$disasm \triangleq disasm\text{-}trigger(vRNA)\rightrightarrows vRNA(\diamond)$$

a) vRNA replication (Figure 11 middle-right)

vRNA-repl ≜ vRNA ⟶ vRNA ∘ vRNA

b) Capsomer translation and nucleocapsid assembly (Figure 11 top-right)

capsomer-tran ≜ !vRNA(⋄) ⇉ vRNA(⋄).drip(capsomers)◖◗

capsomers ≜ vRNA(⋄)⇉⋄(vRNA).capsid

c) Virus envelope protein translation and transport (Figure 11 bottom-right)

ER ≜ !vRNA(⋄)⇉vRNA(⋄). drip(↻.viral-envelope)◖Nucleus◗

Cytosol contents

cytosol ≜ endosome ∘ !disasm-trigger
∘ vRNA-repl ∘ capsomer-tran ∘ ER

A nucap particle is defined as a capsid containing vRNA (we do not model any other content of the capsid, for simplicity). The capsid surface is capable of either budding from the cell (as in section 3.3), or of disassembling the nucap by pushing the vRNA outside the capsid in response to some trigger molecule found in the cytosol (we do not model the fate of the disassembled capsid). There are then three paths that the newly freed vRNA follows:
(a) vRNA is replicated by the standard cellular machinery found in the cytosol:

vRNA-repl ∘ vRNA ➡ vRNA-repl ∘ vRNA ∘ vRNA

(b) The cellular machinery (modeled here by a fictitious empty membrane "capsomer-tran" with an active surface) translates vRNA into capsomer proteins that self-assemble (by dripping) into an entity that inserts vRNA from the cytosol into an empty capsid, hence producing a nucap:

capsomer-tran ∘ vRNA ➡* capsomer-tran ∘ nucap

(c) The E.R. translates vRNA into viral-envelope proteins that are collected (by dripping) into envelope-vesicles that are ready to merge (↻) with the cellular membrane as shown in section 3.3:

ER ∘ vRNA ➡* ER ∘ vRNA ∘ envelope-vesicle

Finally, the cytosol is defined as containing all the ingredients needed for this process.

The whole reaction then works as follows. By the disassembly of the nucap, we first obtain (where !bud◖◗ is the capsid residue):

nucap ∘ cytosol ➡* cytosol ∘ vRNA ∘ !bud◖◗

Then, the vRNA gets replicated *(a)*, and the cytosol can interact to assemble nucaps *(b)* and produce envelope vesicles *(c)*, obtaining any number of copies n,m,p of the respective components, and some residue:

nucap ∘ cytosol ➡* nucap^n ∘ $\text{envelope-vesicle}^m$ ∘ cytosol ∘ vRNA^p ∘ !bud◖◗

4.7 Protein Complexes

The handling of protein complexes requires more sophistication in the structure of molecules. See for example the κ-calculus [5], for an expressive notation for molecular complexes that

includes state parameters and binding constructs, and that can realistically model protein interaction networks. Our bind&release mechanism and the rewrites of κ-calculus should mutually generalize; we think this is a promising direction for combining complexation with membrane operations. Here we just describe a simple extension of our framework, by adding complex formation, $m_1{:}m_2$, between simple molecules:

Molecular Complexes

Systems	$P,Q ::= \ldots \mid c$	systems extended with complexes c
Complexes	$c,d ::= m \mid c{:}d$	basic molecules $m \in M$, or complexation
	$p,q ::= c_1 \circ \ldots \circ c_k$	multisets of complexes
Actions	$a,b ::= \ldots \mid p_1(p_2) \rightleftarrows q_1(q_2)$	bind&release of complexes
B&R	$p_1 \circ p_1(p_2) \rightleftarrows q_1(q_2).\sigma \mid \sigma_0 \langle\!\langle p_2 \circ P \rangle\!\rangle \rightarrow q_1 \circ \sigma \mid \sigma_0 \langle\!\langle q_2 \circ P \rangle\!\rangle$	

Then, we can use the bind&release operator to express, e.g. complexation on the inside surface of a membrane:

$$\diamond(m_1 \circ m_2) \rightleftarrows \diamond(m_1{:}m_2)$$

Protein synthesis in the E.R. has the following structure: membrane bound ribosomes take amino acids (bound to tRNA) from one side of the membrane, and produce complexes (polypeptides) on the other side of the membrane. Hence, decomplexation, membrane-crossing, and complexation are combined in a single process. A completely satisfactory description of this process, though, probably requires either restriction [5], to model the identity of the polypeptide being assembled, or some further notions of complexation with membrane-bound proteins.

5 Extensions

In this section we discuss possible extensions that fit well into the Brane Calculi framework.

5.1 Communication

Although much can be done with purely combinatorial operators, as in the Basic Brane Calculus considered so far, it is possible to add communication operations in the style of CCS or BioAmbients, assuming a substitution $\tau\{p \leftarrow m\}$ of name m for name p in τ.

On-Membrane Communication (CCS style)

Actions	$a,b ::= \ldots \mid p2p_n(m) \mid p2p^{\perp}_n(m)$
peer to peer	$p2p_n(m).\sigma \mid p2p^{\perp}_n(p).\tau \mid \rho\langle\!\langle P \rangle\!\rangle \rightarrow \sigma \mid \tau\{p \leftarrow m\} \mid \rho\langle\!\langle P \rangle\!\rangle$

Cross-Membrane Communication (BioAmbients style)

Actions	$a,b ::= \ldots \mid s2s_n(m) \mid s2s^{\perp}_n(m) \mid p2c_n(m) \mid p2c^{\perp}_n(m) \mid c2p_n(m) \mid c2p^{\perp}_n(m)$
sibling to sibling	$s2s_n(m).\sigma \mid \sigma_0\langle\!\langle P \rangle\!\rangle \circ s2s^{\perp}_n(p).\tau \mid \tau_0\langle\!\langle Q \rangle\!\rangle \rightarrow \sigma \mid \sigma_0\langle\!\langle P \rangle\!\rangle \circ \tau\{p \leftarrow m\} \mid \tau_0\langle\!\langle Q \rangle\!\rangle$
parent to child	$p2c_n(m).\sigma \mid \sigma_0\langle\!\langle p2c^{\perp}_n(p).\tau \mid \tau_0\langle\!\langle Q \rangle\!\rangle \circ P \rangle\!\rangle \rightarrow \sigma \mid \sigma_0\langle\!\langle \tau\{p \leftarrow m\} \mid \tau_0\langle\!\langle Q \rangle\!\rangle \circ P \rangle\!\rangle$

child to parent	$c2p^{\perp}_n(p).\tau \mid \tau_0 ⦇ c2p_n(m).\sigma \mid \sigma_0 ⦇ Q ⦈ \circ P ⦈ \rightarrow \tau\{p \leftarrow m\} \mid \tau_0 ⦇ \sigma \mid \sigma_0 ⦇ Q ⦈ \circ P ⦈$

5.2 Choice

A choice operation can be added to membranes:

Choice

Branes	$\sigma,\tau ::= \ldots \mid \sigma+\tau$

Its main impact is that all reaction rules must then consider more complex normal forms for membranes, of the form $(a.\sigma+\sigma_1) \mid \sigma_0 ⦇ P ⦈$ instead of $a.\sigma \mid \sigma_0 ⦇ P ⦈$. There may be ways to hide this complexity behind appropriate notation, particularly in absence of binding operators.

A good use for choice is to express a shuffle operator $a.b.\sigma+b.a.\sigma$, which is natural when considering individual proteins triggered by two independent binding sites. On the other hand, common forms of choice can be embedded directly in the notation for molecules [5]. Choice at the system level, instead of the membrane level, does not seem very realistic.

Exercise: define (without using choice) a pair of *isolation* actions isl_n, $isl^{\perp}_n$, such that:

$$isl^{\perp}_n.\sigma \mid \sigma_0 ⦇ P ⦈ \circ isl_n.\tau \mid \tau_0 ⦇ Q ⦈ \rightarrow^* \sigma ⦇ \diamond ⦈ \circ \tau \mid \tau_0 ⦇ 0 ⦇ \sigma_0 ⦇ P ⦈ ⦈ \circ Q ⦈$$

that is, $isl^{\perp}_n.\sigma$, when triggered by its co-action, isolates $\sigma ⦇ \diamond ⦈$ as the only residual, and makes the rest of its membrane and its contents inaccessible. Then, use a pair of isolation actions in parallel to implement a limited form of choice.

5.3 Atonal Transport

Although we have emphasized bitonal operators, there are situations in which simple in-out transport operators, as in BioAmbients [14], may be preferable. One example is when representing a protein with multiple interaction domains as a (fictitious) membrane (see [14] for a detailed discussion). When a protein is represented that way, protein transport in/out of a (real) membrane takes the form of *atonal operations* (ones that do not preserve bitonality). Atonal situations may also arise at higher levels of organization, as when a cell enters the bloodstream through a vessel wall.

The following transport operations are similar to the ones in BioAmbients:

Actions	$a,b ::= \ldots \mid in_n \mid in^{\perp}_n \mid out_n \mid out^{\perp}_n$
In	$in_n.\sigma \mid \sigma_0 ⦇ P ⦈ \circ in^{\perp}_n.\tau \mid \tau_0 ⦇ Q ⦈ \rightarrow \tau \mid \tau_0 ⦇ \sigma \mid \sigma_0 ⦇ P ⦈ \circ Q ⦈$
Out	$out^{\perp}_n.\tau \mid \tau_0 ⦇ out_n.\sigma \mid \sigma_0 ⦇ P ⦈ \circ Q ⦈ \rightarrow \sigma \mid \sigma_0 ⦇ P ⦈ \circ \tau \mid \tau_0 ⦇ Q ⦈$

Alternatively, one can think of adding a single atonal primitive to Phago/Exo/Pino in order to encode In/Out. A simple solution is:

$$wrap(\sigma).\tau \mid \tau_0 ⦇ P ⦈ \rightarrow \sigma ⦇ \tau \mid \tau_0 ⦇ P ⦈ ⦈$$

so that wrap + exo = out, and wrap + phago + exo = in.

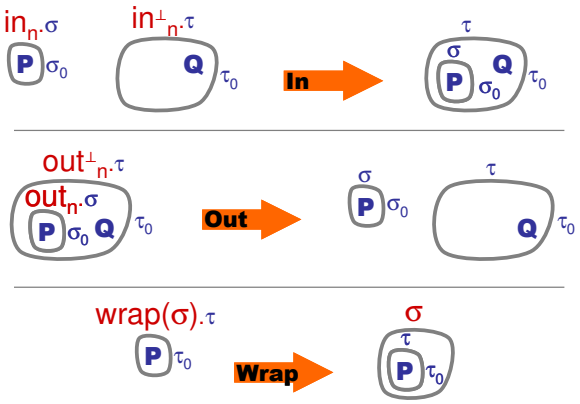

Figure 12 Atonal Reactions

It is conceivable that a simple type system may keep the bitonal and atonal parts of a system separate. It is also conceivable that empty membranes (representing molecules) may harmlessly assume a double tonality, violating bitonality only in a weak sense. This could be achieved by restricting In/Out to the empty membrane case:

SmallIn $\quad in_n.\sigma|\sigma_0\langle\!\langle\diamond\rangle\!\rangle \circ in^{\perp}_n.\tau|\tau_0\langle\!\langle P\rangle\!\rangle \Rightarrow \tau|\tau_0\langle\!\langle\sigma|\sigma_0\langle\!\langle\diamond\rangle\!\rangle \circ P\rangle\!\rangle$

SmallOut $\quad out^{\perp}_n.\tau|\tau_0\langle\!\langle out_n.\sigma|\sigma_0\langle\!\langle\diamond\rangle\!\rangle \circ P\rangle\!\rangle \Rightarrow \sigma|\sigma_0\langle\!\langle\diamond\rangle\!\rangle \circ \tau|\tau_0\langle\!\langle P\rangle\!\rangle$

This way, although $\diamond$ really changes tone in reactions, the systems is consistently bitonal both before and after reactions. Again, a minimal atonal extension could consist of:

SmallWrap $\quad wrap(\sigma).\tau|\tau_0\langle\!\langle\diamond\rangle\!\rangle \Rightarrow \sigma\langle\!\langle\tau|\tau_0\langle\!\langle\diamond\rangle\!\rangle\rangle\!\rangle$

Exercise: show that it is possible to represent small molecules m as empty membranes $mol_m\langle\!\langle\diamond\rangle\!\rangle$, for an appropriate definition of mol_m, in such a way that an operation similar to bind&release of Section 4.1 is definable. Hints: choice is useful; limit the exercise to sequential bind&release of individual molecules, rather than atomic bind&release of multiple molecules.

5.4 Free-Floating Proteins as Membranes

Free-floating proteins are large molecules with complex dynamic behavior and multiple independent domains of interaction: they can interact with membranes and with each other, and can act as catalysts for smaller molecules. In section 4.7 we have discussed how to model protein complexes directly. It may also seem reasonable to model such large molecules as "small membranes", that is, as membranes $\sigma\langle\!\langle\rangle\!\rangle$ with multiple surface actions but (normally) empty contents.

In this view, a free floating protein inside a membrane is just a membrane inside a larger membrane. This idea and the issues it raises are discussed in [14]. (A different proposal is to assume multi-domain molecules as primitive [4][5].)

One problem with representing molecules as membranes, in general, is that molecules can "squeeze through" membranes or through their channels, while membranes cannot. Situations where large molecules cross membranes are however, limited, and can sometimes be modeled by other mechanisms. One common case is when proteins and RNA cross the nuclear membrane through its pores. The nuclear membrane is a double membrane, so

crossing it can be modeled bitonally by Phago and Exo through the lumen. (This is slightly artificial, but an accurate geometrical modeling of the nuclear double membrane and its toroidal pores would in any case require a 3D calculus.)

The problem of complex formation and breaking ([14], Section 3.2) also has a bitonal solution. Assuming proteins are represented as empty membranes $\sigma\text{◖◗}$, $\tau\text{◖◗}$ with all their domains on σ and τ, then complexation is simply merging of two such membranes, $\sigma|\tau\text{◖◗}$ (modulo some interaction). Breakup can be achieved by Pino, to recreate internally the protein fragments, $\rho\text{◖}\sigma_1\text{◖◗}\circ\tau_1\text{◖◗◗}$ followed by Bud to separate them, $\rho_1\text{◖}\sigma_2\text{◖◗◗}\circ\rho_2\text{◖}\tau_2\text{◖◗◗}$, and finally by two Exo, $\sigma\text{◖◗} \circ \tau\text{◖◗}$.

Enzyme interactions ([14], Section 3.3) also have a bitonal solution for enzymes reacting with proteins (as opposed to small molecules). Two proteins $\sigma\text{◖◗}$,$\tau\text{◖◗}$ can bind to an enzyme $\rho\text{◖◗}$ by Phago, $\rho\text{◖}\rho_1\text{◖}\sigma\text{◖◗◗}\circ\rho_2\text{◖}\tau\text{◖◗◗◗}$, followed by Mate to bring them in contact, $\rho\text{◖}\rho_3\text{◖}\sigma\text{◖◗}\circ\tau\text{◖◗◗◗}$, followed by their interaction, e.g. again Mate, $\rho\text{◖}\rho_3\text{◖}\sigma|\tau\text{◖◗◗◗}$, followed by Exo to release the catalyzed product, $\sigma|\tau\text{◖◗} \circ \rho\text{◖◗}$. However, the production of enzymes has to be modeled as the production of membranes, not of molecules, and this might be awkward.

5.5 Bitonal Brane Calculi

While the operations of the Basic Brane Calculus are bitonal in nature (i.e. they preserve the nesting parity of subsystems, with the exception of molecules in bind&release), the calculus framework does not build-in bitonality.

A proper Bitonal Brane Calculus would, instead, adopt a syntax of alternating colored brackets $\sigma_1\text{◖}\sigma_2\text{◐}\sigma_3\text{◖}\sigma_2\text{◐}\ldots\text{◑◗◗}$, with an assumption that the tone-dual of a reaction is also a reaction. (This could also be achieved by type distinction, instead of syntactic distinctions.) All the figures resulting from such a calculus could be consistently shaded in two alternating tones, and atonal operations like In, Out, or Wrap could not be directly supported because they would violate the alternation.

Exercise: show that a bitonal calculus (with Phago+Exo+Pino and alternating brackets) can emulate the atonal calculus (with Phago+Exo+Pino+Wrap). Hint: double walling.

6 Encoding Brane Calculi

Are Brane Calculi really novel, of can they be easily encoded in other calculi? The obvious comparison is with the closely related BioAmbients Calculus. Let us consider the simplest possible idea for a translation P† into BioAmbients, namely "in brane" actions (Figure 13):

$$\sigma\text{◖}P\text{◗}\dagger \triangleq [\sigma\dagger \mid P\dagger]$$

Where the membrane σ is converted into a process inside a membrane […], at the same level as the translation of P. Consider now the induced translation of Exo:

Exo $\quad \circlearrowright^{\perp}_{n}.\tau|\tau_0\text{◖}\circlearrowright_n.\sigma|\sigma_0\text{◖}P\text{◗}\circ Q\text{◗} \rightarrow P \circ \sigma|\sigma_0|\tau|\tau_0\text{◖}Q\text{◗}$

Exo† $\quad [\circlearrowright^{\perp}.\tau\dagger \mid \tau_0\dagger \mid [\circlearrowright.\sigma\dagger \mid \sigma_0\dagger \mid P\dagger] \mid Q\dagger] \rightarrow P\dagger \mid [\sigma\dagger \mid \sigma_0\dagger \mid \tau\dagger \mid \tau_0\dagger \mid Q\dagger]$

Where we would have to devise an appropriate definition for $\circlearrowright^{\perp}.\tau\dagger$ and $\circlearrowright.\sigma\dagger$ so that Exo† had the prescribed behavior. A problem, though, is already apparent. The Exo rule separates P from $\sigma|\sigma_0$ on the r.h.s., and it can do so because the separation between $\sigma|\sigma_0$ and P is built into the term $\circlearrowright_n.\sigma|\sigma_0\text{◖}P\text{◗}$ on the l.h.s.. In Exo†, though, the process $\circlearrowright.\sigma\dagger \mid \sigma_0\dagger \mid P\dagger$ on the l.h.s. is a featureless composition; how does the rule "know" to split off P† precisely at that position?

To avoid this loss of structure, it is necessary to put more structure in the translation:

$\sigma(\!|P|\!)\dagger \triangleq [[\sigma\dagger] \mid P\dagger]$ or:

$\sigma(\!|P|\!)\dagger \triangleq [\sigma\dagger \mid [P\dagger]]$ "Ball bearing" encoding

This requires more complicated encodings of operations, which need to cross multiple level of brackets and therefore have atomicity problems.

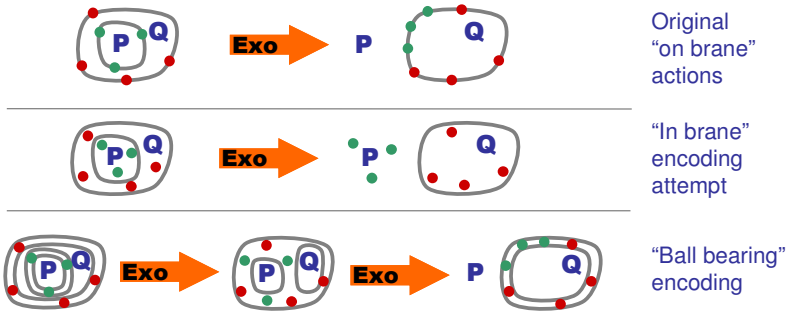

Figure 13 Exo Encodings

Our suspicion is that an encoding of Brane Calculi in Ambients-like calculi may be possible, but it is not easy and almost certainly not practically usable.

7 Conclusions

How are "bio"-calculi different from other process calculi? Both in Brane Calculi and in BioAmbients, (and in BioSPI [13], before that), we have used standard concepts and techniques developed for calculi of concurrency and mobility. We believe that Brane Calculi are beginning to confront some of the pragmatic issues discovered with BioAmbients, by emulating more closely biological processes, in the same way that BioAmbients removed the need for some artificial encodings in BioSPI.

The issue of choosing "realistic" primitives is a tricky one. At one extreme, only the precise mechanisms that have an existing biological implementation are realistic, and those usually have extremely sophisticated and still only partially understood molecular-level implementations. However, even without understanding the molecular details, it is possible to distinguish operations that work via dedicated molecular machinery from those that do not. At the other extreme, biological systems have general constraints and invariants that determine which operations are at least in principle realistic (and which are not). Membrane orientation is one such invariant: it is actively maintained by living cells by consistently orienting proteins on the membrane surface. Bitonality is another invariant, at least in some regimes of operation; it derives from certain transformations of oriented membranes that produce deeper nestings: the basic bitonal structure of a cell and its organs is due to such transformations that happened during evolution ([1] p. 556). These biological invariants suggest a different set of "potentially realistic" basic operations for concurrent calculi than ones that had been considered before.

Another basic aspect of biological membranes is their nature as a two-dimensional fluid embedded in a three-dimensional fluid; this is in fact more fundamental than any orientability or bitonality considerations. This means that there are at least two commutative monoids involved, and not the single one usually seen in process calculi. The formalization of these

two monoids adds complexity, but supports the notion of computation *on* the membrane, that is, of computation that is directly aware of conditions on both sides of the membrane. Trying to emulate this fluid-in-fluid structure by other encodings is awkward (see Section 6), although the issue has been valiantly confronted in BioAmbients.

References

[1] B.Alberts, D.Bray, J.Lewis, M.Raff, K.Roberts, J.D.Watson. **Molecular Biology of the Cell**. Third Edition, Garland.

[2] L.Cardelli. **Bitonal Membrane Systems – Interactions of Biological Membranes**. To appear.

[3] L.Cardelli and A.D.Gordon. **Mobile Ambients.** Theoretical Computer Science, Special Issue on Coordination, D. Le Métayer Editor. Vol 240/1, June 2000. pp 177-213.

[4] V.Danos, M.Chiaverini. **A Core Modeling Language for the Working Molecular Biologist**. 2002.

[5] V.Danos and C.Laneve. **Formal Molecular Biology**. Theoretical Computer Science, to Appear.

[6] L.H.Hartwell, J.J.Hopfield , S.Leibler , A.W.Murray: **From molecular to modular cell biology.** Nature. 1999 Dec 2;402(6761 Suppl):C47-52.

[7] H. Kitano: **A graphical notation for biochemical networks**. BIOSILICO 1:169-176, 2003.

[8] K.W. Kohn: **Molecular Interaction Map of the Mammalian Cell Cycle Control and DNA Repair Systems.** Molecular Biology of the Cell, 10(8):2703-34, Aug 1999.

[9] H.Lodish, A.Berk, S.L.Zipursky, P.Matsudaira, D.Baltimore, J.Darnell. **Molecular Cell Biology**. Fourth Edition, Freeman.

[10] H.H.McAdams, A.Arkin : **It's a noisy business! Genetic regulation at the nanomolar scale**. Trends Genet. 1999 Feb;15(2):65-9.

[11] R.Milner. **Communicating and Mobile Systems: The π-Calculus**. Cambridge University Press, 1999.

[12] C.Priami. **The Stochastic pi-calculus**. The Computer Journal 38: 578-589, 1995.

[13] C.Priami, A.Regev, E.Shapiro, and W.Silverman. **Application of a stochastic name-passing calculus to representation and simulation of molecular processes**. Information Processing Letters, 80:25-31, 2001.

[14] A.Regev, E.M.Panina, W.Silverman, L.Cardelli, E.Shapiro. **BioAmbients: An Abstraction for Biological Compartments**. Theoretical Computer Science, to Appear.

[15] **Systems Biology Markup Language.** <www.sbml.org>