
**Technical Report CoSBi 08/2007**

# Stochastic $\pi$-calculus modelling of multisite phosphorylation based signaling: in silico analysis of the Pho4 transcription factor and the PHO pathway in Saccharomyces cerevisiae

Nicola Segata
*DISI, University of Trento*
segata@dit.unitn.it

Enrico Blanzieri
*DISI, University of Trento*
blanzier@dit.unitn.it

Corrado Priami
*CoSBi*
and
*DISI, University of Trento*
priami@cosbi.eu

**Abstract**

Multisite phosphorylation is known to be an important and dynamic mechanism for regulating the activity of transcription factors. Here we propose a stochastic $\pi$-calculus modelling approach able to handle the complexity of post-translational modifications and to overcome the limitations of the ordinary differential equations based methods. The model can be applied without *a priori* assumptions to every (multisite) phosphorylation regulation for which some basic rates are known or can be indirectly set with experimental data. We apply it to the multisite phosphorylation of the *Pho4* transcription factor that plays a crucial role in the phosphate starvation signalling in *Saccharomyces cerevisiae* using available *in vitro* experiments for the model tuning and validation. The innovative modelling of the sub-path with the stochastic $\pi$-calculus allows quantitative analyses of the kinetic characteristics of the *Pho4* phosphorylation, the different phosphorylation dynamics for each site (possibly combined) and the variation of the kinase activity as the reaction goes to completion. One of the performed predictions indicates that the *Pho80-Pho85* kinase activity on the *Pho4* substrate is nearly distributive and not semi-processive as previously found analysing only the phosphoform concentrations *in vitro*. This result is obtained because the model can consider and quantify the binding events without phosphorylations that cannot be experimentally measured. Thanks to the compositionality property of process algebras, we also developed the whole PHO pathway model that gives new suggestions and confirmations about its general behaviour. The potentialities of process calculi based *in silico* simulations for biological systems are highlighted and discussed.

# Introduction

Protein activity is regulated by a large set of post-translational modifications (PTMs) that very often act cooperatively on different domains in order to increase combinatorially the possible states and behaviours of the target protein [1][2][3]. Among all the possible PTMs, phosphorylation is very common and the multisite phosphorylation is a frequent variant that enhances expressivity and non-linearity of protein responses [4][5][6]. The understanding of this kind of regulations is crucial in many pathways, but experimental analyses of multisite phosphorylation are not simple even if some specific mass spectrometry based techniques were proposed [7][8]. For these reasons the experimental approach is very often coupled with computational models that are mandatory for an in-depth understanding of the kinetics of the regulation [9][10][11][12]. In these cases the modelling approach is based on ordinary differential equations (ODE), but it exhibits important limitations. First, some limiting assumptions like the distributive and ordered behaviour of the multisite phosphorylation could be required in order to maintain the computational cost under a reasonable level [4]. Second, the system description is based on the concentrations which are continuous values while the biological interactions are discrete; this approximation cannot be acceptable when the number of interacting entities is small and when modifications not influencing the concentration occur. Third, the ODE are intrinsically deterministic and do not consider noise (unless artificially added like in [13]) and stochasticity. Fourth, the ODE approach cannot handle all the possible analyses at the same time but it is necessary to focus on some partial aspects. So, up to now, neither the *in vitro* approach nor the ODE simulation approach are able to handle in the same experiment all the potentially interesting aspects in a quantitative way.

The approach we adopt here overcomes the ODE limitations and introduces other desirable features. The modelling framework is based on process algebras theory which usage in biology was for the first

time proposed in [14] and [15], reviewed in [16] and applied on small pathways and gene networks for example in [17][18][19][20]. The constructed model must be consistent with general qualitative description as well as quantitative values derived from biological experiments. Once the model is correctly validated, the *in silico* simulations can provide precise predictions of non-measurable biological aspects and new time-continuous analyses and permits to quantify every possible single or multiple step transitions of the systems. It is possible, for example, to consider also the quantification of binding events without phosphorylations that cannot be sensed by biological experiments because the phosphoform concentrations remain unchanged.

We developed with the stochastic $\pi$-calculus (one of the most famous process calculi developed originally by [21]) the *Pho4* multisite phosphorylation model. The multi-domain phosphorylation of the *Pho4* transcription factor by the *Pho80-Pho85* cyclin-CDK kinase complex [22][23][24] is the key event of the phosphate starvation signalling in *Saccharomyces cerevisiae* (for a review see [25][26][27][28] or [29]). As shown for example in [30] and [31], the profiles produced by the phosphorylation of four of the five *Pho4* phosphorylation sites are associated with a precise and different behaviour of the transcription factor, favouring its extranuclear exportation, preventing its reimportation or inhibiting the cooperative binding with the *Pho2* transcription factor [32] on the promoters of *Pho5* [33] and *Pho81* [34][35] genes. This partially redundant and overlapping regulation mechanism [36] makes crucial, in order to understand precisely the *Pho4* behaviour in different conditions, the quantitative analysis of the 32 different phosphoform dynamics during the kinase reaction. This is relevant in analysing the average number of phosphorylations for binding event and, more generally, in order to quantitatively describe overall behaviours of biological aspects.

The validation of the *Pho4* multisite phosphorylation model is performed with respect to the three datasets of biological *in vitro* experiments taken from [9] that are the only quantitative information available for the *Pho4* phosphorylation sub-path; the first dataset contains the site preferences quantification; the second one consists in the phosphoform concentration variations; and the third one collects the quantification of the phosphoform concentration at the beginning of the reaction from which it is possible to extract the average number of phosphorylations for binding event not including the bindings without phosphorylations.

Thanks to the compositionality property of process algebras not available in the ODE approach, we also extended the *Pho4* validated model to the whole phosphate condition signaling mechanism (extensively studied by the O'Shea lab at Harvard and for which a preeminent review is [25]) obtaining the PHO pathway model. This model can be only qualitatively validated because of the unavailability, at the best of our knowledge, of precise experimental data but permits however to perform predictive analyses and observations about the overall behaviour of the pathway. We also verified the potentiality of the model to be predictive with respect to microarray data.

Our general contribution regarding the phosphorylation PTM consists in a computational model able to handle every phosphorylation reaction with a variable number of phosphorylation sites. If the available biological values are precise enough to set properly and with a reasonable level of confidence the rates of the model, it has the potentialities to analyse and quantify complex and combined aspects that are usually not easily measurable or predictable.

The biological contributions of this work represent the first biological predictions obtained with process calculi based approaches (and in particular with the stochastic $\pi$-calculus) and are relative to two aspects. a) On the *Pho4* multisite phosphorylation mechanism, we provide a model based estimation of the average number of *Pho4* phosphorylations per *Pho80-Pho85* binding event with its variation during the reaction. This shows that the phosphorylation is more likely to be distributive than processive. Moreover, the *Pho4* site specific phosphorylation dynamics suggests quantitative hypotheses on the *Pho4* behaviour. b) On the whole PHO pathway we give some qualitative confirmations about the general mechanisms (like the importance of the *Pho81* feedback loop) and propose an analysis about the percentage of *Pho81* proteins active as inhibitors needed to signal the starvation and the intermediate phosphate conditions.

At the best of our knowledge this is the first time that a process algebras based approach leads to concrete biological predictions.

The work is organized as follows: after introducing the stochastic $\pi$-calculus formalism and the biological background about the PHO pathway we present the sub-models used to describe the each sigle part of the phosphate starvation signalling. We then analyse the adaptation of the multisite phosphorylation model to the *Pho4* phosphorylation obtaining the *Pho4* multisite phosphorylation model, and the integration of all the sub-models resulting in the PHO pathway model. Finally, we expose the biological results reached with the *in silico* experimental approach and discuss the general outcomes of our work.

# The stochastic $\pi$-calculus

The $\pi$-calculus is a process algebra originally developed by Milner, Parrow and Walker in 1989 ([37]) starting from CCS, for describing concurrent computations whose configuration may change during the computation. Central to the $\pi$-calculus is the notion of *name* because in the calculus the names play a dual rule of communication channels and variables. The basic example regards the transfer of a name among a channel between two processes running in parallel; the receiver can use the name as a channel for dynamic further interactions with other processes in the system.

There are a lot of different specification of the $\pi$-calculus, here for the non stochastic version the variant exposed is based on [38] while for the stochastic version, developed by [21], it is exposed the specifications on which the $SPiM$ simulator is based [39].

**Syntax.** The syntax of the $\pi$-calculus processes is presented in table 1. The intuitive behaviors of the

| **Prefixes** | | | | | **Processes** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\pi$ | ::= | $x\langle y\rangle$ | $\mid$ | $x(y)$ | $\mid$ | $\tau$ | $P$ | ::= | $\mathbf{0}$ | $\mid$ $\pi.P$ | $\mid$ $P+P$ | $\mid$ $P\mid P$ | $\mid$ $!\,P$ | $\mid$ $\nu x\,P$ |

**Table 1:** The syntax of the $\pi$-calculus.

process forms listed in the table are the following (integrating together prefixes and processes).

- $x\langle y\rangle.P$ - **the output prefix**: the name $y$ is sent along the name (channel) $x$. After the output is performed (for this the dual input prefix is required), the execution continues with the process $P$.
- $x(y).P$ - **the input prefix**: the name $y$ is received along the name (channel) $x$. After the input is performed (for this the dual output prefix is required), the execution continues with the process $P$.
- $\tau.P$ - **the silent prefix**: The execution evolves to P without interacting with the environment performing only some internal and not specified actions.
- $\mathbf{0}$ - **the null process**: is a process that does nothing or whose execution is complete and has stopped.
- $P+Q$ - **the non deterministic choice**: only one of the two branches can be executed. The activation of one branch inhibits the possible execution of the other one.
- $P|Q$ - **the parallel composition**: the $P$ and $Q$ processes are independently executed. However they can interact performing communication on a shared channel.
- $!\,P$ - **the replication**: multiple (infinite) copies of the same process are defined.

| Process $P$ | Names $\mathsf{n}$(P) | Bound Names $\mathsf{bn}$(P) | Free Names $\mathsf{fn}$(P) |
|---|---|---|---|
| $\mathbf{0}$ | $\emptyset$ | $\emptyset$ | $\emptyset$ |
| $x\langle y\rangle.R$ | $\{x\} \cup \{y\} \cup \mathsf{n}(R)$ | $\mathsf{bn}(R)$ | $\{x\} \cup \{\, y\} \cup \mathsf{fn}(R)$ |
| $x(y).R$ | $\{x\} \cup \{y\} \cup \mathsf{n}(R)$ | $\{y\} \cup \mathsf{bn}(R)$ | $\{x\} \cup \mathsf{fn}(R)$ |
| $\tau.R$ | $\mathsf{n}(R)$ | $\mathsf{bn}(R)$ | $\mathsf{fn}(R)$ |
| $R \vert Q$ | $\mathsf{n}(R) \cup \mathsf{n}(Q)$ | $\mathsf{bn}(R) \cup \mathsf{bn}(Q)$ | $\mathsf{fn}(R) \cup \mathsf{fn}(Q)$ |
| $R + Q$ | $\mathsf{n}(R) \cup \mathsf{n}(Q)$ | $\mathsf{bn}(R) \cup \mathsf{bn}(Q)$ | $\mathsf{fn}(R) \cup \mathsf{fn}(Q)$ |
| $!\, R$ | $\mathsf{n}(R)$ | $\mathsf{bn}(R)$ | $\mathsf{fn}(R)$ |
| $\nu x\, R$ | $\{x\} \cup \mathsf{n}(R)$ | $\{x\} \cup \mathsf{bn}(R)$ | $\mathsf{fn}(\pi.\, R) \setminus \{x\}$ |

**Table 2:** The free and the bound names of the $\pi$-calculus.

- $\nu x\, P$ - **the restriction**: the channel name scope is restricted to the process $P$. It can be seen as the creation of a new name $x$ within $P$.

**Structural congruence.** The grammar of the $\pi$-calculus can make distinction between processes that intuitively are the same. For example is natural to consider that $P|Q$ and $Q|P$ have the same behaviour reasoning that the parallel composition is unordered, but following the syntax definition the two process are not equals.

So the structural congruence concept must be introduced to define in a rigorous way the processes that are structurally equivalent. Notice that the semantics of the calculus is not yet introduced and so the congruence is only structural.

Before giving the congruence rules, it is necessary to introduce the concept and the definitions of the free and bound names of the calculus. A name $x$ (that is both a channel and a variable) is bound on the process $P$ and so $x \in \mathsf{bn}(P)$ if $x$ has a bound occurrence in $P$, while the free names, $\mathsf{fn}(P)$ have no bound occurrences in $P$. The free and bound names definition for every syntactic rule are listed in table 2.

Now it is possible to define the structural congruence, shown in table 3.

1. If $P$ and $Q$ are identical under $\alpha$-conversions then $P \equiv Q$
2. $(\mathcal{P}/_{\equiv}, |, \mathsf{nil})$ are $(\mathcal{P}/_{\equiv}, +, \mathsf{nil})$ commutative monoids
3. The unfolding law: $!\, P \equiv P \mid !\, P$
4. Scope restriction laws:

$$\begin{array}{rcll} \nu x\ \mathsf{nil} & \equiv & \mathsf{nil} & \\ \nu x\,(P \mid Q) & \equiv & P \mid \nu x\, Q & \text{if } x \notin \mathsf{fn}(P) \\ \nu x\,(P + Q) & \equiv & P + \nu x\, Q & \text{if } x \notin \mathsf{fn}(P) \\ \nu x\, \nu y\, P & \equiv & \nu y\, \nu x\, P & \end{array}$$

**Table 3:** Rules of structural congruence for the $\pi$-calculus.

Very briefly the rules can be explained as follows:

1. A process $P$ is identical under $\alpha - \mathit{conversion}$ to process $Q$ if they differ only in the choice of the bound names. For example $a(x).b\langle x\rangle$ is identical under $\alpha$-conversion to $a(y).b\langle y\rangle$ and so they are congruent.

2. Parallel composition and non deterministic choice are commutative and associative operators for the class of processes defined by the grammar and have $\mathbf{0}$ as null element.

3. The unfolding law shows that the number of possible processes of the replication operator is infinite.

4. The restriction is used in order to declare a name private to a process or a sub-process. These rules formalize the intuition that the restriction regarding a process P can be everywhere if $P$ is under its scope and it cover the same occurrences of the name.

**Operational semantics.** For the semantics of $\pi$-calculus also the substitution is needed; $P\{z/y\}$ denotes the process $P$ in which the free name $z$ has been substituted by the free name $y$. Note that this substitution may involve $\alpha$-conversion to avoid possible name conflicts.

The operational semantics is given through a labelled transition system where the transitions are of the kind $P \xrightarrow{\alpha} P'$ for some set of actions ranged over $\alpha$ that can be an input action $a(x)$, an output action $a\langle x\rangle$, or a silent-action $\tau$ corresponding to an internal communication. In the reduction rules we write $P \xrightarrow{\alpha} P'$ if $P$ can perform a computation step consuming $\alpha$, following which it is now $P'$. The reduction relation $\xrightarrow{\alpha}$ is defined as the least relation closed under the set of reduction rules shown in table 4.

$$\text{(prefix)}\ \frac{}{\alpha.\,P \xrightarrow{\alpha} P} \qquad \text{(sum)}\ \frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'}$$

$$\text{(par)}\ \frac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q}\ \mathsf{bn}(\alpha) \cap \mathsf{fn}(Q) = \emptyset \qquad \text{(com)}\ \frac{P \xrightarrow{x(z)} P',\ Q \xrightarrow{x\langle y\rangle} Q'}{P \mid Q \xrightarrow{\tau} P'\{y/z\} \mid Q'}$$

$$\text{(res)}\ \frac{P \xrightarrow{\alpha} P'}{\nu x\, P \xrightarrow{\alpha} \nu x\, P'}\ x \notin \mathsf{n}(\alpha) \qquad \text{(open)}\ \frac{P \xrightarrow{x\langle y\rangle} P'}{\nu y\, P \xrightarrow{x\langle y\rangle} P'}\ x \neq y$$

$$\text{(struct)}\ \frac{P \xrightarrow{\alpha} Q}{P' \xrightarrow{\alpha} Q'}\ P' \equiv P,\ Q' \equiv Q$$

**Table 4:** Operational semantics of $\pi$-calcolo.

The rules can be described informally as follows:

**(prefix)** when the prefix action occurs the prefix is consumed and the execution will proceed with $P$;

**(sum)** if a process inserted in a non deterministic choice structure can evolve, it evolves and the other processes of the non deterministic choice are eliminated;

**(par)** if a process $P$ inserted in a parallel composition structure with another process $Q$ can evolve in $P'$ consuming an action $\alpha$, the resulting process consists in the parallel composition of $P'$ and $Q$. It is possible only if the action $\alpha$ does not have correspondence in $Q$, formally if the intersection between the bound names of $\alpha$ and the free names of $Q$ is empty;

**(com)** this is the main reduction rule which captures the ability of processes to communicate through channels. It requires that two different processes $P$ and $Q$ are in parallel and ready to perform an input $x(z)$ and an output $x\langle y\rangle$ on the same channel $x$ resulting respectively in $P'$ and $Q'$. In this conditions the passage of $y$ trough $x$ can occur resulting in the parallel composition of the evolved processes $P'$ and $Q'$ substituting the place-holder name $z$ with the communicated name $y$;

**(res)** if a process $P$ can evolve to $P'$ consuming $\alpha$ the process $P$ with restriction can evolve to $P'$ with the same restriction if the restricted name is not contained in the name of $\alpha$;

**(open)** if a process $P$ can evolve to $P'$ consuming an output on name $x$, the process consisting in a restriction on $y$ in $P$ evolves in $P'$ without restriction if the restriction target is different from the channel on which the output is performed;

**(struct)** this rule is based on the congruence concept and states that processes that are structurally congruent have the same reductions.

**The stochastic $\pi$-calculus and the SPiM simulator.** The Stochastic Pi Machine (SPiM), is free a simulator for stochastic $\pi$-calculus developed by [40]. It is based on a stochastic version of $\pi$-calculus (called SPi) that is reported in [39]; Xthe grammar is shown in the table 5. In SPi each channel $x$ is

| **Prefixes** | | | **Summation** | | | **Processes** | | |
|---|---|---|---|---|---|---|---|---|
| $\pi$ | ::= | $x\langle y\rangle \mid x(y)$ | $\Sigma$ | ::= | $\mathbf{0} \mid \pi.P + \Sigma$ | $P$ | ::= | $P\|P \mid \Sigma \mid !\pi.P \mid \nu x\, P$ |

**Table 5:** The syntax of the SPi.

associated with a corresponding reaction rate given by $rate(x)$. In the syntax the replication is guarded, i.e. the process to replicate starts with a prefix; also the branches of the non deterministic choice are guarded.

The congruence rules of SPi are shown in table 6 while the reduction rules of the operational semantics are labelled with a reaction rate and are listed in table 7.

$$
\begin{array}{rcl}
P|\mathbf{0} & \equiv & P \\
P|(Q|R) & \equiv & (P|Q)|R \\
\nu x\,\mathbf{0} & \equiv & \mathbf{0} \\
\pi.P + \pi'.P' + \Sigma & \equiv & \pi'.P' + \pi.P + \Sigma \\
x \notin \mathsf{fn}(P) \Rightarrow \nu x\,(P|Q) & \equiv & P|\nu x\,Q \\
P \equiv P' \Rightarrow P|Q & \equiv & P'|Q \\
P \equiv P' \Rightarrow \pi.P + \sigma & \equiv & \pi.P' + \Sigma
\end{array}
\qquad
\begin{array}{rcl}
P|Q & \equiv & Q|P \\
!\pi.P & \equiv & \pi.(P|!\pi.P) \\
\nu x\,\nu y\,P & \equiv & \nu y\,\nu x\,P \\
\Sigma \equiv \Sigma' \Rightarrow \pi.P + \Sigma & \equiv & \pi.P + \Sigma' \\
P \equiv P' \Rightarrow \nu x\,P & \equiv & \nu x\,P' \\
P \equiv P' \Rightarrow !\pi.P & \equiv & !\pi.P'
\end{array}
$$

**Table 6:** Rules of structural congruence in SPi.

$$
\text{(SPi struct)}\ \frac{Q \equiv P \xrightarrow{r} P' \equiv Q'}{Q \xrightarrow{r} Q'}
\qquad
\text{(SPi res)}\ \frac{P \xrightarrow{r} P'}{\nu x\,P \xrightarrow{r} \nu x\,P'}
$$

$$
\text{(SPi par)}\ \frac{P \xrightarrow{r} P'}{P|Q \xrightarrow{r} P'|Q}
\qquad
\text{(SPi com)}\ \frac{}{x\langle n\rangle.\,P + \Sigma | x(m).\,Q + \Sigma' \xrightarrow{rate(x)} P|Q_{\{n/m\}}}
$$

**Table 7:** Reduction rules in SPi.

The SPiM simulator is based on the notion of channel activity and on the Gillespie algorithm [41] to stochastically select the reaction channels.

## The PHO pathway in Saccharomyces cerevisiae

Inorganic phosphate is essential for all organisms, being required for various biosynthetic pathways, for signal transduction, for biosynthesis of diverse cellular components (nucleic acids, proteins, lipids, ...) and for energy metabolism [42]. When microorganisms like Saccharomyces cerevisiae encounter limiting concentrations of phosphate, they activate complex signal transduction pathways to make optimal use of the scarce availability of this nutrient in the medium. In Saccharomyces cerevisiae this regulatory mechanism is named PHO pathway and, in condition of low phosphate in the medium, it increases the expression of multiple genes involved in the acquisition, uptake and storage of inorganic phosphate. The list of the most important genes that are regulated directly by the PHO pathway includes *Pho5* (an acid phosphatase), *Pho8* (a non-specific repressible alkaline phosphatase), *Pho84* (high affinity $P_i/H^+$ symporter), *Pho89* ($Na/P_i$ cotransporter) and *Pho81* (CDK inhibitor) as demonstrated in [43].

In high-phosphate conditions, cells can uptake ample phosphate from their environment without inducing any phosphate-dependent genes, so the pathway is not induced. In no-phosphate conditions, regardless of the amount of phosphate transporters at the plasma membrane, it is essential to scavenge phosphate from the environment as well and therefore it is necessary to induce all the phosphate-responsive genes.

Very briefly, when the $P_i$ concentration in the medium is low, the so called *minimum domain* of *Pho81* protein is modified (the way it happens is still unknown) resulting in the inhibition of the cyclin-CDK complex *Pho80-Pho85* by *Pho81*. *Pho80-Pho85* complex, in its active state, catalyzes a hyperphosphorylation of *Pho4*, a transcription factor essential for the PHO regulated genes expression. *Pho4* has 5 phosphorylation sites that regulate its nuclear import and export and its affinity with *Pho2*, a coactivator protein that is required to interact with *Pho4* for transcription of a subset of the PHO target genes. These target genes include *Pho5* and *Pho81* which causes a positive feedback loop in the PHO regulation pathway. So phosphate starvation results in the inhibition of the *Pho80-Pho85* complex by *Pho81* and this leads to the accumulation of *Pho4* in the nucleus and its capability to bind to its target promoters cooperatively with *Pho2*.

| PHO subpath | Bibliografy |
|---|---|
| Modification of *pho81* inhibition capability by $P_i$ concentration | [44] [45] |
| *Pho80-Pho85-Pho81* complex: formation and activity | [24] [44] [46] [45] [23] [22] |
| Regulation through multisite phosphorylation of the transcription factor *Pho4* by *Pho80-Pho85-Pho81* complex | [26] [24] [30] [36] [47] [48] [9] [49] |
| Cooperative binding of *Pho4* and *Pho2* at the *Pho5* and *Pho81* promoters | [26] [36] [50] [51] [34] [32] [35] [33] [52] |
| *Pho81* Positive feedback loop | [27] [53] |
| Effects on phosphate transporters of the PHO regulated genes and and on the $P_i$ concentration | [25] [54] [55] |
| Low $P_i$ concentration driven response of the polyP metabolism | [43] [55] |

**Figure 1:** Subdivision of the PHO pathway in distinct parts with relative bibliography. The last two parts are not considered in this work.

In Figure 2 and in Figure 3 the PHO pathway is represented in a schematic way respectively in normal and in phosphate starvation conditions.

An exhaustive discussion on the whole PHO metabolic path can be found in [28], [29] and [27], while

in-depth studies on subparts of the pathway are listed in Figure 1 and discussed separately in the following subparagraphs.

**Modification of *Pho81* inhibition capability by $P_i$ concentration.** The starting point of the whole path is the detection of the $P_i$ conditions. It is certain that this role is performed by *Pho81*; however, very little is known about the mechanism by which *Pho81* "understand" the low phosphate condition and consequently regulates the *Pho80-Pho85* kinase activity. A simple model can be the following: the interaction of *Pho81* and the *Pho80-Pho85* complex in high $P_i$ allows the *Pho81* to associate with the kinase complex without inhibition, while in low $P_i$ conditions the affinity of the interaction is increased or altered so the *Pho81* can inhibit the kinase.

From the observation that *Pho81* forms a stable complex with *Pho80-Pho85* under both high and low phosphate conditions, but it is active as inhibitor only under low phosphate conditions, it is possible to deduce that its inhibitory activity is due to post-transcriptional modifications. A domain for these modifications probably consists in the minimum domain containing 80 amino acids (from aminoacid 645 to 724, the so called *Pho81 minimum domain*), that is necessary and sufficient for *Pho81* inhibition of *Pho80-Pho85* complex in response to phosphate conditions. The *Pho81 minimum domain* resides C terminal to the six ankyrin repeats that appear to have no significant role in the inhibition activity.

The *Pho81* is localized prevalently in the nucleus under both high and low $P_i$, so the activity of the protein is not controlled through regulation of its nuclear localization. Some experiments also demonstrate that the localization of the *Pho81* in the nucleus is dependent on *Pho80* but not on *Pho85* and it is in relation with the fact that in associating to the kinase complex, *Pho81* recognizes and binds only to the *Pho80* cyclin subunit.

A recent study, [45], suggests that *Pho81* is regulated, at least partially, through phosphorylation by *Pho80-Pho85* complex. There are nine sites within *Pho81* that contain the typical consensus sequence for possible phosphorylations by a CDK, and a mutant with all nine sites changed to alanine confirms the influence in the inhibitory activity. In particular four sites generate a significant amount of *Pho5* in low phosphate suggesting that *Pho81* phosphorylation is important for its activity during phosphate starvation.

**The *Pho80-Pho85-Pho81* complex: formation and activity.** The *Pho80-Pho85* cyclin-CDK (cyclin-dependent protein kinase) complex works as a key regulator of the phosphatase system being able to phosphorylate in different ways the transcription factor *Pho4* in response to phosphate conditions.

The *Pho85* protein is a kinase shared between several pathways, having multiple functions and ten cyclin-binding partners [24] one of which is the *Pho80* cyclin. The role of the cyclin subunits is to modify the structure of the enzyme to facilitate the active configuration of the kinase. The *Pho81* can inhibit only the *Pho80-Pho85* kinase activity and not the activity of *Pho85* when it is bound to other cyclins different from *Pho80*. Both *Pho80* and *Pho85* are predominantly localized in the nucleus in all phosphate conditions.

In the formation of the stable *Pho80-Pho85-Pho81*, *Pho81* can associate with the cyclin *Pho80* and the *Pho80-Pho85* complex, but not with *Pho85* alone. Recently it has been shown that also *Pho80* is a substrate of the *Pho85* kinase; in particular *Pho80* seems to be phosphorylated on serine 267 and serine 234. The phosphorylation on serine 267 is considered not crucial for the *Pho80* function, while the phosphorylation on serine 234 seems to be required in maintaining the *Pho5* repression in high phosphate conditions. The phosphorylations can modify the affinity between the proteins of the complex and with *Pho4*.

**Regulation through multisite phosphorylation of the transcription factor *Pho4* by *Pho80-Pho85-Pho81* complex.** The transcription factor *Pho4* is common to all the PHO regulated genes, so it plays a crucial role in the pathway. In low $P_i$ conditions *Pho4* is active and concentrated in

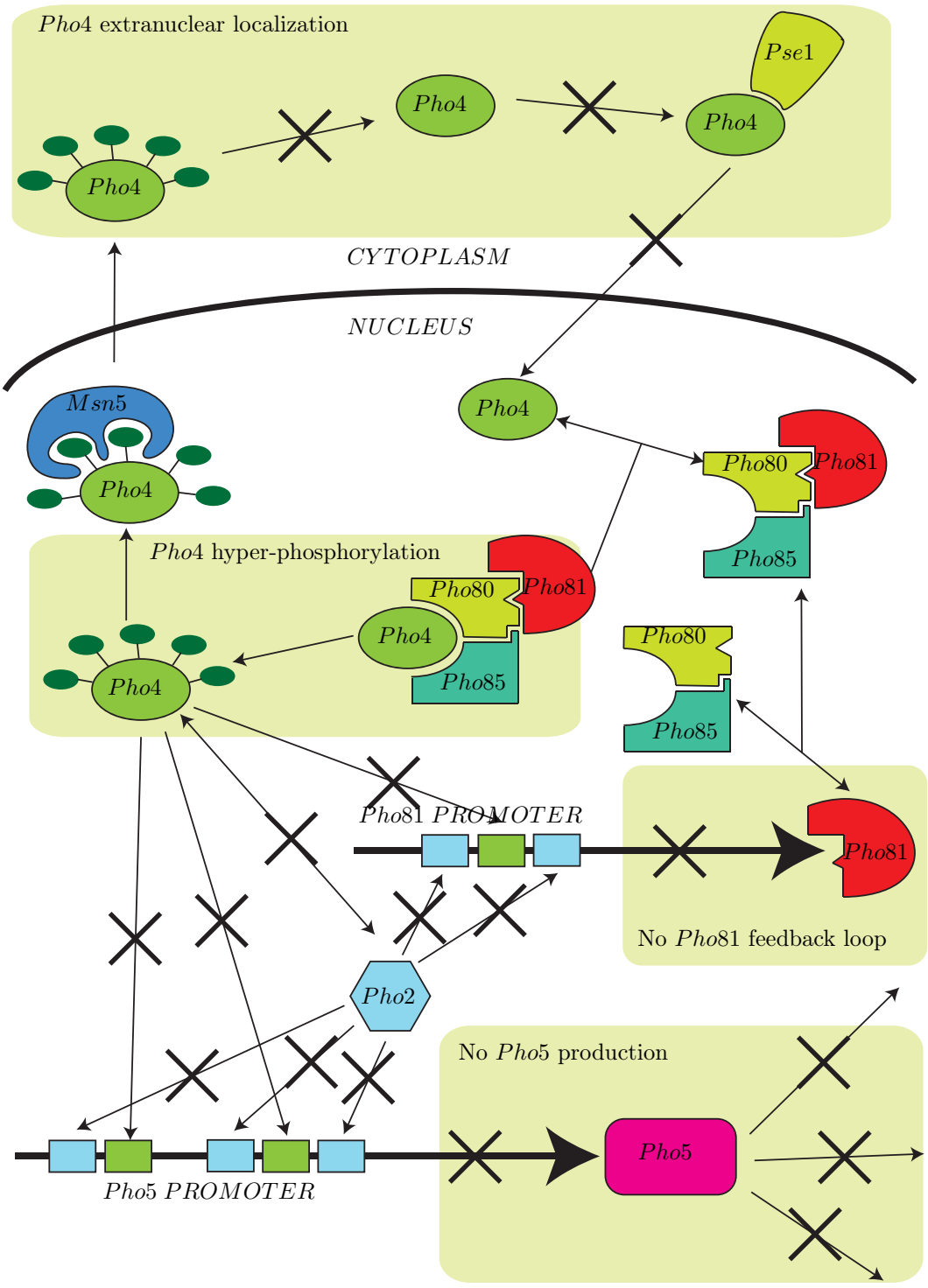



**Figure 2:** Schematic and simplified representation of PHO pathway in high-phosphate conditions.

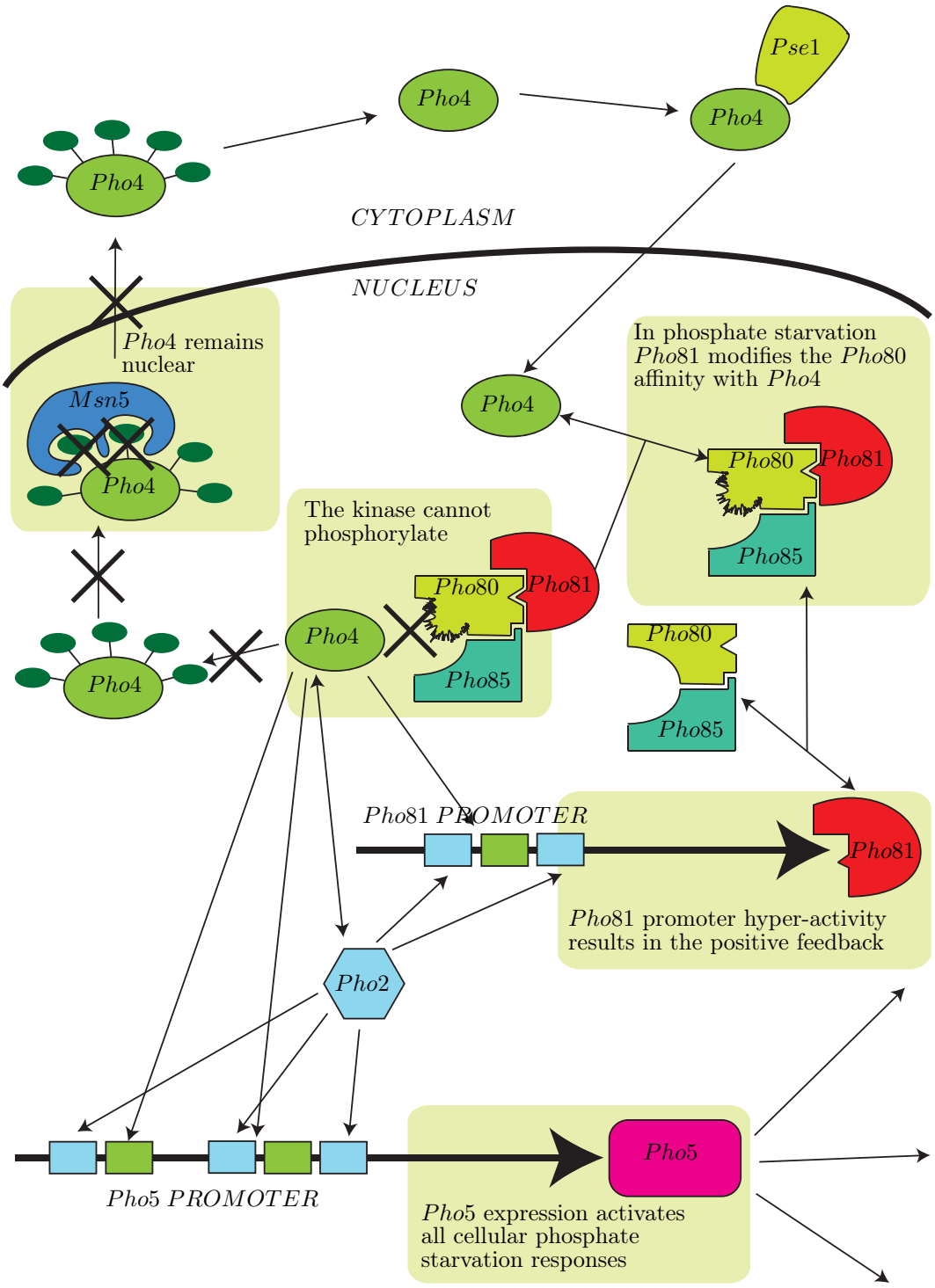



**Figure 3:** Schematic and simplified representation of PHO pathway under phosphate starvation.

the nucleus resulting in hyper-transcription of almost all the PHO-related genes; in contrast, in high $P_i$ conditions *Pho4* is transcriptionally inactivated and predominantly located in the cytoplasm. The behaviour of *Pho4* is regulated through phosphorylation of five sites by *Pho80-Pho85* active complex; in particular each of the five sites play distinct and separable roles in regulating the factor activity. Multiple phosphorylation sites provide also overlapping and partially redundant layers of regulation that efficiently control the activity of the transcription factor; for example, when *Pho4* is localized into the nucleus and phosphorylated on site $sp_6$, it can efficiently bind to the promoter of *Pho84* and activate its transcription, but it does not efficiently bind to *Pho5* promoter to activate the relative acid phosphatase because the cooperative binding with *Pho2* is inhibited. The functions of each phosphorylation site are summarized below:

$sp_1$ the function of this site is unknown but seems not to influence the localization or the activity of *Pho4*;

$sp_2$ **(serine 114)** phosphorylation promotes the rapid export of *Pho4* from the nucleus favouring the interaction with the *Msn5* exporting protein ([49]);

$sp_3$ **(serine 128)** the role of $sp_3$ is the same of $sp_2$;

$sp_4$ **(serine 152)** phosphorylation inhibits the interaction of *Pho4* with the nuclear importing protein *Pse1* ([47],[48]);

$sp_6$ **(serine 223)** phosphorylation prevents the interaction of *Pho4* with its transcriptional co-activator *Pho2*.

In discussing *Pho4* phosphorylation mechanism it is often useful to group the 32 possible phosphorylation patterns in phosphoforms; all the *Pho4* species with only one site phosphorylated (whichever the site is) are denominated *phosphoform 1*, all the *Pho4* species with two sites phosphorylated (whichever the sites are) are denominated *phosphoform 2* and so on until *phosphoform 5* in which *Pho4* is completely phosphorylated. *Phosphoform 1* and *phosphoform 4* consists in 5 different species, *phosphoform 2* and *phosphoform 3* consists in 10 different species while for *phosphoform 5* (and for dephosphorylated *Pho4*) only one combination is possible.

The following two important aspects of the *Pho4* phosphorylation mechanism with simplified *in vitro* experiments has been shown and quantified in [9]:

- the phosphorylation is not processive (it means that for every binding event all the phosphorylation sites of *Pho4* are phosphorylated) but it is also not completely distributive (it means that only one phosphorylation occurs for binding event). So the phosphorylation activity is defined *semi processive* and it means that during the binding, *Pho80-Pho85* can phosphorylate more than one site but not all five sites every time. In particular, the average number of phosphorylations for binding event is estimated in 2.1 at least for the first phases of the reaction, but we claim that this value is overestimated because the binding events without phosphorylations are not considered. In this work the estimation is corrected to 1.45 with the *in silico* approach.

- the phosphorylation mechanism exhibits site preferences. In other words, when the *Pho80-Pho85* is bound to a non-phosphorylated *Pho4* it has different probabilities to phosphorylate the different sites. In particular the preference for $sp_6$ in the non-phosphorylated *Pho4* is 60%, of $sp_4$ is 16% and of $sp_2$ and $sp_3$ together is 24%. Similar preferences are exposed referred to the different phosphoforms. The high preference for $sp_6$ permits a rapid repression of *Pho4-Pho2* dependent genes, while the lower preferences for $sp_2$ and $sp_3$ result in a relatively slower inactivation due to the consequent *Pho4* exportation. It is interesting to notice that $sp_4$ has a higher preference than $sp_2$ and $sp_3$ so the exporting species are prevalently already phosphorylated on $sp_4$, preventing a futile cycle due to the immediate reimportation of the exported species.

Normally in literature the PHO pathway is analysed in phosphate starvation conditions and in high phosphate conditions, but in [36] there is a description of the behaviour in intermediate-phosphate conditions, when a form of *Pho4* preferentially phosphorylated on one of the four sites accumulates and activates the transcription of a subset of phosphate-responsive genes.

The *Pho4* nuclear exporter is *Msn5* while the nuclear importer is *Pse1* and both are members of the $\beta$-importin family of nuclear transport receptors. A part of the *Pho4* case, *Pse1* is normally also an exporting protein and it works interacting with the nuclear pore complex; its role as transporter is well known and the target set of proteins is very large and include for example *Pdr1*, *Yap1*, *Ste12* and *Aft1*. Also *Msn5* is normally involved in nuclear import and export and it has been shown to be responsible for nuclear importation of replication protein *A* and for export of several proteins including *Swi6* and *Far1*; the cargo dissociation involves binding to *RanGTP*.

From literature we know the mechanisms by which *Pho4* is phosphorylated but, unfortunately, nothing is said about how it is dephosphorylated even if it represents a crucial aspect for the proper *Pho4* function study *in vivo*. Also outside the nucleus the dephosphorylations must occur, otherwise the binding with *Pse1* and the reimportation during phosphate starvation is not possible; it is also probable that the dephosphorylation mechanisms inside and outside the nucleus are different.

**Cooperative binding of *Pho4* and *Pho2* at the *Pho5* and *Pho81* promoters and the *Pho81* positive feedback loop.** *Pho5* promoter encodes the major isoenzyme of acid phosphatase, an oligonumeric, heavily glycosylated extracellular enzyme, which provides cells with phosphate by hydrolyzing phosphomonoesters scavenged from environment. So *Pho5* is the most important gene for the cellular response to phosphate starvation.

There are two regulatory elements on the *Pho5* promoter: *UASp1* and *UASp2*. The DNA binding to these elements was demonstrated to be highly cooperative between *Pho4* and *Pho2*. One *Pho4* protein and one *Pho2* protein bind to *UASp1*, while two *Pho2* and one *Pho4* bind cooperatively to *UASp2*. The cooperative mechanism avoids *Pho4* to bind to the promoter without interaction with *Pho2*. Some other low-affinity binding site for *Pho4* and *Pho2* has been found on the *Pho5* promoter, but they are still unclear and they seems to have a negligible role.

*Pho2* is a homeobox transcription factor that exemplifies combinatorial control in transcriptional activation of a diverse array of genes; in [51] it is shown that also *Pho2* is phosphorylated and its kinase is probably *CDC*28.

Even if the response to phosphate levels in the medium can be measured roughly only by *Pho5* concentration, *Pho81* regulation is also very important since it results in a positive feedback loop for the path. It has been demonstrated that the feedback mechanism is necessary for signalling the phosphate starvation, because without increasing the *Pho81* production, the *Pho80-Pho85* complex can continue at least partially to maintain *Pho4* phosphorylated. The biological behaviour of the promoter of *Pho81* is not so studied as the *Pho5* promoter, so little is known about it. From [27], and [53] we can assume that the *Pho81* promoter has only two binding sites corresponding to *UASp1* region of *Pho5* promoter.

## Sub-models for the PHO pathway

In this section we introduce the $\pi$-calculus models developed to simulate single sub-paths of the PHO pathway. The sub-models are described separately and are specified as general as possible because they have a general biological valence (in particular the multisite phosphorylation model) and can be useful for other pathways. The possibility of developing independently each sub-model is allowed by the compositionality property of process algebras through which the single sub-models can be integrated to model a whole pathway.

**The stochastic $\pi$-calculus multisite phosphorylation model.** Here we present a model applicable to every (multisite) phosphorylation regulation of a general substrate with a desirable number of phosphorylation sites and without any *a priori* assumptions.

The schematic description of the behaviour we want to model is represented in Figure 4 and in Figure 5. The first shows how the kinase-substrate binding and unbinding is modelled. The binding is performed through a communication on the *bind* channel on which the channel for the unbinding communication is transmitted. The binding rate, following the Gillespie algorithm, depends on the rate associated to the *bind* channel and on the number of kinase and substrate proteins, while the unbinding rate is governed by a stochastic delay in the *kinase_ bound* process after which the communication on the *unbind* channel occurs with an infinite rate.

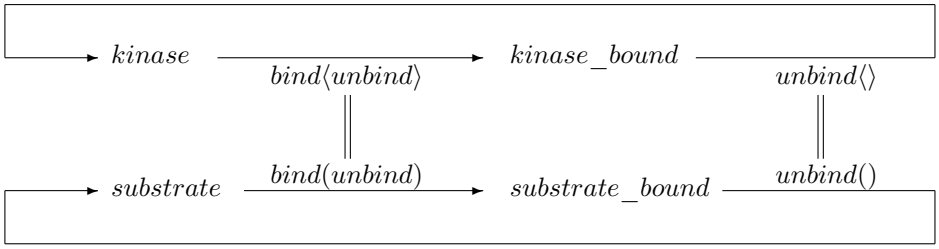



**Figure 4:** The kinase-substrate binding and unbinding mechanism.

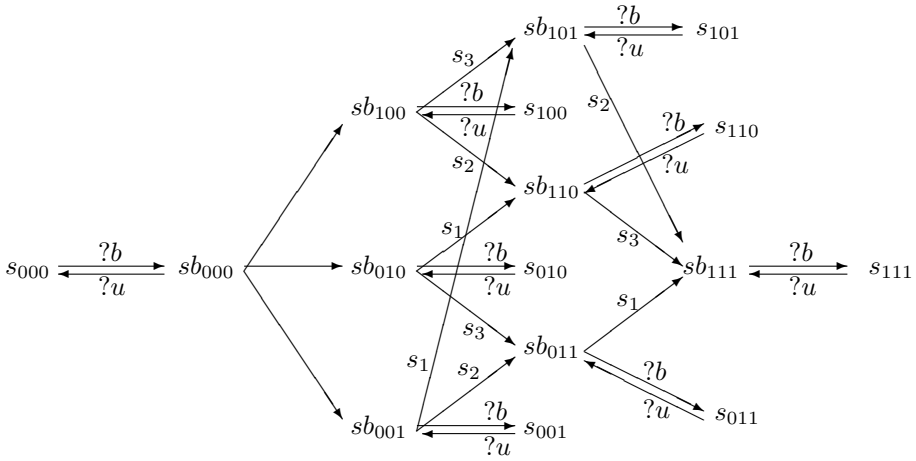



**Figure 5:** All the possible phosphorylation and binding/unbinsing transitions for a phosphorylation reaction with three phosphorylation sites. The free substrate is denoted with $s$, the kinase-substrate complex by $sb$ and the phosphorylation profile is defined by three boolean values (1 denotes phosphorylation). The phosphorylation event is denoted by $s_i$ where $i$ is the site number, the binding with $b$ and the unbinding with $u$.

The possible evolution of the system for a phosphorylation regulation with 3 different sites ($s_1$, $s_2$ and $s_3$) is shown in Figure 5. In the general, with $n$ phosphorylation sites, there are $2^n$ states representing each possible substrate phosphorylation profile and $2^n$ states for each possible substrate bound to the kinase.

Every kinase-substrate complex can disassociate in the two components or can perform a phosphorylation on a non already phosphorylated sites. All the phosphorylation transitions are based on synchronous communications on private channels.

In the description of the stochastic $\pi$-calculus model we consider a substrate with four phosphorylation sites and we focus only on one of the 16 possible phosphorylation profiles, an in particular on the one with the first and the last sites phosphorylated ($sub_{1001}$ and $sub_{1001}\_bound$ for the bound version). The other profiles have the same structure and phosphorylation mechanism. The description is performed with the non-stochastic formalism for clearness. A first approximation of the model is shown in (1). It contains the global channel $sub\_kin\_bind$ (line 1) responsible for the binding between the kinase and the substrate, the non-bound substrate processes (only $sub_{1001}$ is shown), the processes of the substrate bound to the kinase (only $sub_{1001}\_bound$ is shown) and the kinase process (line 12).

$$
\begin{array}{ll}
1 & \nu\, sub\_kin\_bind \\
2 & (\ !sub_{1001}().\nu\, unbind\ \ \nu\, kin\_unbind\ \ \nu\, s_2\ \ \nu\, s_3 \\
3 & \qquad (\ sub\_kin\_bind(kin\_unbind). \\
4 & \qquad\qquad (\ unbind\langle\rangle.kin\_unbind\langle\rangle.\mathbf{0} \\
5 & \qquad\qquad |\ s_2\langle\rangle.\mathbf{0} \\
6 & \qquad\qquad |\ s_3\langle\rangle.\mathbf{0} \\
7 & \qquad\qquad |\ sub_{1001}\_bound\langle unbind, s_2, s_3\rangle.\mathbf{0}\ )\ ) \\
8 & |\ !sub_{1001}\_bound(unbind, s_2, s_3). \\
9 & \qquad (\ unbind().sub_{1001}\langle\rangle.\mathbf{0} \\
10 & \qquad +\ s_2().sub_{1101}\_bound\langle unbind, s_3\rangle.\mathbf{0} \\
11 & \qquad +\ s_3().sub_{1011}\_bound\langle unbind, s_2\rangle.\mathbf{0}\ ) \\
12 & |\ !kin().sub\_kin\_bind\langle unbind\rangle.unbind().kin\langle\rangle\ )
\end{array}
\tag{1}
$$

The substrate process (lines $2-7$) is waiting for an interaction on the global channel $sub\_kin\_bind$; when the interaction occurs it calls its bound version ($sub_{1001}\_bound$) thus performing the kinase-substrate binding, passing as parameters the channels for the unbinding ($unbind$), and for the phosphorylation on the sites that are not already phosphorylated ($sp_2$ and $sp_3$). The three passed channels, on which three in parallel sub-processes of the substrate are willing to perform an output (lines $4-6$), are used by the relative bound version of the kinase (lines $8-11$) to perform the unbinding or the phosphorylation on $sp_2$ or $sp_3$. More precisely when a (empty) communication occurs on the $unbind$ channel the execution returns to the free version of the kinase ($sub_{1001}$), when a communication occurs on the $sp_2$ channel the bound kinase process with $sp_2$ phosphorylated is launched ($sub_{1101}\_bound$ not shown here) and similarly for the communication on the $sp_3$ channel.

The binding between the kinase and the substrate (schematized in Figure 4) occurs through the $sub\_kin\_bind$ on which the channel for the unbinding is transmitted. The unbinding signal, however, is sent by the substrate and not by the kinase because the binding duration depends also on the substrate phosphorylation profile which is not known by the kinase; so the unbinding signal is thrown with an infinite rate by the substrate after an internal delay which duration depends on the substrate phosphorylation state.

We can state two properties of the described model that are crucial for the correct modelling of the multisite phosphorylation biological behaviour:

- The interactions on the phosphorylation and unbinding channels are in competition and only one can occur in a single kinase-substrate reaction. The concurrent activity of the channels is assured by the parallel composition of the processes willing to perform a communication on the phosphorylation and unbinding channels (lines 4-6), while the competitive behaviour and the uniqueness of the performed communication is guaranteed by the semantics of the non-deterministic choice with which the receiving processes are organized (lines 9-11).

- The phosphorylation of a site is independent from the others. This is assured because when a phosphorylation occurs (lines 10 or 11) the relative new bound version of the substrate is launched passing as parameters the other (private) phosphorylation channels on which the sending operation is still willing to be performed since the kinase-substrate binding event (lines 10 or 11). Note that the unbinding channel acts similarly and so it is passed to the bound versions of the substrate (line 7); the only difference is that when the communication on the unbinding channel occurs (line 9), the not bound version of the kinase is launched without parameters thus terminating the site phosphorylations capability.

The model (1) is complete for a qualitative description but presents some problems from an effective simulation point of view. This is caused by the fact that some sub-processes remain active willing to perform an operation on some private channel that will never have a correspondent complementary communication operation. In a quantitative simulation framework this results in a huge and useless computational effort. In particular, when one of the branches of the non deterministic choice of the bound substrate (lines 9-11) is selected denoting a phosphorylation, the other branches are removed and so the parallel sub-processes of the substrate that are willing to perform an output on the other phosphorylation channels (lines 5-6) are blocked for ever.

The problem is tackled organising each sending operation on a phosphorylation channel in a non deterministic choice structure and forcing, if the phosphorylation does not occur, the execution of the other branch when the substrate and kinase disassociate.

$$
\begin{array}{ll}
1 & \nu\, sub\_kin\_bind \\
2 & (\ !sub_{1001}().\nu\, unbind\ \ \nu\, kin\_unbind\ \ \nu\, s_2\ \ \nu\, s_3\ \ \nu\, end\_s_2\ \ \nu\, end\_s_3 \\
3 & \qquad (\ sub\_kin\_bind(kin\_unbind). \\
4 & \qquad\quad (\ \overline{unbind}\langle\rangle.\overline{end\_s_2}\langle\rangle.\overline{end\_s_3}\langle\rangle.\overline{kin\_unbind}\langle\rangle.\mathbf{0} \\
5 & \qquad\quad |\ s_2\langle\rangle.end\_s_2().\mathbf{0} + end\_s_2().\mathbf{0} \\
6 & \qquad\quad |\ s_3\langle\rangle.end\_s_3().\mathbf{0} + end\_s_3().\mathbf{0} \\
7 & \qquad\quad |\ sub_{1001}\_bound\langle unbind, s_2, s_3\rangle.\mathbf{0}\ )\ ) \\
8 & |\ !sub_{1001}\_bound(unbind, s_2, s_3). \\
9 & \qquad (\ unbind().sub_{1001}\langle\rangle.\mathbf{0} \\
10 & \qquad +\ s_2().sub_{1101}\_bound\langle unbind, s_3\rangle.\mathbf{0} \\
11 & \qquad +\ s_3().sub_{1011}\_bound\langle unbind, s_2\rangle.\mathbf{0}\ ) \\
12 & |\ !kin().sub\_kin\_bind\langle unbind\rangle.unbind().kin\langle\rangle\ )
\end{array}
\tag{2}
$$

The correct model is shown in (2); two new private channels are added, $end\_s_2$ and $end\_s_2$ (line 4), to send the termination signal to the sub-processed for the phosphorylation input (lines 5 and 6) after the unbinding event which occurs on the *unbind* channel. Note that these two sub-processes have the receiving operation on the termination channels not only on the right branch of the non deterministic choice, but also on the left one. This because, without them, if a phosphorylation occurs the termination signal could no more be received and so the process is blocked on the sending of the termination signal.

The code fragments of the SPiM implementation of the model (2) is shown in figure 6.

**Reducing the processes number.** The conceptual model of the multisite phosphorylation needs $2^n$ states where $n$ is the number of phosphorylation sites, to represent each possible substrate phosphorylation profile. In our case the states are $2 \times 2^n = 2^{n+1}$ because the substrate and the kinase-substrate complex are conceptually separated. In the proposed $\pi$-calculus model there is a different process for each different states. This means that as the number of phosphorylation sites become larger, the number of the needed processes grows exponentially leading to a huge and non-readable model in addition to the computational and memory simulation problems.

The SPi machine, and so the SPiM simulator, permits the reduction of the processes with respect to the states allowing the parameters passing between the processes. In the phosphorylation model the

```
...
new sub_kin_bind@sub_kin_bind_rate : chan(chan)
...
and sub1001() = (
    new unbind@unbinding_rate_1001 : chan()
    new s2@s2_rate : chan()
    new s3@s3_rate : chan()
    new end_s2 : chan()
    new end_s3 : chan()
    ?sub_kin_bind(kin_unbind);
            (   !unbind;!end_s2;!end_s3;!kin_unbind;()
            |   do      !s2;?end_s2;()   or   ?end_s2;()
            |   do      !s3;?end_s3;()   or   ?end_s3;()
            |   sub1001_bound(unbind,sp2,sp3)   )   )
...
and sub1001_bound(unbind:chan,s2:chan,sp3:chan) = (
    do      ?unbind;sub1001()
        or  ?s2;substrate_bound_1101(unbind,s2)
        or  ?s3;substrate_bound_1011(unbind,s1)
)
...
and kin() = (
    new unbind : chan()
    sub_kin_bind(unbind);?unbind;kin()
)
...
```

**Figure 6:** The SPiM fragments of the stochastic version of the model (2).

idea could be to use only two processes (one for the substrate and one for the kinase-substrate complex) denoting the states with $n$ boolean values that are passed and possibly modified every time a recursive call of the processes is performed. Obviously every operation of the two processes that depends on the number of phosphorylation sites must be "guarded" by an *if-then-else* like operator which permits to enable only the desired operations. Different semantics of the *if-then-else* are possible (for example blocking or non-blocking behaviours) and this heavily influences the development of the model. However, the most important theoretical problem is the semantics of the non-deterministic choice; in fact it states that when a branch is selected the other ones can not be executed. So it is clear that if the *if-then-else* like operator is executed, the relative non-deterministic choice branch is selected independently if the conditional statement evaluates to *true* or to *false*.

Theoretically, the expressivity of the $\pi$-calculus does not decrease, under certain conditions, if the non deterministic choice is removed from the language. However, from a modelling point of view, the non deterministic choice is a powerful, intuitive and elegant primitive and with the parallel composition is one of the main reasons why the process calculi is particularly suitable in biology.

Summing up, in the $\pi$-calculus the processes reduction is intrinsically prevented by the use of the non-deterministic choice structure. The way for an effective reduction of the processes number can be represented, rather then by complex and possibly useless model modifications, by other $\pi$-calculus based languages like for example *Beta Binders* [56][57] that natively allows the multiple state handling by means of particular abstractions called interfaces.

**Model parameters** The rates associated to the channels of the model that are biologically meaningful are those regarding the site-specific phosphorylation rates and the kinase-substrate association and disassociation rates. These are the rates that need to be estimated and tuned consistently with the biological data. Apart for the phosphorylation rates, the model permits to set different values for each phosphorylation profile. All the other rates have only technical modelling purposes without biological

$$
\begin{array}{ll}
1 & \quad !\, S_1().\,(\tau_1.\, S_1\langle\rangle + \tau_2.\, S_2\langle\rangle) \\
2 & \mid\, !\, S_2().\,(\tau_1.\, S_1\langle\rangle + \tau_2.\, S_2\langle\rangle)
\end{array}
\tag{3}
$$

meanings and all have an infinite rate because the synchronous communications in which they are involved are instantaneous and do not need biological validation.

**Dynamic regulation of the number of processes.** In the PHO pathway, the modification of the inhibition capability of *Pho81* is phosphate concentration-driven. Unfortunately, in the $\pi$-calculus approach the information on the environment cannot be directly modelled. The possible solution is the modelling of the effects of the concentration modifications; the idea is that the user sets the percentage of inhibited *Pho81* proteins instead of the phosphate concentration. In the PHO pathway this is not a limitation since a precise correlation exists between the phosphate concentration and the number of inhibited proteins.

In principle, it is sufficient to set, in the initialization phase, the number of inhibited *Pho81*. However, since *Pho81* is regulated by the pathway with a feedback loop, the *Pho81* number changes dynamically while the percentage of inhibited proteins must remain stable. So also the regulation of the percentage of inhibited proteins is required to act dynamically.

For these reasons, we propose here a general mechanism to regulate the relative amount of a certain set of processes in a state $S_1$ with respect to another set of processes in state $S_2$ with possibility of adding or removing new processes in one of the two states and of changing the state. The basic mechanism is represented in Figure 7 and in the model (3); a process in $S_1$ can remain in the same state with a rate $r_1$ associated with the silent action $\tau_1$ or change the state with a rate $r_2$ associated with $\tau_2$, and viceversa for a process in $S_2$.

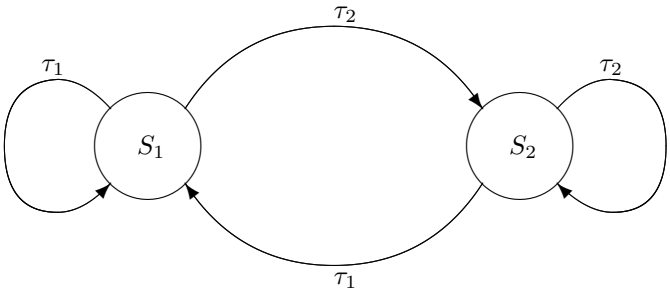



**Figure 7:** The representation of the model for the dynamic regulation of the number of processes. Each process can return in itself or can go to the other one; the ratio of the rates at which it happens determines the ratio between the number of one process with respect to the other one.

The balance between the two rates that regulate the two branches of the non deterministic choice is responsible for the balance of the number of the two processes. Intuitively, if we want the number of processes in $S_1$ to be $n$ times greater than those in $S_2$ it is necessary that $r_1/r_2 = n$ where $r_1$ and $r_2$ are the rates associated to the transition $\tau_1$ and $\tau_2$ respectively. Formally, denoting the percentage of processes in $S_1$ (and in $S_2$) with respect to the total number of processes in $S_1$ or in $S_2$ with $P_{S_1}$ (and $P_{S_2}$) it is required that:

$$
\begin{cases}
r_1 \times P_{S_2} = r_2 \times P_{S_1} \\
P_{S_1} + P_{S_2} = 100
\end{cases}
$$

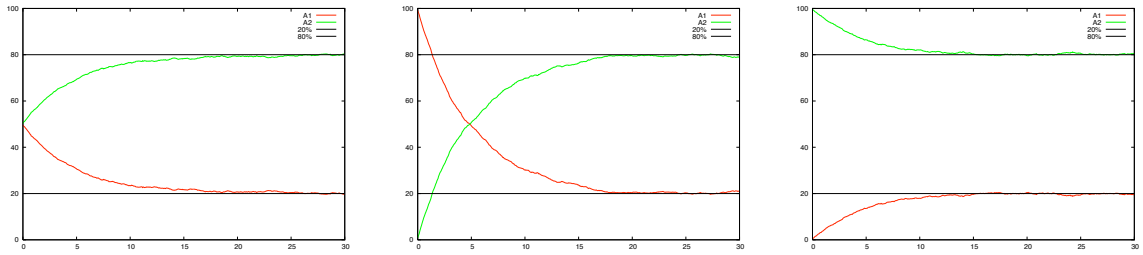



**Figure 8:** These three plots show the independency of the percentages at the steady state from the initial percentages. In this case $S_1$ percentage is set to 80% and, as consequence, the $S_2$ percentage to 20%.

Defining with $r = r_1 + r_2$ the algebraic sum of $r_1$ and $r_2$, we can derive the value of $r_1$ and $r_2$ with respect to $r$ in the following way:

$$r_1 = \frac{P_{S_1}}{100} \times r \quad r_2 = \frac{P_{S_2}}{100} \times r \qquad with \quad P_{S_1} + P_{S_2} = 100 \tag{4}$$

With this assumption and with a sufficient number of processes in $S_1$ and in $S_2$ and possibly after a convergence time, the framework assures that the percentage of processes in $S_1$ and in $S_2$ with respect to the total number of processes in the framework converges respectively to $P_{S_1}$ and $P_{S_2}$.

Some properties of the model:

- The percentage of process in $S_1$ or in $S_2$ after a convergence time is independent from the initial concentration of the two processes as shown experimentally in Figure 8. The initial percentages influence only the convergence time.
- The noise heavily depends on the number of processes.
- The convergence time depends on $r = r_1 + r_2$. It means that the average time needed by a single process in $S_1$ (or in $S_2$) to return in itself or going in the opposite state is $1/r$. The value of $r$ must be chosen before forcing the values of $r_1$ and $r_2$ with the equations (4). Notice that theoretically higher is $r$ higher is the accuracy of the framework but also the number of transitions and thus the computational effort.

However, this model has some limitations like the fact that the regulator and regulated processes are the same and that only a single process definition can belong to a state.

In (7) is proposed an extension in which the two states are *active* (A) or *inactive* (I). The regulated processes can be every processes with the following structure:

$$!P\_act().\,(\,\tau^1.\,P\_act\langle\rangle + \tau^2.\,P\_act\langle\rangle + \ldots + \tau^{n-1}.\,P\_act\langle\rangle + \tau^n.\,P\_act\langle\rangle\,) \tag{5}$$

with $n \geq 1$ and where each $\tau^j$ with $j \in \{1 \ldots n\}$ represents every possible process with a non infinite and reasonably short execution time. To be regulated each branch of this process is decomposed in the following way:

$$\begin{array}{ll} 1 & !\,P^1\_act(end).\,\tau^1.\,end\langle\rangle.\,\mathbf{0} \\ 2 & |\,!\,P^2\_act(end).\,\tau^2.\,end\langle\rangle.\,\mathbf{0} \\ 3 & |\ldots \\ 4 & |\,!\,P^{n-1}\_act(end).\,\tau^{n-1}.\,end\langle\rangle.\,\mathbf{0} \\ 5 & |\,!\,P^n\_act(end).\,\tau^n.\,end\langle\rangle.\,\mathbf{0} \end{array} \tag{6}$$

These branches are called from the $P_i\_act()$ process of line 3 of the model; note the index $i$ which denotes the possibility to have more than one type of process under regulation. The original model shown in (3)

acts here (lines 13 and 14) as the engine of the regulation but it is separated by the regulated processes; for every regulated process exists a $A$ or $D$ process reflecting its state. The engine works sending an empty communication on the global *activate* channel to activate an inactive regulated process on which it also synchronizes the state changing from inactive ($I()$) to active ($A()$). The inverse operation of inactivating a process acts similarly. Suppose, for example, that the state moves from active to inactive; as consequence a sending operation with an infinite rate is performed on the *inactivate* channel that must be received from the correspondent non-deterministic choice branch (line 6) of one of the real processes in the active state ($P_i\_act()$).

The engine processes ($A()$ and $I()$) contain in the non deterministic choice structure also the receiving operation for the termination of an active (or inactive) process; the signal is sent by an apposite process ($P_i^{end}\_active()$ or $P_i^{end}\_inactive()$ lines 17 and 18) and, after forwarding the termination signal to the regulated processes, the process $A()$ or $I()$ simply dies. To add a new process in the regulation framework it is sufficient to call the $P_i^{init}\_act()$ or $P_i^{init}\_inact()$ (lines 15 and 16) process according to the initial state we want to impose to the process. The $P_i^{init}\_act()$ process simply starts a new active process of the engine ($A()$) in parallel with the regulated process in its active state (and similarly $P_i^{init}\_inact()$).

$$
\begin{array}{ll}
1 & \nu\, P_i\_act\ \ \nu\, P_i\_inact\ \ \nu\, A\ \ \nu\, I\ \ \nu\, P_i^{init}\ \ \nu\, P_i^{init}\ \ \nu\, P_i^{end}\_act\ \ \nu\, P_i^{end}\_inact \\
2 & \nu\, activate\ \ \nu\, inactivate\ \ \nu\, active\_end\ \ \nu\, inactive\_end \\
3 & (\ !P_i\_act().(\quad \nu\, end\ \ P_i^1\_act\langle end\rangle.end().P_i\_act\langle\rangle \\
4 & \qquad\qquad + \ \ldots \\
5 & \qquad\qquad + \ \nu\, end\ \ P_i^n\_act\langle end\rangle.end().P_i\_act\langle\rangle \\
6 & \qquad\qquad + \ inactivate().P_i\_inact\langle\rangle \\
7 & \qquad\qquad + \ active\_term().\mathbf{0}\ ) \\
8 & \mid !P_i\_inact().(\quad \nu\, end\ \ P_i^1\_inact\langle end\rangle.end().P_i\_inact\langle\rangle \\
9 & \qquad\qquad + \ \ldots \\
10 & \qquad\qquad + \ \nu\, end\ \ P_i^n\_inact\langle end\rangle.end().P_i\_inact\langle\rangle \\
11 & \qquad\qquad + \ activate().P_i\_act\langle\rangle \\
12 & \qquad\qquad + \ inactive\_term().\mathbf{0}\ ) \\
13 & \mid !A().(\tau_1.A\langle\rangle + \tau_2.inactivate\langle\rangle.I\langle\rangle + active\_end().active\_term\langle\rangle) \\
14 & \mid !I().(\tau_1.activate\langle\rangle.A\langle\rangle + \tau_2.I\langle\rangle + inactive\_end().inactive\_term\langle\rangle) \\
15 & \mid !P_i^{init}\_act().(A\langle\rangle \mid P_i\_act\langle\rangle) \\
16 & \mid !P_i^{init}\_inact().(I\langle\rangle \mid P_i\_inact\langle\rangle) \\
17 & \mid !P_i^{end}\_act().active\_end\langle\rangle \\
18 & \mid !P_i^{end}\_inact().inactive\_end\langle\rangle\ )
\end{array}
\tag{7}
$$

The model, shown in (7), maintains the same properties of the one shown in (3) but can be applied to an arbitrary number of different process types and the regulated processes are separated by the regulation engine.

**The promoter modelling.** The gene transcription regulation is one of the powerful mechanisms through which a pathway can modulate its responses. The transcription dynamics are usually based on the promoters activity which is heavily influenced by the transcription factors which binds to particular regions of the promoter. Here we propose a simple model for describing the promoter activity; very briefly, when all the needed transcription factors are bound to the promoter, it allows the gene transcription starting and the unbinding of the bound transcription factors.

The model showed in (8) represents a promoter that needs the binding with three different transcription factors to transcribe the gene responsible of a particular protein. In this case we also assume that the biological mechanisms through which the transcribed *mRNA* results in the target protein is negligible or moldable with a single stochastic delay.

Theoretically the number of different processes needed to represent each possible binding profile for modelling a promoter regulated by $n$ transcription factors is $2^n$. However the representation of each

$$
\begin{array}{ll}
1 & \nu\, p_1 \;\; \nu\, p_2 \;\; \nu\, p_3 \\
2 & !promoter().\\
3 & \quad (\ \nu\, ok \;\; \nu\, tf_1^{out} \;\; \nu\, tf_2^{out} \;\; \nu\, tf_3^{out} \\
4 & \quad\quad (\ tf_1\langle tf_1^{out}\rangle.\, ok\langle\rangle.\, \mathbf{0} \\
5 & \quad\quad |\ tf_2\langle tf_2^{out}\rangle.\, ok\langle\rangle.\, \mathbf{0} \\
6 & \quad\quad |\ tf_3\langle tf_3^{out}\rangle.\, ok\langle\rangle.\, \mathbf{0} \\
7 & \quad\quad |\ ok().\, ok().\, ok().\, \tau_{transcription}. \\
8 & \quad\quad\quad (\ \tau_{prot\_cod}.\, new\_prot\langle\rangle \ |\ tf_1^{out}\langle\rangle.\, tf_2^{out}\langle\rangle.\, tf_3^{out}\langle\rangle.\, \tau_{reconfig}.\, promoter\langle\rangle\ ) \\
9 & \quad\quad )\\
10 & \quad )
\end{array}
\tag{8}
$$

possible state is redundant since the only important state is the one with all the transcription factors bound because it represents the only condition that allows the gene transcription. For this reason the model is based on the parallel composition of $n$ sub-processes for the binding of the $n$ transcription factors (lines 4-6) and on the process representing the transcription and the reconfiguration of the promoter after the transcription (line 7 and 8). The trick used to avoid the state space explosion consists in guarding the transcription process with $n$ inputs in sequence that are fired one by one by the transcription factor bindings. More precisely the guard consists in $n$ private channels called *ok* (declared in line 3 and used as guards on line 7) that are waiting for an input before the transcription process; a step of the guard is fired when one of the transcription factors has performed the binding through a public channel and the corresponding branch of the parallel composition has notified the event with an output on the *ok* channel. When the guard is completely evaluated the transcription is enabled and the model start the process of the new protein (line 8) after a transcription delay. In parallel with the new protein the promoter must perform the unbinding of the transcription factors, the reconfiguration delay and the recursive call to the promoter process. The unbindings occur through apposite private channels passed to the transcription factors at the binding time and are tuned with an infinite rate.

**Transmembrane transport modelling.** Here we propose a simple general structure for modelling the protein exportation and importation trough a membrane. It can be applied to membrane receptor or to general transport proteins. The schematic behaviour of the importer and of the exporter are shown in Figure 9; notice that the behaviour of a transport carried out by a membrane receptor or by a non membrane transporter can be distinguished only by the binding and transporting rates, since in the $\pi$-calculus modelling no spatial information are taken into account.

The $\pi$-calculus sub model for the exporter is shown in (9); the model for the importer is very similar. The exporter binds to the protein performing an output on the *prot_binding* channel on which it sends also the channel (with an infinite rate) for signalling the unbinding after the transport. A particular branch of the protein process (line 10) receives the binding signal and immediately performs an output on the unbinding private channel which can be received by the exporter only after performing a silent action simulating the transport delay. During the silent action both the exporter and the protein cannot perform other actions thus simulating a real binding even if an apposite process is not instantiated. After the unbinding the protein calls the process denoting its extra-membrane location characteristics ($prot^{out}$) while the exporter, after a delay for the reimportation process, performs a recursive call to itself.

**Protein synthesis and degradation.** Very often, in the computational specification of biological system, the number of the species in the model is considered constant. For this reason the modelled protein have no synthesis and degradation mechanisms. However, when the studied pathway presents some non constitutively expressed protein such mechanisms are needed. It is the case, for example, of the *Pho81* and *Pho5* species whose synthesis is regulated by the respective promoters. As a consequence, it is necessary to have a degradation mechanism, otherwise the number of the synthesized proteins

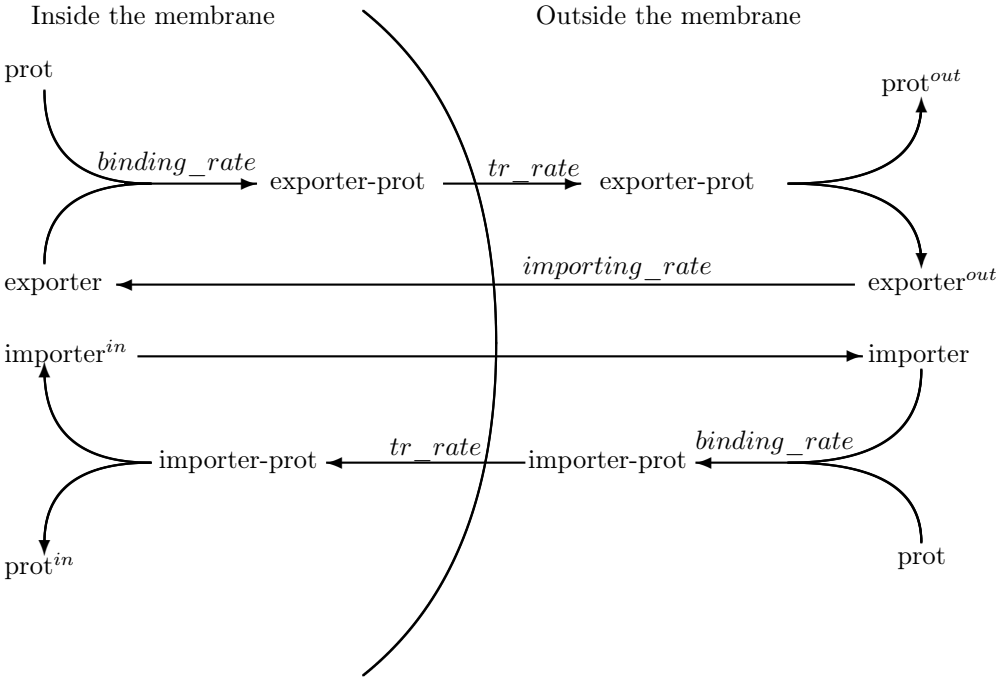



**Figure 9:** The schematic behaviour of the importing and exporting mechansims.

$$
\begin{array}{l}
\nu\, prot\_binding \\
(\ !\, exporter().\\
\qquad (\ \nu\, binding\_end \\
\qquad\quad (\ prot\_binding\langle unbinding\rangle.\, \tau_{trasnport}.\, unbinding().\, \tau_{import}.\, exporter\langle\rangle\ )\ ) \\
|\ !\, prot\langle\rangle. \\
\qquad (\ sub\_process\_1\langle\rangle \\
\qquad + sub\_process\_2\langle\rangle \\
\qquad + \ldots \\
\qquad + sub\_process\_n\langle\rangle \\
\qquad + prot\_binding(unbinding).\, unbinding\langle\rangle.\, prot^{out}\langle\rangle\ )\ )
\end{array}
\tag{9}
$$

grows for ever. However, also in the case of constitutively expressed proteins the synthesis/degradation mechanism can have a crucial role. Two cases are highlighted by the PHO pathway. The first regards the *Pho4* phosphorylation; since the protein is synthesized not phosphorylated, the constant degradation of phosphorylated *Pho4* causes an indirect dephosphorylation mechanisms. The second is related to the fact that when the synthesis and the degradation reach an equilibrium the number of proteins is constant but it is affected by a random noise; in some cases this noise can lead the system to a different behaviours with respect to a perfectly stable equilibrium.

```
1   ! prot_synth().τ_synth_rate.
2           ν degradation
3            ( prot_synth⟨⟩
4            | degradation⟨⟩
5            | prot⟨degradation⟩ )
6   | ! prot(degradation).
7            ( ...
8            + degradation().0 )
```
(10)

The model is shown in (10); the synthesis process (line 1-5) simply consists in a delay reflecting the synthesis rate after which the process of the new protein is launched in parallel with a recursive call of the synthesis process. The degradation is based on a rate (associated to the *degradation* channel declared on line 2) reflecting the half-life value of the protein; the new protein process is called (line 5) passing as argument the degradation channel on which a parallel sub-process is willing to perform an output. The new protein process has a reserved sub-process in non deterministic choice with all the other sub-processes that is willing to receive an input on the degradation channel after which the *prot* process terminates simulating the degradation.

Note that the synthesis process can be embedded in other processes like the case of the promoter modelling. The new protein process in (10) is not defined very precisely; in general, it must be assured that if another branch is selected a sub-branch for the degradation signal receiving is present and that if another process is called, the degradation channel is passed and handled similarly.

## The *Pho4* multisite phosphorylation model

The stochastic $\pi$-calculus multisite phosphorylation approach previously described is easily customizable to model the multi-domain phosphorylation of the *Pho4* transcription factor by the *Pho80-Pho85* cyclin-CDK kinase complex. Apart from the number of phosphorylation sites (five) the quantitative parameters of the model that need to be tuned are those regarding the different phosphorylation rates and the binding time between *Pho4* and *Pho80-Pho85* kinase complex with respect to the *Pho4* phosphorylation configuration. The model can set a different unbinding rate for every possible phosphorylation profile of *Pho4* but we choose to use only the five rates representing the mean of each possible state of the five different phosphoforms to be consistent with the existing biological data; the binding and unbinding rates are retrieved directly from [9]. Reproducing the *Pho4* phosphorylation reaction in vivo, the same work quantifies three biological aspects: a) the phosphoform concentration variations, b) the site preferences of the substrate calculated as the amount of each phosphopeptide as the percentage of the total phosphorylation for every phosphoform and c) the average number of phosphorylations per binding event without considering the binding with no phosphorylations. These data give enough information to infer the phosphorylation rates; we artificially set the rates and see then how they affect the overall behaviour and how they fit the experimental data. If the fitness is not satisfying, we modify some rates iterating the process until a reasonably good configuration is found. Notice that, in general, is not assured neither that the process eventually converges nor the available data give enough constraint to the system to apply

the process. In the particular case of setting the phosphorylation rates of *Pho4* the process is relatively simple because of the multidimensionality of the available data: the site preferences define the ratios between the phosphorylation rate of each site while the site preferences and the average phosphorylation influence the absolute "scale" of the values.

The *Pho4* multisite phosphorylation model code is available in the additional materials.

## The PHO pathway model

The whole modelling of the PHO pathway is developed applying the described sub-models to single PHO sub-paths; this approach is made effective by the compositionality property of process algebras. In particular the sub-models are applied as follows:

- The multisite phosphorylation model is applied to the reaction between *Pho4* and *Pho80-Pho85* as discussed in the previous section. The general sub-model is also extended with a simple dephosphorylation mechanism that is as general as possible because of the lack of biological information about this mechanism. The dephosphorylation is implemented internally of the *Pho4* processes and applied to the not *Pho80-Pho85* bound substrate profiles both inside and outside the nucleus. It consists in some branches of the non-deterministic choice (one for each phosphorylated site) that after a delay regulated by a rate perform the dephosphorylation on the site calling the *Pho4* form with the specific site not phosphorylated. Similarly to the phosphorylation mechanism the dephosphorylation rates can be set with site preferences.

- The framework to dynamically regulating the number of processes is applied to the *Pho81* cyclin-dependent kinase inhibitor to maintain a constant percentage of active inhibitors also with a variable total number of *Pho81* proteins. This because the phosphate concentration cannot directly be integrated in the model and the effect of the concentration is the level of *Pho81* active as *Pho80-Pho85* inhibitors.

- The promoter modelling is applied to the promoter of *Pho5* and *Pho81*. The first consists in five binding sites subdivided in 2 regulatory elements (*UASp1* and *UASp2*); three sites are for the *Pho2* binding and two for the *Pho4* binding. The promoter of *Pho81* has only one regulatory element (*UASp2*). Notice that the *Pho2* transcription factor can always bind to the promoters, while *Pho4* must be nuclear and with particular phosphorylation profiles.

- The transmembrane transport modelling is applied to the extranuclear exportation of *Pho4* with *Msn5* and the nuclear reimportation with *Pse1*. Both the importer and the exporter recognize and bind only to a subset of the phosphorylation profiles.

- The synthesis an degradation mechanisms are applied to *Pho4* (for its influence in the phosphorylation dynamics) and to the non constitutively expressed protein *Pho5* and *Pho81*

Some of the sub-models are applied to the same specie or mechanism; *Pho4* is interested by the transportation, phosphorylation, synthesis and degradation and the promoters integrate the synthesis process. In these cases the compositional approach is not immediate from a modelling point of view; however, if the processes are based on the non deterministic choice, very often the only needed operation is the union of the two structure representing different behaviours of the same entity. Figure 10 shows a fragment of the PHO pathway model representing the *Pho4* transcription factor in one of the 32 phosphorylation profiles. The structure is the non deterministic composition of the phosphorylation (first branch of the more external non deterministic choice), degradation (second branch), dephosphorylation (third, fourth and fifth branches) and extranuclear exportation (the last three branches) mechanisms. Moreover, the degradation operation needs that the degradation channel is passed to every sub-processes (*pho4_11010_bound*, *pho4_01010*, *pho4_10010*, *pho4_11000*, *pho4_11010*, *pho4_11010* and *pho4_11010*).

```
...
and pho4_11010(degradate:chan) = (
    new unbind@bound_time_phospho3_11010 : chan()
    new degradation : chan()
    new sp3@sp3_c : chan()
    new sp6@sp6_c : chan()
    new end_sp3 : chan()
    new end_sp6 : chan()
    (
    do  ?pho80pho85pho81_pho4_bind(pho80pho85pho81_pho4_bind_out);bind();
        (
            do      !unbind;!pho80pho85pho81_pho4_bind_out;!end_sp3;!end_sp6;()
                or  ?degradation;!pho80pho85pho81_pho4_bind_out;!end_sp3;!end_sp6;()
        |   do      !sp3;?end_sp3;() or ?end_sp3;()
        |   do      !sp6;?end_sp6;() or ?end_sp6;()
        |   pho4_11010_bound(degradate,degradation,unbind,sp3,sp6)   )
    or  ?degradate;()
    or  delay@dephospho_sp1_rate;pho4_01010(degradate)
    or  delay@dephospho_sp2_rate;pho4_10010(degradate)
    or  delay@dephospho_sp4_rate;pho4_11000(degradate)
    or  ?UASp1s1(unbindP:chan);?unbindP;pho4_11010(degradate)
    or  ?UASp2s2(unbindP:chan);?unbindP;pho4_11010(degradate)
    or  ?UASp1s1b(unbindP:chan);?unbindP;pho4_11010(degradate)  )    )
...
```

**Figure 10:** The SPiM fragments of the stochastic model of the *Pho4* transcription factor when it is nuclear and phosphorylated on sites $sp_1$, $sp_2$ and $sp_4$.

The protein abundances are taken from [58] and [59] and scaled by a factor of 10 for computational reasons. The PHO pathway model code is available online in the additional materials.

## *In silico* biological analyses

The *in silico* experiments were all performed on ordinary desktop systems (standard Pentium 4, 3.0 GHz processors); the *Pho4* multisite phosphorylation model requires about 2 hours of computation, while the PHO pathway model (with the proteins number scaled by a factor of 10) requires from 10 to 15 hours depending on the initial percentage of active *Pho81*. In order to improve robustness and precision of the computational results, basic Montecarlo statistical methods were applied, leading the simulation time necessary to obtain the proposed analyses to about 50 hours for the *Pho4* multisite phosphorylation model and 400 hours for the PHO pathway model.

The biological outcomes of the in silico analyses are listed in the following paragraphes.

***Pho4* multisite phosphorylation model tuning and validation.** The model fit well the three *in vitro* experimental datasets taken from [9]. The phosphoforms time dependent quantification is reported in Figure 11 while Figure 12 points out the differences with regard to the site preferences of site $sp_2$, $sp_3$, $sp_4$ and $sp_6$ ($sp_1$ is not taken in account because no *in vitro* data are available) between the traditional and the new proposed approach. In [9] the average number of phosphorylations for binding event is calculated simply multiplying the percentage of each phosphoform by the number of times it was phosphorylated (we call this calculation *ExpAvgPho*). The model based estimation of *ExpAvgPho* is very similar to biological data: the *in vitro* experiments of [9] give 2.06 while the output of the *in silico* experiments is 2.05.

***In silico* estimation of the average number of *Pho4* phosphorylation events per binding event between *Pho4* and the cyclin-CDK *Pho80-Pho85*.** The average number of phosphorylations for

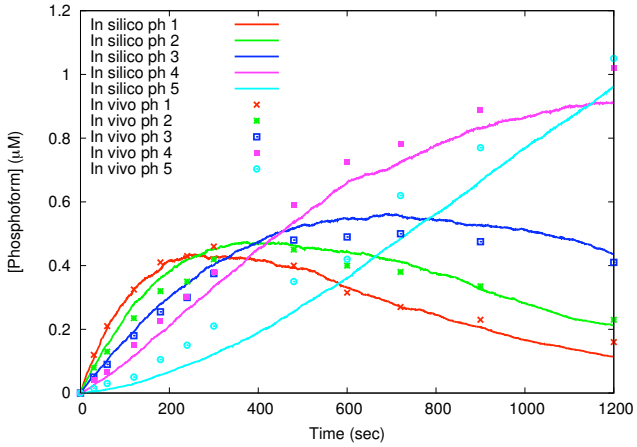



**Figure 11:** The *Pho4* multisite phosphorylation model validation. The comparison between *in vitro* and *in silico* on the phosphoform concentrations among the reaction course; the points represent the *in vitro* experimental values taken from [9] (Figure 5a), the curves the continuous-time *in silico* simulations.

| | $sp_2$ site | | | $sp_3$ site | | | $sp_4$ site | | | $sp_6$ site | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bio | sπ | err | bio | sπ | err | bio | sπ | err | bio | sπ | err |
| Phosphoform 1 | 12.0 | 12.2 | 0.2 | 12.0 | 12.1 | 0.1 | 16.1 | 15.2 | 0.7 | 59.9 | 60.5 | 0.6 |
| Phosphoform 2 | 17.1 | 16.9 | 0.2 | 17.1 | 16.7 | 0.4 | 22.0 | 20.9 | 1.1 | 43.8 | 45.5 | 1.7 |
| Phosphoform 3 | 20.6 | 20.4 | 0.2 | 20.6 | 20.5 | 0.1 | 24.4 | 23.9 | 0.5 | 34.4 | 35.2 | 0.8 |
| Phosphoform 4 | 23.0 | 23.1 | 0.1 | 23.0 | 23.1 | 0.1 | 24.2 | 24.8 | 0.6 | 29.8 | 29.0 | 0.8 |
| Phosphoform 5 | 24.0 | 24.0 | 0.0 | 24.0 | 24.0 | 0.0 | 25.1 | 25.0 | 0.1 | 26.9 | 27.0 | 0.1 |

**Figure 12:** The *Pho4* multisite phosphorylation model validation. This table represents the quantification of the site preferences calculated as the amount of each phosphopeptide as the percentage of the total phosphorylation for every phosphoform. The *bio* column is extracted from [9] (Figure 3g) that it is derived from *in vitro* experiments, the sπ column contains the results of the stochastic $\pi$-calculus simulations and the *err* column contains the absolute differences between the two approaches.

binding event computed following the *ExpAvgPho* calculation does not consider the bindings with no phosphorylations, leading to an overestimation of the phosphorylation activity. The *Pho4* multisite phosphorylation model is developed respecting the independence between the kinase-substrate binding and each site-specific phosphorylation and removing the phosphorylation channels of the sites on which an interaction occurred. In this way the model natively prevents attempts to phosphorylate sites that were already phosphorylated and permits kinase-substrate unbindings before any phosphorylation events thus allowing the possibility of binding events without phosphorylations. Dividing the total number of binding events for the total number of phosphorylation events the model permits to estimate the average number of phosphorylations per binding in a correct, complete and natural way (we call this calculation *ModelAvgPho*). Applying statistical methods to repeated stochastic *in silico* experiments referred to a small time window at the beginning of the reaction in order to improve robustness and precision, we obtained a value of the average number of phosphorylations for binding event of 1.45. It means that

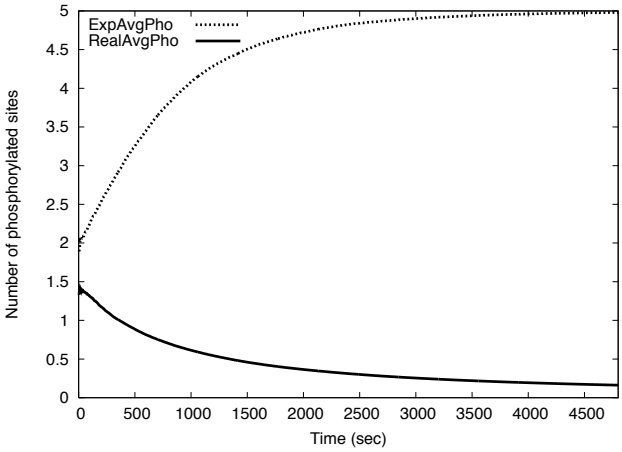



**Figure 13:** The *Pho4* multisite phosphorylation simulation. Here are represented the modifications during the reaction of the two different ways to quantify the number of phosphorylations for binding event. *ExpAvgPho* represents the calculation without considering the bindings with no phosphorylations (as performed in [9]), *ModelAvgPho* the quantification obtained from the stochastic $\pi$-calculus model considering also the null events. The horizontal line points out the limit under which the phosphorylation mechanism is considered completely distributive.

the *Pho80-Pho85* kinase activity on *Pho4* substrate remains semi-processive but closer to a distributive behaviour than to a processive one. Moreover, moving the time window along the reaction course, it is possible to notice, as shown in Figure 13, that the kinase begins to have a complete distributive behaviour (the average phosphorylation value is lower than 1) after about 6 minutes from the beginning. Theoretically, as confirmed by Figure 14 which extends the duration time of the analysis in Figure 13, the average model based phosphorylation value (*ModelAvgPho*) correctly approaches zero as the reaction goes to completion while the quantification following the calculation used in [9] (*ExpAvgPho*) reaches 5 when all the *Pho4* proteins are phosphorylated and only phosphoforms 5 are present.

**Kinetic analysis of *Pho4* phosphorylation profiles predicts biological outcomes.** The *Pho4* multisite phosphorylation computational model allows the analysis of the concentration of each of the 32 phosphorylation profiles among the reaction course; it shows that only few species (one or at most three) per phosphoform are predominant (see the additional materials for the analysis of each phosphoform and Figure 16 where only the predominant profiles are reported).

Figure 17 reports the number of phosphorylations on the five different sites and on $sp_2$ and $sp_3$ at

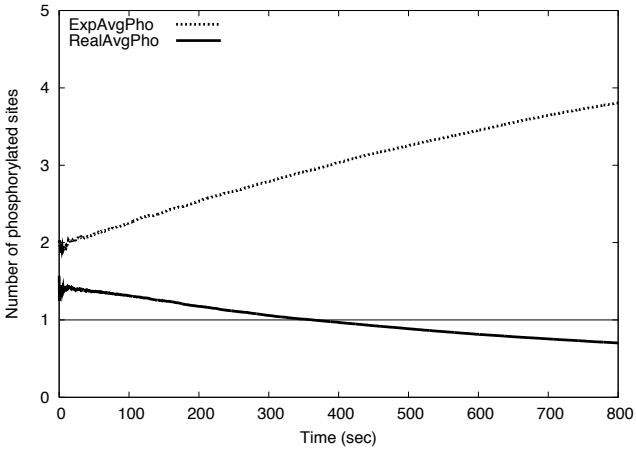



**Figure 14:** The *Pho4* multisite phosphorylation simulation. This plot represents the $ExpAvgPho$ and $ModelAvgPho$ as defined in Figure 13 but extending the simulation time (80 minutes) in order to better show how $ExpAvgPho$ approaches 5 and $ModelAvgPho$ approaches 0 as the reaction goes to completion.

the same time, thus providing for each time instant the percentage of *Pho4* molecules having particular functions and behaviours. For example, it states that about after three minutes half of the transcription factors are phosphorylated on $sp_6$ and so inactivated for the cooperative binding with *Pho2* to the promoter of *Pho5* gene, while about ten minutes are necessary to have half of the promoters phosphorylated both on $sp_2$ and $sp_3$ and so ready to be exported in the cytoplasm by the extranuclear transporter $Msn5$. The simulation confirms that the *Pho5* transcription inhibition is the result of two different regulations (as stated in [36]): the cooperative inhibition and the extranuclear localization that act simultaneously but with different dynamics. More precisely we give an account of the fact that the regulations of *Pho2*-dependent and *Pho2*-independent genes are separated and influenced by two different phosphorylation set of sites. The $sp_2$ and $sp_3$ sites regulate the nuclear localization of the *Pho4* and so its possibility to control the *Pho2*-independent genes, while $sp_6$ inhibits the cooperative binding with *Pho4* and so the *Pho2*-dependent gene expression. Moreover, the very low number of *Pho4* phosphorylated on $sp_2$ and on $sp_3$ but not on $sp_6$ (always under 0.6% of total *Pho4*) means that the *Pho2*-dependent gene regulation is not influenced by the nuclear exportation. These predictions allow us to state some new hypotheses about the *Pho2* role. In particular it follows that $sp_6$ is the only site that regulates the *Pho2*-dependent genes because it is very unlikely (less than 1% of the cases) that *Pho4* proteins not phosphorylated on $sp_6$ are exported out of the nucleus. From the kinetics point of view, with the results of the *in silico* analyses, we can conclude that the $sp_6$ phosphorylation dynamic determines the $Pho2$-dependent genes transcription kinetics while the $sp_2$ and $sp_3$ ones determine the $Pho2$-independent genes transcription kinetics.

An observation that confirms an already noticed behaviour [9] is that the $sp_4$ is phosphorylated rapidly with respect to $sp_2$ and $sp_3$ especially if considered together, meaning that the nuclear reimportation is prevalently inhibited before forcing the cytoplasm localization and thus avoiding a futile iterative cycle.

**The whole PHO pathway results.** No precise quantitative biological information is available about the entire PHO pathway but the *Pho4* regulation characteristics and so the *in silico* predictions have a lower confidence with respect to the ones performed with the *Pho4* multisite phosphorylation model; however, the model proves to be consistent with the qualitative description of normal and starvation phosphate conditions.

Considering the level of unphosphorylated *Pho4*, of nuclear *Pho4* active and inactive as *Pho5* promoter

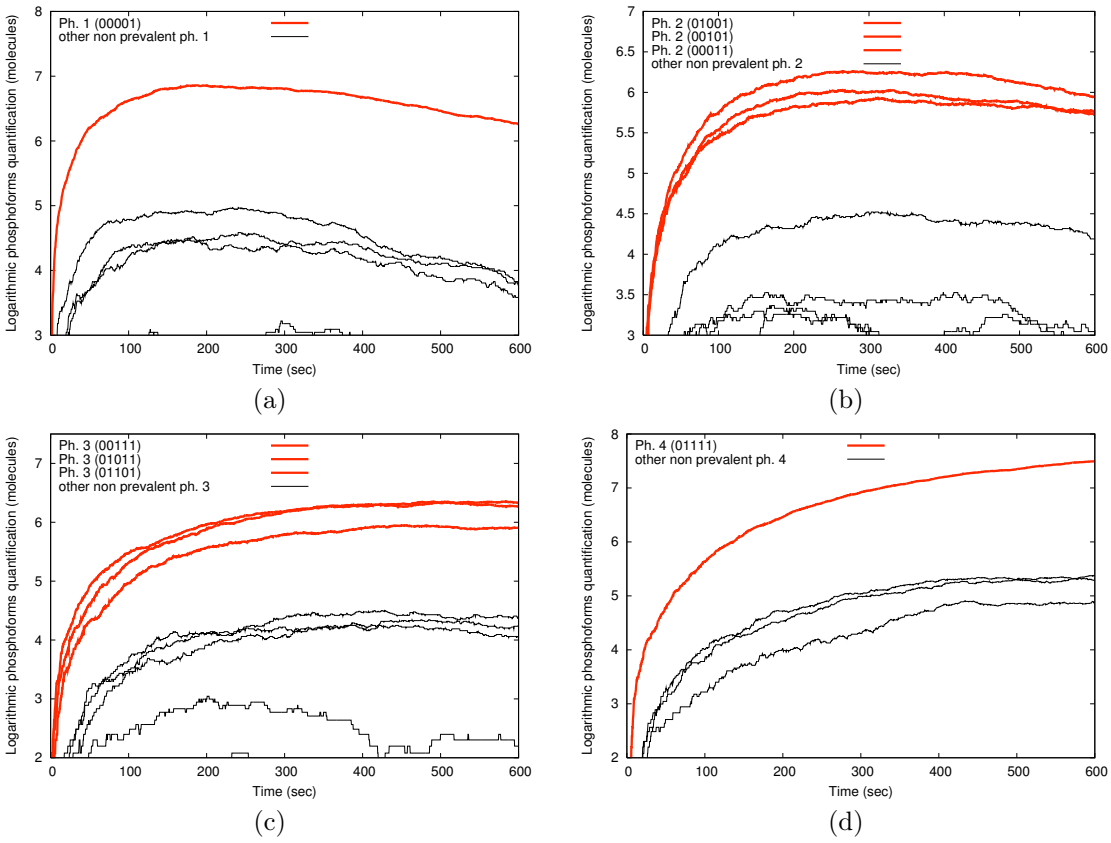



**Figure 15:** The *Pho4* multisite phosphorylation simulation. The representation of the absolute logarithmic amount (in number of molecules) of every phosphorylation pattern of phosphoform 1 (a), 2 (b), 3 (c) and 4 (d). The total amount of *Pho4* proteins is 10000. For each phosphoform the predominant species are highlighted (in red) and identified with five binary values representing respectively the phosphorylation state of $sp_1$, $sp_2$, $sp_3$, $sp_4$ and $sp_6$ (1 denotes the phosphorylation). Some non predominant species do not appear in the plots because their expression is too low.

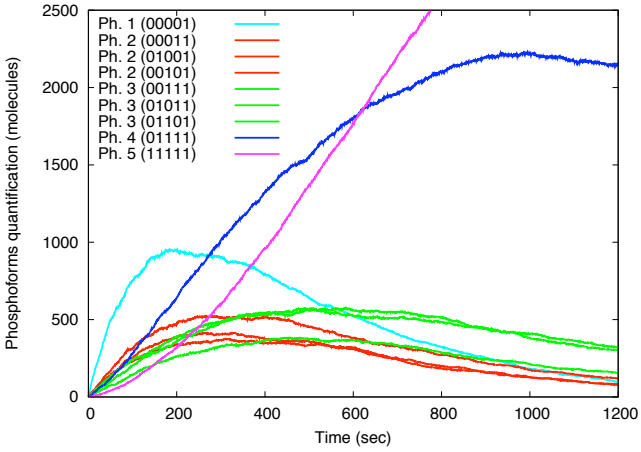



**Figure 16:** The *Pho4* multisite phosphorylation simulation. This graph represents the absolute amount during the reaction course of the prevalent phosphorylation patterns of every phosphoform (the total number of *Pho4* molecules is 10000). In particular every phosphoform has only one or three (in the case of phosphoform 2 and 3) predominant patterns. The predominant species are identified with five binary values representing respectively the phosphorylation state of $sp_1$, $sp_2$, $sp_3$, $sp_4$ and $sp_6$ (1 denotes the phosphorylation).

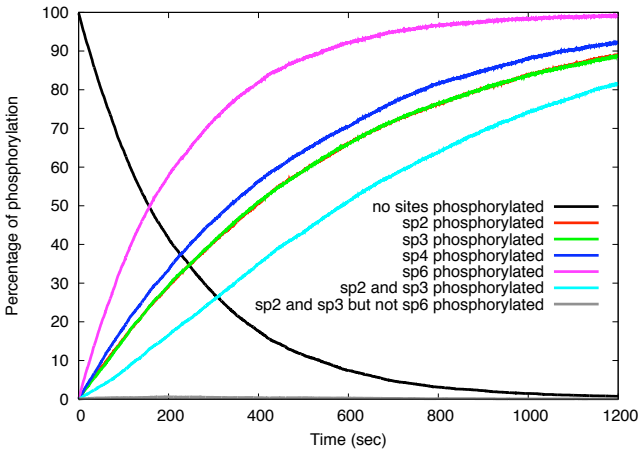
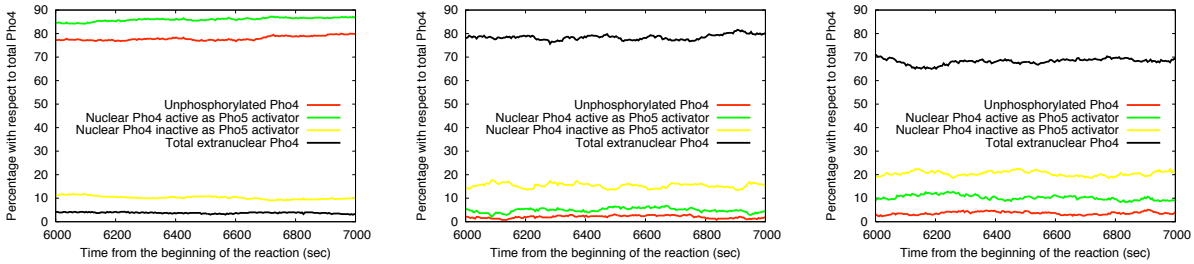
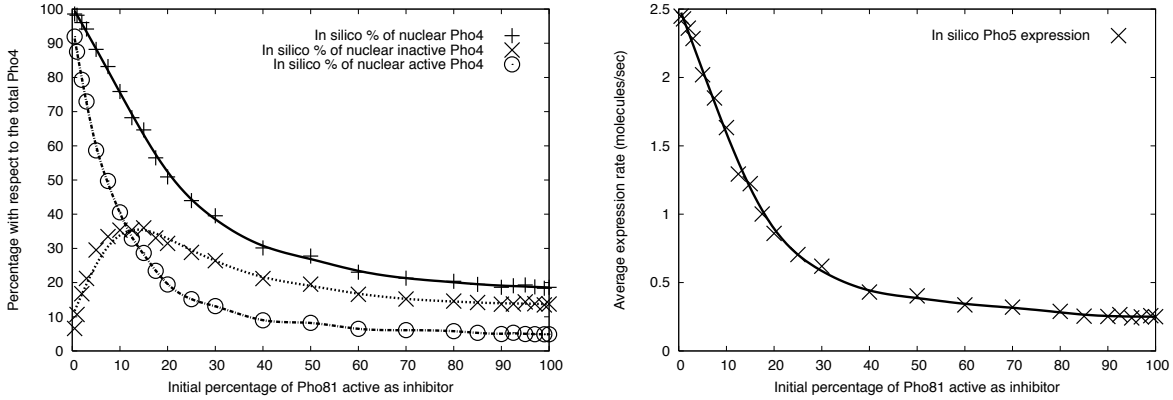



**Figure 17:** The *Pho4* multisite phosphorylation simulation. This plot represents the percentage of *Pho4* species phosphorylated in different and possible combined sites ($sp_1$ site is not considered because its role is unknown). The $y$ axis represents the percentages with respect to the total mount of *Pho4* (10000 molecules), the $x$ axis the time expressed in seconds.




(a) (b) (c)

**Figure 18:** The whole PHO pathway simulation. The three plots show the behaviour of the *Pho4* during phosphate starvation (a), normal phosphate condition (b), and starvation condition without the *Pho81* feedback loop in the model (c).




(a) (b)

**Figure 19:** The whole PHO pathway simulations in different phosphate conditions. The graph in (a) describes the predicted levels of nuclear *Pho4* and of transcriptional active and inactive nuclear *Pho4* with respect to the initial percentage of *Pho81* active as *Pho4* inhibitor. The graph in (b) represents the model-based *in silico* estimations of the *Pho5* expression rate with different initial percentages of *Pho81* active as *Pho4* inhibitor.

activator and of extranuclear *Pho4*, we show computationally the evidence of the crucial role of the *Pho81* feedback loop. This happens because the removal of the *Pho81* feedback influence from the model causes the *in silico* experiments to exhibit (Figure 18) a starvation induced behaviour (c) more similar to the normal phosphate condition (b) than to the starvation condition with the feedback mechanism (a).

The analysis of the nuclear *Pho4* confirms quantitatively the existence as stated in [36] of partial phosphate starvation responses to intermediate levels of phosphate in the medium. In the first graph of Figure 19 we can notice that for percentages between 10% and 20% the *Pho4* is prevalently nuclear but the cooperative binding with *Pho2* is inhibited allowing only the transcription of the *Pho2* independent genes. This is reasonably due to the fact related to the already discussed different phosphorylation kinetics of $sp_6$ site and $sp_2$ and $sp_3$ sites that cause, as the percentage of active *Pho81* increases, an initial fast increasing of nuclear inactive *Pho4* concentration (for the rapid $sp_6$ phosphorylation) followed by a gradual decreasing of the same species (for the slower combined phosphorylation of $sp_2$ and $sp_3$).

We performed a series of computational analyses about the expression levels of *Pho5*, a phosphatase responsible for example of the *Pho84* regulation [54], with respect to increasing phosphate concentrations. Since the phosphate concentration in the medium cannot be directly used (as discussed in Materials and Methods), the input of the PHO pathway model is indirectly defined by the initial percentage of *Pho81*

proteins active as inhibitors. We can notice from the simulations (second graph of Figure 19) that a low percentage of active *Pho81* is sufficient to increase the phosphatase transcription, but it is required that at least 50% of *Pho81* proteins are active to have the complete starvation signal. In fact for percentages greater than 50% the *Pho5* expression is more or less stable.

Finally we checked the qualitative consistency of the PHO pathway model with microarray data. The expression level ratios of *Pho5* and *Pho81* between normal and phosphate starvation conditions agree qualitatively with the microarray dataset available in [43] in the sense that the computational model is able to predict if they are overexpressed or underexpressed.

# Discussion

***Pho4* regulation model based analysis.** The model based estimation of the average number of *Pho4* phosphorylation events per *Pho80-Pho85* binding event includes also the null interactions in the weighted average. It permits to estimate a value of phosphorylations per binding which is 30% lower than the estimation without non effective bindings (from 2.06 to 1.45). This leads to consider the *Pho80-Pho85* kinase activity on *Pho4* substrate more likely to be distributive rather than processive or semi-processive. Moreover, if we consider not only the first time instants but all the reaction course, the model predicts that the number of phosphorylations per binding event further decreases obtaining a completely distributive behaviour (the average number of phosphorylation per binding event is lower than 1) after 6 minutes from the beginning of the reaction. It means that the kinetics of the *Pho4* regulation in phosphate starvation depends more on the number of kinase-substrate bindings than on the multiple phosphorylations per binding event with respect to the findings reported in [9].

We analysed the modifications during the reaction of the concentrations of each possible phosphorylation profile and in particular of the phosphorylation profiles that determine different behaviours of the transcription factor. This analysis, never performed with other approaches, allows to understand for each time instant the percentages of *Pho4* with a particular function. From the graph in Figure 17 two sets of sites with different roles can be observed. The $sp_6$ site regulates the transcription rate of the *Pho2*-dependent genes while the $sp_2$ and $sp_3$ sites determine the transcription rate of only the *Pho2*-independent genes because the cases in which the *Pho4* is phosphorylated on both $sp_2$ and $sp_3$ but not on $sp_6$ are extremely rare. Also the $sp_4$ phosphorylation occurs predominantly before the $sp_2$ and $sp_3$ phosphorylations avoiding the immediate nuclear reimportation after the cytoplasmic localization confirming computationally what already noticed in [9] and [36].

**Usefulness of the whole PHO $\pi$-calculus modelling.** The whole PHO model has very good theoretical prediction capabilities, but the lack of in-depth quantitative biological studies through which the model can be validated prevents the *in silico* experiment from performing very precise simulations. Even so, the computational model can verify some general mechanisms from a qualitative point of view. The first concerns the crucial role of the *Pho81* positive feedback loop for phosphate conditions signalling [60] [61]. The *in silico* simulations show that if in phosphate starvation the induced hyper-transcription of *Pho81* is not allowed, the *Pho4* regulation is very similar to the regulation in non starvation conditions.

Another analysis shows that multiple levels of phosphate induced response exists. In particular the minimum percentage of *Pho81* proteins active as *Pho80-Pho85* inhibitors necessary to signal a pure phosphate starvation is about 50%. For percentages between 10% and 50% the presence of a considerable amount of *Pho4* nuclear but prevalently phosphorylated on $sp_6$, results in an intermediate response in which the expression of the *Pho2* dependent and independent genes is very different. The possibility of intermediate phosphate induced behaviour was already proposed [36] but this work gives model based quantitative estimations (second graph of Figure 19).

The *in silico* analysis of the *Pho5* (and *Pho81*) expression is performed with respect to the initial percentage of *Pho81* proteins active as *Pho80-Pho85* inhibitors. The resulting graph, shown in Figure 19 (a), is useful for analysing the different phosphate starvation levels and activation thresholds. In particular, through this prediction and assuming to have experimental values of the *Pho5* expression in different phosphate conditions, it is possible to characterize the dynamics of the *Pho81 minimum domain* sensing of phosphate conditions [46] which is still almost unknown.

The computational model has also the potentialities to be validated and to be predictive with respect to microarray data; the genes whose expression is regulated by the proposed PHO model (*Pho5* and *Pho81*) agree qualitatively with the microarray dataset available in [43]. Unfortunately, in the PHO modelling the rates regarding the transcription dynamics are not well known and so precise quantitative predictions about the gene expression ratios between two phosphate conditions are at the moment not possible.

**The *in silico* approach for modelling (multisite) phosphorylation mechanisms.** The proposed *in silico* approach to the *Pho4* multisite phosphorylation mechanism revealed the possibility to detect, explore and quantify some aspects that have not been analysed previously mainly because of the intrinsic difficulty in planning proper experimental or computational approaches. This property is highlighted by the model based estimation of the average number of *Pho4* phosphorylation per *Pho80-Pho85* binding event.

The model proved to be able to handle the complexity of the phosphorylation mechanism and to be consistent with biological data. With such a model it is possible to computationally quantify every possible event. Also in relation to aspects already known in literature the *in silico* experiments adds to the qualitative knowledge also an in depth quantitative description permitting new possible analyses and predictions. Another desirable feature of the model is the time-continuous property of the *in silico* analyses typically not peculiar of the *in vitro* or *in vivo* ones.

The stochastic $\pi$-calculus model developed for studying the *Pho4* multisite phosphorylation is extendible with very few and non crucial modifications to all the phosphorylation mechanisms with a desired number of phosphorylation sites. Considering that multisite phosphorylation is one of the major cellular regulation mechanisms, the number of cases in which the proposed model can be applied for *in silico* experiments is potentially very large. Among all the possible multisite phosphorylation cases the most potentially promising ones are those that are still partially unknown from a biological point of view but for which indirect experimental values sufficient for the tuning and validation are available. The values of the model to set are the number of phosphorylation sites, the affinity between the kinase and the substrate, the phosphorylation rate of each site and the kinase-substrate disassociation rate. With respect to ODE based approach, no *a priori* assumptions on the general behaviour of the system are needed, no convergence and stability problems can occur even for small quantities, the non-determinism and noise are natively included in the simulator and a single simulation can investigate every aspect of the reaction.

Note that we can also use the model to try to infer the rates. We can artificially set the rates and see then how they affect the overall behaviour. This approach can be useful to plan specific *in vitro* or *in vivo* experiments to validate or disprove hypotheses.

**Process calculi approach for biological modelling of discrete and null events.** Among the possible paradigms to the *in silico* biological approach, the process calculi have formal, computational and modelling advantages as reviewed in [62]. In this work we found also a good feature related to the biological potentialities of process calculi approach. It regards the possible description of biological behaviours that are intrinsically very hard to measure through the qualitative knowledge of the mechanism and the indirect quantitative traditional experimental data. The binding and subsequent unbinding of the kinase to the substrate without phosphorylations is hardly detectable with *in vivo* or *in vitro* techniques

because its duration is very limited and does not change any species concentration; the process calculi description, instead, can detect and quantify every possible single or multiple step system transition. It follows that the central problem for process calculi in this context is not the quantification and simulation phase, but the adequate setting of the model parameters with respect to measured values; the tuning and validation phase cannot always be successfully performed and it heavily depends on the model structure and on the type and precision of the indirect biological data. In the specific case of the multisite phosphorylation of *Pho4* the rates can be set properly using only the *in vitro* phosphoform quantification and the site preferences.

Summing up, the innovative theoretical biological modelling feature of the process algebras highlighted by this work consists in the possibility, under certain conditions about the model structure and the available indirect data, to describe biological events that are null (in the sense that no concentration variation are appreciable), discrete and hardly measurable.

**Compositionality of process algebras for handling systems biology complexity.** The modelling approach used in this work consists in the development of small models regarding single biological aspects that can be integrated to specify the entire studied pathway. The sub-models integration is allowed by the compositionality property of the process algebras that is a modelling feature not handled by ODE based techniques and which can reasonably be the key for computationally handling the intrinsic complexity of systems biology. The PHO pathway modelling by means of a set of sub-models with very few code adaptations shows the theoretical crucial role of compositionality property [62].

The proposed (multisite) phosphorylation, promoter, transmembrane transporting, synthesis and degradation models are recurrent aspects in all the pathways. Starting from these models it is possible to implement a sort of "computational biological sub-paths library"; the idea is then to adapt the predefined general small models of the library to specific biological entities and aggregate them by means of compositionality to model complex pathway in a relatively easy and fast way. Applying compositionality also to the modelled pathway we can really obtain a powerful framework to lift the computational biology to the systems biology level.

# References


[1] Seet B-T, Dikic I, Zhou M, and Pawson T. Reading protein modifications with interaction domains. *Nat Rev Mol Cell Biol*, 7:473–483, 2006.

[2] Yang X-J. Multisite protein modification and intramolecular signaling. *Oncogene*, 24:1653Ű1662, 2005.

[3] Seo J and Lee K-J. Post-translational modifications and their biological functions: Proteomic analysis and systematic approaches. *J Biochem Mol Biol*, 37(1):35–44, 2004.

[4] Gunawardena J. Multisite protein phosphorylation makes a good threshold but can be a poor switch. *Proc Natl Acad Sci USA*, 102(41):14617–14622, 2005.

[5] Holmberg C-I, Tran S-E-F, Eriksson J-E, and Sistonen L. Multisite phosphorylation provides sophisticated regulation of transcription factors. *Trends Biochem Sci*, 27(12):619–627, 2002.

[6] Cohen P. The regulation of protein function by multisite phosphorylation - a 25 year update. *Trends Biochem Sci*, 25:596–601, 2000.

[7] Mayya V, Rezual K, Fong M-B, and Han D-K. Absolute quantification of multisite phosphorylation by selective reaction monitoring mass spectrometry. *Mol Cell Proteomics*, 5(6):1146–1157, 2006.

[8] Glinski M and Weckwerth W. Differential multisite phosphorylation of the trehalose-6-phosphate synthase gene family in arabidopsis thaliana. *Mol Cell Proteomics*, 4(10):1614–1625, 2005.

[9] Jeffery D-A, Springer M, King D-S, and O'Shea E-K. Multi-site phosphorylation of Pho4 by the cyclin-CDK Pho80-Pho85 is semi-processive with site preference. *J Mol Biol*, 306:997–1010, 2001.

[10] Markevich N-I, Hoek J-B, and Kholodenko B-N. Signaling switches and bistability arising from multisite phosphorylation in protein kinase cascades. *J Cell Biol*, 164(3):353–359, 2004.

[11] Batchelor E and Goulian M. Robustness and the cycle of phosphorylation and dephosphorylation in a two-component regulatory system. *Proc Natl Acad Sci USA*, 100(2):691–696, 2003.

[12] Huang C-Y-F and Ferrell J-E. Ultrasensitivity in the mitogen-activated protein kinase cascade. *Proc Natl Acad Sci USA*, 93:10078–10083, 1996.

[13] Cateau H and Tanaka S. Kinetic analysis of multisite phosphorylation using analytic solutions to michaelisŨmenten equations. *J Theor Biol*, 217:1–14, 2002.

[14] Regev A, Silverman W, and Shapiro E. Representation and simulation of biochemical processes using the $\pi$-calculus process algebra. *Pac Symp Biocomput*, 6:459–470, 2001.

[15] Priami C, Regev A, Shapiro E-Y, and Silverman W. Application of a stochastic name-passing calculus to representation and simulation of molecular processes. *Inf Process Lett*, 80(1):25–31, 2001.

[16] Errampalli D-D, Priami C, and Quaglia P. A formal language for computational systems biology. *OMICS*, 8:370–380, 2004.

[17] Ciocchetta F, Priami C, and Quaglia P. Modeling kohn interaction maps with beta-binders: An example. *T Comp Sys Biol*, pages 33–48, 2005.

[18] Lecca P, Priami C, Laudanna C, and Constantin G. A biospi model of lymphocyte-endothelial interactions in inflamed brain venules. *Pac Symp Biocomput*, pages 521–532, 2004.

[19] Kuttler C and Duchier D. Simulation of bacterial transcription and translation in a stochastic pi-calculus. *T Comput Syst Biol VI, Lecture Notes in Computer Science*, 4220:113–149, 2006.

[20] Curti M, Degano P, Priami C, and Baldari CT. Modelling biochemical pathways through enhanced pi-calculus. *Theor Comput Sci*, 325(1):111–140, 2004.

[21] Priami C. Stochastic pi-calculus. *Comput J*, 38(7):578–589, 1995.

[22] Lee M, O'Regan S, Moreau J-L, Johnson A-L, Johnson L-H, and Goding C-R. Regulation of the Pcl7-Pho85 cyclin-cdk complex by Pho81. *Mol Microbiol*, 38(2):411–422, 2000.

[23] Waters N-C, Knight J-P, Creasy C-L, and Bergman. The yeast Pho80-Pho85 cyclin-CDK complex has multiple substrates. *Curr Genet*, 46:1–9, 2004.

[24] Carroll A and O'Shea E-K. Pho85 and signaling environmental conditions. *Trends Biochem Sci*, 27(2):87–93, 2002.

[25] Wykoff D and O'Shea E-K. Phosphate transport and sensing in saccharomyces cerevisiae. *Genetics*, 159:1491–1499, 2001.

[26] Gregory P-D, Barbari S, and Hörz W. Transcriptional control of phosphate-regulated genes in yeast: the role of specific transcription factors and chromatin remodeling complexes in vivo. *Food technol biotechnol*, 38(4):295–303, 2000.

[27] Oshima Y. The phosphatase system in saccharomyces cerevisiae. *Genes Genet Syst*, pages 323–334, 1997.

[28] Lenburg M-E and O'Shea E-K. Signaling phosphate starvation. *Trends Biochem Sci*, 21:383–387, 1996.

[29] Persson B-L, Lagerstedt J-O, Pratt J-R, Pattison-Gramberg J, Lundth K, Shokrollahzadeh S, and Lundth F. Regulation of phosphate acquisition in saccharomyces cerevisiae. *Curr Genet*, 43:225–244, 2003.

[30] Komeili A and O'Shea E-K. Roles of phosphorylation sites in regulating activity of the transcription factor Pho4. *Science*, 284:977–980, 1999.

[31] Byrne M, Miller N, Springer M, and OŠShea E-K. A distal, high-affinity binding site on the cyclin-CDK substrate Pho4 is important for its phosphorylation and regulation. *J Mol Biol*, (335):57–70, 2004.

[32] Bhoite L-T, Allen J-M, Garcia E, Thomas L-R, Gregory I-D, Voth W-P, Whelihan K, Rolfes R-J, and Stillman D-J. Mutations in the Pho2 (Bas2) transcription factor that differentially affect activation with its partner proteins bas1, Pho4, and swi5. *J Biol Chem*, 277(40):37612–37618, 2002.

[33] Rudolph H and Hinnen A. The yeast ph05 promoter: Phosphate-control elements and sequences mediating mrna start-site selection. *Genetics*, 84:1340–1344, 1987.

[34] Barbaric S, Munsterkotter M, Goding C, and Horz W. Cooperative Pho2-Pho4 interactions at the PHO5 promoter are critical for binding of Pho4 to uasp1 and for efficient transactivation by Pho4 at uasp2. *Mol Cell Biol*, 18(5):2629–2639, 1998.

[35] Barbaric S, Munsterkotter M, Svaren J, and Horz W. The homeodomain protein Pho2 and the basic-helix-loop-helix protein Pho4 bind DNA cooperatively at the yeast PHO5 promoter. *Nucleic Acids Res*, 24(22):4479–4486, 1996.

[36] Springer M, Wykoff D-D, Miller N, and O'Shea E-K. Partially phosphorylated Pho4 activates transcription of a subset of phosphate-responsive genes. *PLoS Biol*, 1(2):261–270, 2003.

[37] Milner R, Parrow J, and Walker D. A calculus of mobile processes, parts I and II. Technical Report 86, 1989.

[38] Parrow J. *Handbook of Process Algebra*, chapter An Introduction to the $\pi$-calculus. Bergstra, Ponse and Smoka, 1999.

[39] Phillips A and Cardelli L. A correct abstract machine for the stochastic $\pi$-calculus. *Microsoft Research*, 2004.

[40] http://research.microsoft.com/˜aphillip/spim/.

[41] Gillespie D-T. Exact stochastic simulation of coupled chemical reactions. *J Phys Chem*, 81(25):2340–2361, 1977.

[42] Torriani-Gorini A, Silver S, and Yagil E. Phosphate in microorganism: Cellular and molecular biology. *American Society for Microbiology*, 1994.

[43] Ogawa N, DeRisi J, and Brown P-O. New components of a system for phosphate accumulation and polyphosphate metabolism in saccharomyces cerevisiae revealed by genomic expression analysis. *Mol Biol Cell*, 11:4309–4321, 2000.

[44] Huang S, Jeffery D-A, Anthony M-A, and O'Shea E-K. Functional analysis of the cyclin-dependent kinase inhibitor Pho81 identifies a novel inhibitory domain. *Mol Cell Biol*, 21(19):6695–6705, 2001.

[45] Knight J-P and Daly T-M. Regulation by phosphorylation of Pho81p, a cyclin-dependent kinase inhibitor in saccharomyces cerevisiae. *Curr Genet*, 46:10–19, 2004.

[46] Swinnen E, Rosseels J, and Winderickx J. The minimum domain of Pho81 is not sufficient to control the Pho85-Rim15 effector branch involved in phosphate starvation-induced stress responses. *Curr Genet*, 48:18–33, 2005.

[47] Kaffman A, Rank N-M, and O'Shea E-K. Phosphorylation regulates association of the transcription factor Pho4 with its import receptor pse1/kap121. *Genes Dev*, 12:2673–2683, 1998.

[48] Delahodde A, Pandjaitan R, Corral-Debrinski M, and Jacq C. Pse1/kap121-dependent nuclear localization of the major yeast multidrug resistance (mdr) transcription factor pdr1. *Mol Microbiol*, 39(2):304–312, 2001.

[49] Reiner A, Yekutieli D, and Benjamini Y. Identifying differentially expressed genes using false discovery rate controlling procedures. *Bioinformatics*, 19(3):368–375, 2003.

[50] Lau W-T-W, Schneider K-R, and O'Shea E-K. A genetic study of signaling processes for repression of PHO5 transcription in saccharomyces cerevisiae. *Genetics*, 150:1349–1359, 1998.

[51] Liu C, Yang Z, Xia Z, and Ao S. Regulation of the yeast transcriptional factor PHO2 activity by phosphorylation. *J Biol Chem*, 275(41):31972–31978, 2000.

[52] Bhoite L-T, Allen J-M, Garcia E, Thomas L-R, Gregory I-D, Voth W-P, Whelihan K, Rolfes R-J, and Stillman D-J. Mutations in the Pho2 (Bas2) transcription factor that differentially affect activation with its partner proteins bas1, Pho4, and swi5. *J Biol Chem*, 277(40):37612–37618, 2002.

[53] Murphy D. Gene expression studies using microarrays: Principles, problems, and prospects. *Adv Physiol Educ*, 26:256–270, 2002.

[54] Martinez P, Zvyagilskaya R, Allard P, and Persson B-L. Physiological regulation of the derepressible phosphate transporter in saccharomyces cerevisiae. *J Bacteriol*, 180(8):2253–2256, 1998.

[55] Pinson B, Merle M, Franconi J-M, and Daignan-Fornier B. Low affinity orthophosphate carriers regulate PHO gene expression independently of internal orthophosphate concentration in saccharomyces cerevisiae. *J Biol Chem*, 279(34):35273–35280, 2004.

[56] Priami C and Quaglia P. *Lecture Notes in Computer Science*, volume 3082, chapter Beta Binders for Biological Interactions, pages 20–33. Springer Berlin / Heidelberg, 2005.

[57] Degano P, Prandi D, Priami C, and Quaglia P. Beta-binders for biological quantitative experiments. In *QAPL 2006*, 2006.

[58] Ghaemmaghami S, Huh W-K, Bower K, Howson R-W, Belle A, Dephoure N, OŠShea E-K, and Weissman J-S. Global analysis of protein expression in yeast. *Nature*, 425:737–741, 2003.

[59] Huh W-K, Falvo J-V, Gerke L-C, Carroll A-S, Howson R-W, Weissmann J-S, and O'Shea E-K. Global analysis of protein localization in budding yeast. *Nature*, 425:686–691, 2003.

[60] Creasy C-L, Madden S-L, and Bergman L-W. Molecular analysis of the PHO81 gene of saccharomyces cerevisiae. *Nucleic Acids Res*, 21(8):1975–1982, 1993.

[61] Ogawa N, Noguchi K, Yamashita Y, Yasuhara T, Hayashi N, Yoshida K, and Oshima Y. Promoter analysis of the PHO81 gene encoding a 134 kda protein bearing ankyrin repeats in the phosphatase regulon of saccharomyces cerevisiae. *Mol Gen Genet*, 238:444–454, 1993.

[62] Priami C and Quaglia P. Modelling the dynamics of biosystems. *Brief Bioinform*, 5(3):259–269, 2004.