Petri Nets are a Biologist's Best Friend

Nicola Bonzanni[1,2], Anton Feenstra[1], Wan Fokkink[1], and Jaap Heringa[1]

[1] VU University Amsterdam, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands,
{n.bonzanni,k.a.feenstra,w.j.fokkink,j.heringa}@vu.nl
[2] The Netherlands Cancer Institute, Plesmanlaan 121, 1066 CX Amsterdam, The Netherlands

Abstract Understanding how genes regulate each other and how gene expression is controlled in living cells is crucial to cure genetic diseases such as cancer and represents a fundamental step towards personalised medicine. The complexity and the high concurrency of gene regulatory networks require the use of formal techniques to analyse the dynamical properties that control cell proliferation and differentiation. However, for these techniques to be used and be useful, they must be accessible to biologists, who are currently not trained to operate with abstract formal models of concurrency. Petri nets, owing to their appealing graphical representation, have proved to be able to bridge this interdisciplinary gap and provide an accessible framework for the construction and execution of biological networks. In this paper, we propose a novel Petri net representation, tightly designed around the classic basic definition of the formalism by introducing only a small number of extensions while making the framework intuitively accessible to a biology-trained audience with no expertise in concurrency theory. Finally, we show how this Petri net framework has been successfully applied in practice to capture haematopoietic stem cell differentiation, and the value of this approach in understanding the heterogeneity of a stem cell population.

Keywords: Petri nets, biology, gene regulatory networks

1 Introduction

Cancer causes the second largest proportion of noncommunicable (*i.e.* noninfective) disease deaths (21%) worldwide. It is projected that the annual number of deaths due to cancer will increase worldwide from 7.6 million to 13 million in 2030 [1]. Cancer and other genetic diseases are caused by abnormalities in genes which lead to changes in gene expression (*i.e.* the amount of information transcribed from genes in gene products such as RNAs and proteins) and/or in the structure of gene products. Understanding how gene expression is regulated is a necessary step to cure genetic illnesses. Gene expression, however, is a complex and tightly regulated process in which multiple regulatory elements interact concurrently to produce a global state which, in turn, determines the composite of a cell's observable characteristics (*i.e.* its phenotype, which includes the cell's shape, function, and health condition). The collection of the

interactions that controls gene expression can be encoded as a network. Gene regulatory networks (GRNs) are often represented as directed graphs in which nodes represent regulatory elements and arcs represent the effect of the presence of these elements (*i.e.* expression or repression) on the target node. However, simple directed graphs are not expressive enough to encode the complex interplay between different regulatory elements. In fact, the expression (or repression) of a single gene is controlled by the coordinated effort of multiple regulatory elements which sometimes compete to achieve opposite effects. The more nodes are included in a GRN, the more it becomes prohibitive, even for the experts in the field, to mentally associate with each set of arcs the complex interplay between the source nodes that cooperate to achieve gene expression (or repression). It seems therefore convenient to encode GRNs with a formalism which represents the prerequisites of each regulatory transition in an explicit way. Petri nets with their intuitive and explicit graphical notation already serve to model biological processes [2,3,4,5] and GRN in particular. Since the basic place-transition (PT) formalism has limitations that are incompatible with fundamental GRN properties, many extensions have been proposed during the years to model GRN [6,7,8,9]. However, while these extensions enable the use of Petri nets for GRN modelling, often they also pose an insurmountable obstacle for experimental biologists that are not used to handle complex formal methods and relate better to the simplicity of the original formalism. Therefore, we decided to address the limitation of the basic PT definition using only a small number of extensions and creating a representation intuitively accessible for experimental biologists, providing a way to fold (and unfold) the new network formalism into a traditional PT network. In collaboration with a team of experimental biologists led by Dr. Berthold Göttgens at the Cambridge Institute for Medical Research, we modelled the haematopoietic stem cells differentiation using our Petri net framework. The use of this framework (described for the first time in the following Sections) was instrumental to discover the inhibitory role of Gata1 on Fli1 – two crucial genes involved in blood cell differentiation and leukemia. The biological significance of this result and its implications are discussed in depth in a companion paper [10]. This paper was recently presented at ISMB 2013 (the largest and most prestigious Bioinformatics meeting), where it received the *Award for Best Paper in Translational Bioinformatics*. We view this prize as a token of appreciation from the biological community towards a framework that aims to bridge computer science and biology with an intuitive but powerful formalism.

The current paper, rather than focusing on theoretical advancements, aims to be an experience report describing the successful application of Petri nets to real-world experimental biology. The paper is structured as follows. In Sect. 2 we define a novel Petri net representation for GRNs based on the seminal work of Chaouiya and colleagues [7]. Section 3 elaborates on how one can interpret relevant state space properties in a biological sense. In Sect. 4 we provide an experience report on the application of our framework in the experimental lab directed by Dr. Berthold Göttgens, at the Wellcome Trust Institute/MRC in

Cambridge, UK. In Sect. 5 we discuss related work, and we present our conclusions in Sect. 6.

2 Gene regulatory networks as Petri nets

Gene regulatory networks are usually grounded on different assumptions relative to other biological networks such as signalling networks. Gene regulatory networks are based on two elements: a set of genes and a set of interactions between them. In turn, interactions can be either positive or negative, when a gene product has an enhancing or repressive effect, respectively, on the expression of another gene. A directed graph can not capture the cooperative interactions that are essential to correctly reproduce the behaviour observed during *in vivo* experiments. Intuitively, we want to be able to express, without ambiguity, whether a *single* gene (*de facto* its gene product) or *multiple* genes are required for a specific interaction.

An elegant way to avoid ambiguities is to use Petri nets to encode GRNs. In this framework, places represent genes, transitions represent interactions, and the marking of a place models the level of gene expression. Unfortunately, this simple construction, based on the standard definition of PT nets, conflicts with three main assumption of GRNs; in a GRN (i) the gene products (tokens) are not consumed by the interactions (transitions); (ii) interactions might have negative effects on the gene products (tokens can be removed from post-set places); (iii) the absence of a gene product (a place not marked) can be a prerequisite for an interaction. Instead of redefining enabling conditions and a marking function, we decided to build simple Petri nets modules that satisfy these assumptions by construction, and then use these modules to build larger GRNs.

Based on the above specification, and building on previous work by Chaouiya *et al.* [7], we represent each gene g in a GRN using two complementary places $\{p_g, \bar{p}_g\} \subseteq P$ where p_g represents g being expressed, and $\bar{p}_g$ represents g being repressed. The sum of tokens in p_g and $\bar{p}_g$ always equals $\mathcal{N} \in \mathbb{Z}_{>0}$, with $\mathcal{N}$ the maximum gene expression level. Each interaction is modelled by a transition. Let i be a positive interaction of the GRN. The set of genes R_i defines the gene products necessary for the occurrence of i, S_i defines the set of genes that block the occurrence of i, and g is the gene activated by the occurrence of i. Thus, it is possible to define a transition t_i modelling i such that the pre-set of t_i is

$$^\bullet t_i = \{p_r \in P \mid r \in R_i\} \cup \{\bar{p}_s \in P \mid s \in S_i\} \cup \{\bar{p}_g\}, \tag{1}$$

and the post-set of t_i is

$$t_i{}^\bullet = \{p_r \in P \mid r \in R_i\} \cup \{\bar{p}_s \in P \mid s \in S_i\} \cup \{p_g\}. \tag{2}$$

Intuitively, we want to enforce that t_i can be enabled if all the required gene products are available and all gene products blocking interaction i are absent. As a result of an occurrence of t_i, a token is moved from $\bar{p}_g$ to p_g, while all the tokens consumed in the places belonging to $^\bullet t_i$ are replaced by new ones.

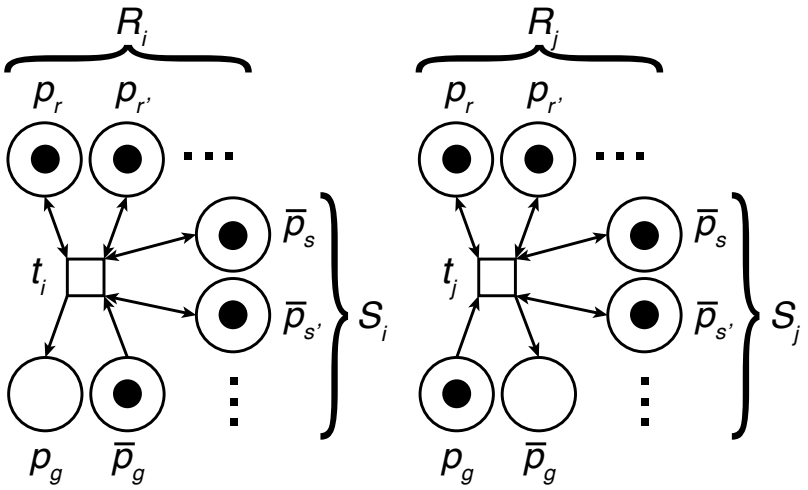

Figure 1. Petri net modules used to model positive (a) and negative (b) gene interactions in GRNs. R is the set of genes required for the occurrence of the interaction. S is the set of genes that block the interaction. Places p represent genes in expressed state while places $\bar{p}$ represent genes in repressed state. Arcs with a double arrow head denote arcs in both directions.

This construction, depicted in Fig. 1(a), complies with assumption (i) and (iii). Similarly, we can define a transition t_j that represents a negative interaction j on a gene g by moving a token from p_g to $\bar{p}_g$. Thus, the pre-set of t_j is

$$^\bullet t_j = \{p_r \in P \mid r \in R_j\} \cup \{\bar{p}_s \in P \mid s \in S_j\} \cup \{p_g\}, \tag{3}$$

and the post-set of t_j is

$$t_j{}^\bullet = \{p_r \in P \mid r \in R_j\} \cup \{\bar{p}_s \in P \mid s \in S_j\} \cup \{\bar{p}_g\}. \tag{4}$$

This construction, depicted in Fig. 1(b), models the negative effect of j on g gene products (see assumption (ii)), and also complies with assumptions (i) and (iii). Therefore, by combining these constructions, and inferring F from the pre- and post-sets, it is possible to build a full GRN that satisfies all three assumptions using the basic PT net formalism.

One limitation of this approach is that the graphical representation of non-trivial GRNs loses its intuitiveness. Indeed, the number of places and arcs necessary to model an interesting GRN explodes. Since an intuitive graphical representation is one of our goals, we tried to compress our construction by enriching the formalism. A minimal enrichment is sufficient to achieve an elegant formalism and an intuitive graphical representation. The extended formalism includes just an additional distinction between "positive" and "negative" arcs. Hence, our new framework is defined as a tuple $B = \langle \Pi, T, F, A, I \rangle$. Π is a set of places

such that there exists a single place in Π for every gene in the GRN. T is the set of transitions. F is the set of flow relations as defined for PT nets. $A \subseteq F$ and $I \subseteq F$ are disjoint sets of positive and negative arcs respectively such that $F = A \cup I$. Given a transition $t \in T$, we call

$$\begin{array}{ll} {}^\bullet t = \{\pi \in \Pi \mid (\pi, t) \in A\} & \text{positive pre-set of } t, \\ {}^\bullet \bar{t} = \{\pi \in \Pi \mid (\pi, t) \in I\} & \text{negative pre-set of } t, \\ t^\bullet = \{\pi \in \Pi \mid (t, \pi) \in A\} & \text{positive post-set of } t, \text{ and} \\ \bar{t}^\bullet = \{\pi \in \Pi \mid (t, \pi) \in I\} & \text{negative post-set of } t. \end{array} \tag{5}$$

Note that, by definition, there exists a surjective function $\gamma : P \to \Pi$ that associates with each place $p \in P$ the place $\pi \in \Pi$ that corresponds to the same gene. Now, it is possible to fold the constructions of Fig. 1 into our new Petri net definition using Alg. 1. This algorithm has two steps. The first step, intuitively, generates the set of places Π by compressing each couple of complementary places of P into a single place. The second step generates the set of arcs A and I. For each bidirectional arc from a positive place p to a transition t we create the arc $(\gamma(p), t)$ in A. For each bidirectional arc from a negative place $\bar{p}$ to a transition t we create the arc $(\gamma(\bar{p}), t)$ in I. Finally, given the effect, positive or negative, of the transition on a gene g we create an arc $(t, \gamma(p_g))$ in A or I, respectively. The unfolding procedure is similar. Each network $B = \langle \Pi, T, F, A, I \rangle$ can be graphically represented with great parsimony of elements as shown in Fig. 2. This representation is intuitive as well as formally rigorous.

Algorithm 1 fold(N, m), where the input $N = \langle P, T, F \rangle$ is a PT Petri net marked by a vector m

$\Pi \leftarrow \emptyset$, $A \leftarrow \emptyset$, $I \leftarrow \emptyset$, m' $\leftarrow \emptyset$
for all pairs of complementary places $p_g, \bar{p}_{g'} \in P$ such that $g = g'$ **do**
    $\Pi \leftarrow \Pi \cup \text{new}(\pi)$ ▷ where π is a new place corresponding to gene g
    m'$[\pi] \leftarrow$ m$[p_g]$
end for
for all $t \in T$ **do** ▷ see Fig. 1 notation
    for all $p_r \in P$ such that $r \in R_t$ **do**
        $A \leftarrow A \cup \{(\gamma(p_r), t)\}$
    end for
    for all $\bar{p}_s \in P$ such that $s \in S_t$ **do**
        $I \leftarrow I \cup \{(\gamma(\bar{p}_s), t)\}$
    end for
    if t models a positive interaction **then**
        $A \leftarrow A \cup \{(t, \gamma(p_g))\}$
    else
        $I \leftarrow I \cup \{(t, \gamma(p_g))\}$
    end if
end for
return $B = \langle \Pi, T, A, I, m' \rangle$

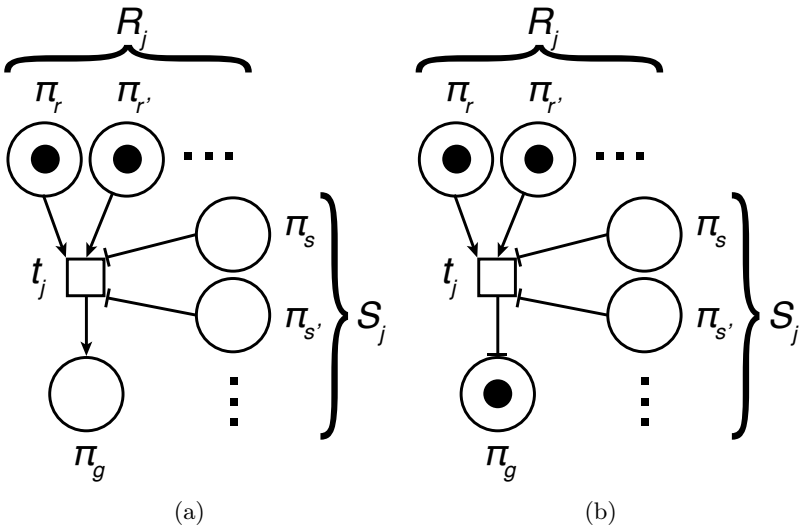

Figure 2. Graphical representation of the positive (a) and negative (b) gene interactions depicted in Fig. 1 using the folded network definition $B = \langle \Pi, T, F, A, I \rangle$. Negative arcs belonging to I have a flat arrowhead. Both transitions are enabled by the depicted markings. In practice, R and S are small, further simplifying the representation.

Although it is possible to unfold a network in its basic PT form where the traditional enabling conditions and marking transformations apply, it is convenient to lift these conditions and transformations. Thus, given a Petri net in the form $B = \langle \Pi, T, F, A, I \rangle$, a transition $t \in T$ is said to be pre-enabled by a marking m if every place in ${}^\bullet t$ is marked by m, and every place in ${}^\bullet \bar{t}$ is *not* marked by m; it is said to be post-enabled if $\mathrm{m}(\pi) < \mathcal{N}$ for each place $\pi \in t^\bullet$, and $\mathrm{m}(\pi) > 0$ for each place $\pi \in \bar{t}^\bullet$. Finally, t is said to be enabled by a marking m if it is pre- and post-enabled by m. The occurrence of t transforms the marking m into a mapping m' defined as

$$\mathrm{m'}(\pi) = \begin{cases} \mathrm{m}(\pi) - 1 & \text{if } \pi \in \bar{t}^\bullet, \\ \mathrm{m}(\pi) + 1 & \text{if } \pi \in t^\bullet, \\ \mathrm{m}(\pi) & \text{otherwise.} \end{cases} \tag{6}$$

This formalisation can be extended to use arc weights as shown before. However, by preserving this simple definition and assigning $\mathcal{N} = 1$, it is still possible to generate biologically meaningful networks as demonstrated in [10]. These networks behave in a Boolean fashion, *i.e.* each gene can either be expressed or repressed.

In order to build biologically faithful networks, an additional fourth assumption must hold: if transcription is suspended, all gene products should, eventually, degrade over time. In Petri net terms, although tokens are not consumed by gene interactions, they must be consumed whenever the conditions that enable

their production cease to hold. Intuitively, if a gene product g was produced in a previous step, but currently there are no more pre-enabled transitions that could have a positive effect on π_g, then gene product g should be degraded. This behaviour is achieved by adding new *ad hoc* transitions enabled when the conditions for gene activation are not met. Fortunately, this new set of transitions D and their pre- and post-sets can be inferred from the initial network topology, dispensing the user with the task of manually specifying them.

Formally, given a place $\pi \in \Pi$ it is possible to define

$$T_A^\pi = \{t \mid t \in T \wedge (t, \pi) \in A\} \subseteq T, \tag{7}$$

as the set of transitions that have a positive effect on π. Then, we represent T_A^π as a Boolean formula in disjunctive normal form (DNF). Each conjunctive clause of the DNF defines a transition $t \in T_A^\pi$. More specifically, each conjunctive clause uses as variables the places in ${}^\bullet t$, and the places in ${}^\bullet \bar{t}$ as negated variables. For instance, the network in Fig. 2(a) corresponds to the formula $(\pi_r \wedge \pi_{r'} \wedge \ldots \wedge \overline{\pi_s} \wedge \overline{\pi_{s'}} \wedge \ldots)$. Thus, the Boolean formula corresponding to a generic $T_A^\pi = \{t_1, t_2, \ldots, t_k\}$ is

$$\begin{aligned} \mathcal{B}(T_A^\pi) = & (\pi_{r_{t_1}} \wedge \pi_{r'_{t_1}} \wedge \ldots \wedge \overline{\pi_{s_{t_1}}} \wedge \overline{\pi_{s'_{t_1}}} \wedge \ldots) \\ & \vee (\pi_{r_{t_2}} \wedge \pi_{r'_{t_2}} \wedge \ldots \wedge \overline{\pi_{s_{t_2}}} \wedge \overline{\pi_{s'_{t_2}}} \wedge \ldots) \\ & \vee \ldots \vee (\pi_{r_{t_k}} \wedge \pi_{r'_{t_k}} \wedge \ldots \wedge \overline{\pi_{s_{t_k}}} \wedge \overline{\pi_{s'_{t_k}}} \wedge \ldots). \end{aligned} \tag{8}$$

Similarly, it is possible to reconstruct the network topology starting from a formula in DNF. A variable of (8) is *true* if the corresponding place is marked, *false* otherwise. Hence, if all preconditions of a transition are met, *i.e.* it is pre-enabled, then all the literals of the corresponding conjunctive clause, and the whole formula itself, evaluate as *true*. Therefore, for a generic place π, (8) evaluates as *true* if there exists a pre-enabled transition that has a positive effect on π.

$\overline{\mathcal{B}(T_A^\pi)}$, the negation of (8) in DNF, is *true* if at least one of its conjunctive clauses evaluates to *true*. By constructing D_π, the set of degradations of π, from the conjunctive clauses of $\overline{\mathcal{B}(T_A^\pi)}$, it is guaranteed that at least one of the transitions in D_π will be enabled if and only if *none* of the transitions with a positive effect on π is pre-enabled, satisfying the fourth assumption. For example, given the simple network of Fig. 3(a), $\mathcal{B}(T_A^{\pi_g})$ equals $\pi_a \wedge \pi_b \wedge \overline{\pi_c}$; therefore, $\overline{\mathcal{B}(T_A^{\pi_g})}$ is $\overline{\pi_a} \vee \overline{\pi_b} \vee \pi_c$. Fig. 3(b) shows the degradation transitions built from the conjunctive clauses of $\overline{\mathcal{B}(T_A^{\pi_g})}$.

3 State space analysis

For the analysis of GRNs we focus on computing the attractors of the model state space. The state space (or marking graph) is a directed multigraph where each node identifies a marking, and each arc represents the occurrence of a transition. An attractor is a forward invariant subset of the state space, *i.e.* a

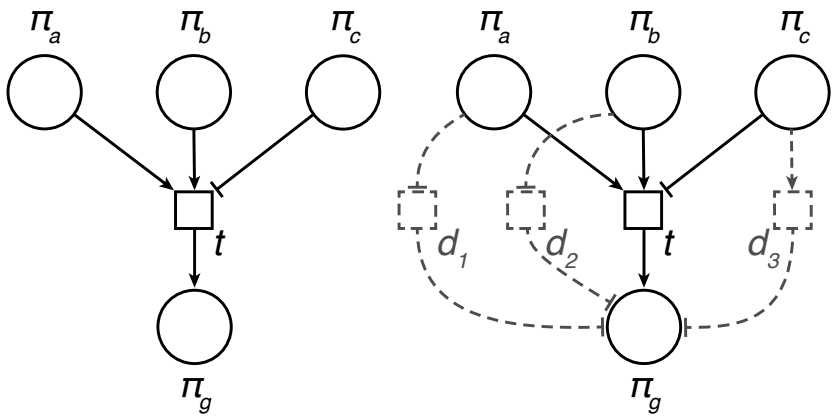

Figure 3. The network on the right (b) shows the set of transitions D (dashed lines) necessary to model the degradation of π_g. Notice that at least one transition in D is enabled if t is not pre-enabled. Since D can be automatically inferred from the gene interactions in the initial topology (a), it is safe and usually convenient to omit the transitions of D from the graphical representation to avoid cluttering.

set such that if a marking m belongs to it, then each marking reachable from m also belongs to this set. In a biologically faithful model, each attractor should correspond to an observable biological steady state. Intuitively, as in a biological steady state the recently observed behaviour of the system will continue into the future, likewise, in the state space the model execution will keep cycling between the same states in the attractor set. Since we are particularly interested in steady states, we can abstract from time. Therefore, we can compute the state space using a fully asynchronous semantics as shown in Alg. 2. The algorithm works in two steps. First, it generates all vertices of the state space graph by computing the ensemble of all possible initial markings, *i.e.* all bit vectors of size $|\Pi|$. Second, for each bit vector the algorithm fires all enabled transitions one by one. For each transition fired, an arc from the current bit vector to the vector generated by the occurrence of the transition is added to the state space.

The strategy we choose to identify the attractors is to compute the terminal strongly connected components (TSCCs) of the state space. TSCCs are a particular class of strongly connected components (SCCs). The SCCs of a directed graph are its maximal strongly connected subgraphs, *i.e.* the induced subgraphs formed by the equivalence classes defined on the vertices by the relation of *mutual reachability*. Two vertices u and v are said to be mutually reachable if and only if there exist a path from u to v and from v to u. A TSCC T is a SCC such that if $u \in T$, then $v \in T$ for each directed arc (u, v). Thus, a TSCC is a SCC that does not have outgoing arcs to other SCCs. Therefore, by trapping the execution in a subset of states, each TSCC is an attractor of the the dynamical system. Given

Algorithm 2 computeStateSpace(B), where $B = \langle \Pi, T, F, A, I \rangle$

```
1: E ← ∅
2: D ← computeDegradations(B)
3: T ← T ∪ D
4: V ← computeAllMarkings(|Π|)        ▷ compute all possible bit vectors of size |Π|
5: for all v ∈ V do
6:     E ← computeAllEnabledTransitions(T, v)
7:     for all e ∈ E do
8:         v' ← fireTransition(v, e)      ▷ compute the marking obtained by firing e in
   marking v
9:         E ← E ∪ (v, v')                                  ▷ add the new edge (v, v')
10:    end for
11: end for
12: return G = ⟨V, E⟩
```

a graph $G = \langle V, E \rangle$, the TSCCs can be easily computed by the Tarjan algorithm [11] in $O(|V| + |E|)$. The complexity of this analysis lies in the generation of the state space. The size of the multigraph G is exponential in the number of places; $|V| = \mathcal{N}^{|\Pi|}$, where we recall that $\mathcal{N}$ is the maximal gene expression level. For our case studies [5,10], based on the Boolean-like approach explained above, $|V|$ is 2^{11}, therefore, still tractable. Despite the complexity, building and exploring the whole state space instead of identifying single attractors may be valuable since the state space graph contains more information about the model dynamics that can be interpreted in a biological fashion. This information can lead to significative biological discoveries, as shown in [10]. However, more efficient strategies should be devised to further extend the formalism to multiple gene expression levels. Some strategies to cope with state space graphs with millions of nodes are presented in [12].

4 Modelling haematopoietic stem cell differentiation: An experience report

The Petri net framework defined in Sect. 2, has been specifically designed to be easily accessible to a biology-trained audience. We had the opportunity to test the applicability of our framework in the Haematopoietic Stem Cell Lab, directed by Dr. Berthold Göttgens, at the Wellcome Trust Institute/MRC in Cambridge, UK. The long term research goal of the Göttgens group is to decipher the regulatory networks responsible for blood stem cell development (*i.e.* haematopoiesis). Understanding the transcriptional control during blood cell differentiation and maturation is a necessary milestone to cure leukaemia and other types of cancer of the blood. Haematopoietic stem/progenitor cells (HSPC) have long served as a model for studying stem cells, *i.e.* cells able to differentiate, in a tree-like fashion, from a single stem (*i.e.* the stem cell) into multiple cells that accomplish different physiological functions (*i.e.* "mature" cells), such as red blood cells, white blood cells, and platelets.

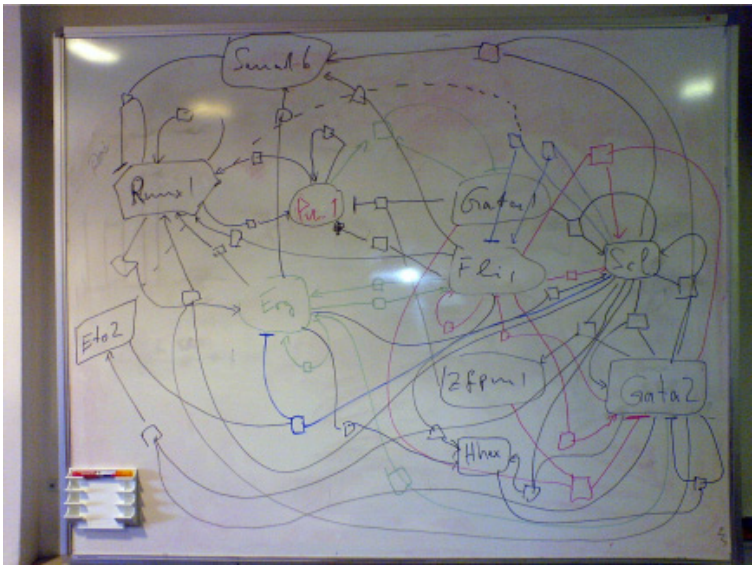

Figure 4. The first attempt to draw the GRN that controls the haematopoietic stem cell differentiation using the Petri net formalism explained in Sect. 2. Haematopoietic Stem Cell Lab, Wellcome Trust Institute, Cambridge, UK.

The first challenge we had to face, in order to model haematopoiesis (*i.e.* blood cell differentiation), was to encode all the expertise and knowledge acquired over many years of methodical experimentation into a single formal representation. This encoding procedure is traditionally done by bioinformaticians with little experience in the application domain (in our case haematology) but good knowledge of the theoretical framework used to model biological processes. However, when the application domain is extensive and exceedingly complex, as it is often the case in biology, it becomes likely that the resulting models no longer represent faithfully the current status of the knowledge domain. Therefore, instead of trying ourselves to assimilate and interpret the enormous amount of information available in the Göttgens group or the literature, we decided to explain the basics of our framework to the experimental biologists, leveraging on the simplicity of our formalism, enabling them to encode their knowledge directly using our Petri net representation.

Although the members of the Göttgens group had no previous experience whatsoever with Petri nets or other models of concurrency, they were able to draw on the whiteboard (see Fig. 4) a first draft of the GRN that regulates the haematopoietic stem/progenitor cell differentiation in a matter of minutes. The formalisation process proved useful as well for spotting latent ambiguities in the knowledge shared within the field, such as controversial hypotheses regarding the effect of certain interactions between regulatory elements. After only a few iterative drawing and discussion cycles, the Göttgens group members were able

to draw the current state of the art of the haematopoietic differentiation GRN [10]. We could then easily implement the GRN sketched on the whiteboard using our custom-made tools. The resulting Petri net (Fig. 5) comprises 11 densely connected genes. At the heart of the network lies the triad of Scl, Gata2 and Fli1, which is characterized by extensive positive feedback loops, albeit negative regulatory interactions are common outside this central triad.

In our experience, the possibility to draw the network model on a whiteboard (Fig. 4), which then directly provides a formal and full specification of the underlying model (Fig. 5) lowered significantly the access barrier to the formalism, compared to a model that can only be formally specified using sets of equations. The consistent use of transitions nodes in Fig. 4 shows the appreciation of the additional value provided by Petri net transitions over more classic gene regulatory graphs. Being able to visualize the model on the whiteboard in the form of a graph rather than a system of equations was instrumental also to identify patterns that guided biological intuitions (such as the functional similarities between Gata2 and Fli1) which were explored later using computational approaches. Furthermore, the one-to-one correspondence between Boolean formulas and network topologies allows to rapidly re-encode the model in other formulas-based boolean formalisms such as GenYsis [13] as shown in [10].

In order for a network model to be useable as a predictive tool, its behaviour needs to be assessed using available experimental data. We therefore explored the expression patterns of the 11 regulatory genes and related these patterns to the various haematopoietic cell types. We computed the state space of our model and we performed a TSCC analysis as described in Sect. 3. Our experimentally validated network allows for three biologically sound TSCCs (*i.e.* TSCCs composed by states which markings resemble the gene expression of real blood cell types): (i) all genes are off, (ii) only Gata1 and Scl are expressed and (iii) a TSCC formed by 32 interconnected markings with multiple genes active but Gata1 always repressed. TSCC (i) corresponds to a non-haematopoietic cell. TSCC (ii) matches the erythroid cell profile. Most interesting, however, is TSCC (iii), which is composed of 32 interconnected internal states, including a state that matches the expected pattern for HSPCs. This analysis, therefore, not only demonstrates that our knowledge-driven network topology is compatible with expression patterns observed in HSPCs *in vivo*, but also suggests that the HSPC is not a homogeneous cell population; instead it is composed from cells in different stages of activation, as proved experimentally [13]. Traditional networks derived from gene expression data provide a population average which offers little insight into cellular heterogeneity and the regulatory processes likely to be critical for lineage commitment. Therefore, characterization and modelling of single cell heterogeneity is a necessary step to understand differentiation not only at intra-cellular level but also at population level, where a heterogenous population of stem cells is responsible for macroscopic changes at the level of an organism.

Analysis of transitions between different states in the model state space can be useful to predict experimental conditions for cells to differentiate out of the

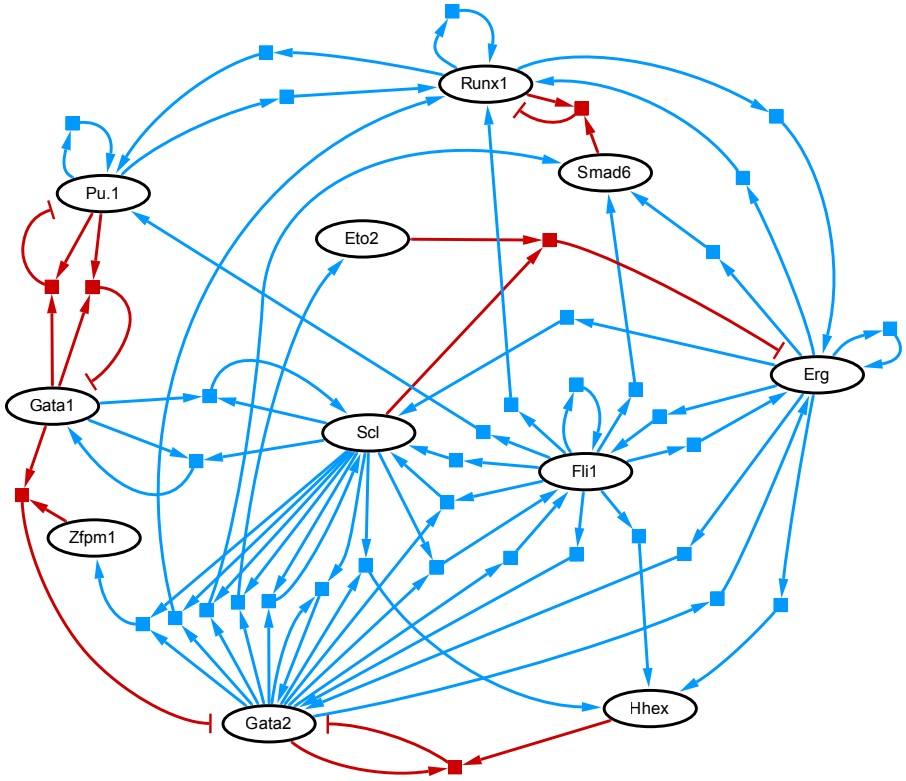

Figure 5. Petri net of the haematopoietic gene regulatory network. Activating interactions are shown as blue arrows, repressing interactions in red with flat arrow heads. All regulatory information encoded in this model can be found in [10].

HSPC state. Therefore we performed a reachability analysis between each state and each state matching a mature cell types in the developmental tree. Although there are no paths out of the HSPC state, which is consistent with the HSPC being a TSCC, we determined which are the closest states to the HSPC connected to a state matching a mature cell type. For example, the closest state to the HSPC that can reach the the state matching the erythrocyte cell type is at a distance of two from the HSPC, where distances are measured using the Hamming distance between state bit vectors. This observation corresponds to the notion that the transition from HSPC to erythrocyte would need at least to change the marking of two genes in the HSPC marking in order to be in a state able to reach the erythrocyte state.

Of note, experimental evidence suggests that a single change (*e.g.* ectopic expression of Gata1) would be sufficient to drive immature blood progenitors towards an erythroid fate. However, as noted above, our modelling results suggest that HSPC cells need to change at least the marking of two genes from the HSPC state. We therefore considered potentially missing network links from our current topology. In particular, we extended our model by introducing a new transition, the possible repression of Fli1 by Gata1, based on the rationale that the Fli1 regulatory element is structurally similar to the Gata2 element, which is known to be repressed by Gata1 [14]. Interestingly, just introducing this single additional repressive transition the reachability analysis of the updated network revealed that a single change in the HSPC marking (*i.e.* Gata1 marking, from 0 to 1) would be sufficient to reach the erythrocyte state.

Following on from this modelling result, an experiment was conducted in Göttgens lab which confirmed the novel inhibition on the haematopoietic progenitor cell line HPC7.

Our Petri net framework, therefore, allowed us to predict a previously unrecognized network link, which we were able to validate experimentally. Due to their preeminent biological nature, these results are discusses in more details in a companion publication [10].

5 Related work

Formal models can be an excellent way to store and share knowledge on biological systems, and to reason about such systems [15]. Petri nets in particular are among the most used formal frameworks; for a review see [16,17]. Petri nets can be used to build both qualitative and quantitative biological models [18]. However, due to the lack of biochemical parameters, GRNs are usually described in qualitative terms. The seminal work of Kauffman *et al.* [19] showed that GRNs can be abstracted as Boolean networks, while Chaouiya *et al.* [7] showed how to encode Logical Regulatory Graphs [20] using different Petri net dialects. We built on the fundamental work of Chaouiya and colleagues. We aimed to build a simple formal representation based on PT nets that would resemble the cartoons used in experimental labs, and in which preconditions are explicitly shown (as opposed to Logical Regulatory Graphs). We followed these design principles in order to

facilitate the collaboration between computer scientists and biologists and foster the use of formal methods in the experimental biology practice. Furthermore, in Sect. 2 we introduced a novel method to capture and automatically derive degradation processes in GRNs that, in our experience, proved to be crucial to faithfully simulate the regulation of gene expression.

The biological questions that drove the investigation of the haematopoietic stem cell differentiation were addressed using the state space analysis described in Sect. 3. However, it is reasonable to think that traditional analysis techniques applied to PT nets (*e.g.* T- and P-invariants analysis) could also be lifted within our framework. Some of these techniques have been shown to be useful also in the context of GRNs [8].

Although we used *ad hoc* tools for our analysis, several Petri net based tools, designed specifically for biological purposes, became recently available. Some of the most complete, popular, and stable are [21,22,23], and some interesting case studies have been already explored with these tools [8,24,25]. This is a crucial first step towards the adoption of formal models in the every day workflow of experimental labs. However, in our experience, Petri net tools can also be too general or too technical.

6 Conclusions

Biology, like computer science, is a very broad field in which each sub-community has profoundly diverse interests and a specialised terminology. Therefore, frameworks and tools that aim to be too general, targeting too many types of biological processes, tend to have an excessive complexity and therefore be unpractical. Furthermore, sometimes, tools use a computer science jargon which does not relate with the specialised vocabulary used by experimental biologists. We argue that the next generation of formalism and tools should be designed around specific biology domains (*e.g.* gene regulation, or signal transduction, or genome analysis) and their terminology should be adapted to that specific domain, allowing a natural transition from the lab bench to the computer model. To achieve this objective it will be essential to cooperate with experimental biologists through all phases of development; from the definition of the formalism to the conception of the user interface.

7 Acknowledgements

Parts of this work have been supported by ENFIN; a Network of Excellence funded by the European Commission within its FP6 Program, under the thematic area 'Life Sciences, genomics and biotechnology for health', contract number LSHG-CT-2005-518254.

References

1. World Health Organization: World Health Statistics 2012. (2012)

2. Koch, I., Junker, B.H., Heiner, M.: Application of petri net theory for modelling and validation of the sucrose breakdown pathway in the potato tuber. Bioinformatics **21**(7) (2005) 1219–1226
3. Gilbert, D., Heiner, M., Lehrack, S.: A unifying framework for modelling and analysing biochemical pathways using petri nets. In: Computational Methods in Systems Biology. Volume 4695 of Lecture Notes in Computer Science. Springer-Verlag (2007) 200–216
4. Bonzanni, N., Krepska, E., Feenstra, K.A., Fokkink, W., Kielmann, T., Bal, H., Heringa, J.: Executing multicellular differentiation: quantitative predictive modelling of c. elegans vulval development. Bioinformatics **25**(16) (2009) 2049–2056
5. Bonzanni, N., Zhang, N., Oliver, S.G., Fisher, J.: The role of proteosome-mediated proteolysis in modulating potentially harmful transcription factor activity in Saccharomyces cerevisiae. Bioinformatics **27**(13) (2011) I283–I287
6. Steggles, L.J., Banks, R., Shaw, O., Wipat, A.: Qualitatively modelling and analysing genetic regulatory networks: a petri net approach. Bioinformatics **23**(3) (2007) 336–343
7. Chaouiya, C., Remy, E., Ruet, P., Thieffry, D.: Qualitative modelling of genetic networks: From logical regulatory graphs to standard petri nets. In Cortadella, J., Reisig, W., eds.: Applications and Theory of Petri Nets. Volume 3099 of Lecture Notes in Computer Science. Springer-Verlag (2004) 137–156
8. Grunwald, S., Speer, A., Ackermann, J., Koch, I.: Petri net modelling of gene regulation of the duchenne muscular dystrophy. Biosystems **92**(2) (2008) 189–205
9. Matsuno, H., Doi, A., Nagasaki, M., Miyano, S.: Hybrid petri net representation of gene regulatory network. In: Pacific Symposium on Biocomputing. Volume 5. (2000) 338–349
10. Bonzanni, N., Garg, A., Feenstra, K.A., Schütte, J., Kinston, S., Miranda-Saavedra, D., Heringa, J., Xenarios, I., Göttgens, B.: Hard-wired heterogeneity in blood stem cells revealed using a dynamic regulatory network model. Bioinformatics **29**(13) (2013) i80–i88
11. Tarjan, R.: Depth-first search and linear graph algorithms. SIAM Journal on Computing **1**(2) (1975) 146–160
12. Krepska, E.: Towards Big Biology: High-Performance Verification of Large Concurrent Systems. PhD thesis, VU University Amsterdam (2012)
13. Garg, A., Xenarios, I., Mendoza, L., DeMicheli, G.: An efficient method for dynamic analysis of gene regulatory networks and inÂăsilico gene perturbation experiments. In Speed, T., Huang, H., eds.: Research in Computational Molecular Biology. Volume 4453 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2007) 62–76
14. Grass, J.A., Boyer, M.E., Pal, S., Wu, J., Weiss, M.J., Bresnick, E.H.: Gata-1-dependent transcriptional repression of gata-2 via disruption of positive autoregulation and domain-wide chromatin remodeling. Proceedings of the National Academy of Sciences **100**(15) (2003) 8811–8816
15. Bonzanni, N., Feenstra, K.A., Fokkink, W., Krepska, E.: What can formal methods bring to systems biology? In: Formal Methods. Volume 5850 of Lecture Notes in Computer Science. Springer-Verlag (2009) 16–22
16. Will, J., Heiner, M.: Petri nets in biology, chemistry, and medicine - bibliography. Technical Report 04/2002, BTU Cottbus, Computer Science (2002)
17. Chaouiya, C.: Petri net modelling of biological networks. Briefings in Bioinformatics **8**(4) (2007) 210

18. Heiner, M., Gilbert, D., Donaldson, R.: Petri nets for systems and synthetic biology. In Bernardo, M., Degano, P., Zavattaro, G., eds.: Formal Methods for Computational Systems Biology. Volume 5016 of Lecture Notes in Computer Science. Springer-Verlag (2008) 215c–264
19. Kauffman, S., Peterson, C., Samuelsson, B., Troein, C.: Random boolean network models and the yeast transcriptional network. Proceedings of the National Academy of Sciences of the United States of America **100**(25) (2003) 14796–14799
20. Thomas, R.: Regulatory networks seen as asynchronous automata: A logical description. Journal of Theoretical Biology **153**(1) (1991) 1–23
21. Heiner, M., Herajy, M., Liu, F., Rohr, C., Schwarick, M.: Snoopy âĂŞ a unifying petri net tool. In Haddad, S., Pomello, L., eds.: Application and Theory of Petri Nets. Volume 7347 of Lecture Notes in Computer Science. Springer-Verlag (2012) 398–407
22. Gonzalez, A.G., Naldi, A., Sánchez, L., Thieffry, D., Chaouiya, C.: Ginsim: A software suite for the qualitative modelling, simulation and analysis of regulatory networks. Biosystems **84**(2) (2006) 91–100
23. Nagasaki, M., Saito, A., Jeong, E., Li, C., Kojima, K., Ikeda, E., Miyano, S.: Cell illustrator 4.0: A computational platform for systems biology. In Silico Biology **10**(1–2) (2010) 5–26
24. Marwan, W., Rohr, C., Heiner, M.: Petri nets in snoopy: A unifying framework for the graphical display, computational modelling, and simulation of bacterial regulatory networks. In Helden, J., Toussaint, A., Thieffry, D., eds.: Bacterial Molecular Networks. Volume 804 of Methods in Molecular Biology. Springer-Verlag (2012) 409–437
25. Doi, A., Nagasaki, M., Matsuno, H., Miyano, S.: Simulation-based validation of the p53 transcriptional activity with hybrid functional petri net. In Silico Biology **6**(1) (2006) 1–13