Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, eds.

A MACHINE LEARNING APPROACH FOR GENERATING TEMPORAL LOGIC CLASSIFICATIONS OF COMPLEX MODEL BEHAVIOURS

Daniele Maccagnola
Enza Messina

University of Milano-Bicocca
DISCo - U14 building- Viale Sarca 336
20126 Milano, ITALY

Qian Gao
David Gilbert

Brunel University
St.John's Building
UB8 3PH Uxbridge, Middlesex, UK

ABSTRACT

Systems biology aims to facilitate the understanding of complex interactions between components in biological systems. Petri nets (PN), and in particular Coloured Petri Nets (CPN) have been demonstrated to be a suitable formalism for modelling biological systems and building computational models over multiple spatial and temporal scales. To explore the complex and high-dimensional solution space over the behaviours generated by such models, we propose a clustering methodology which combines principal component analysis (PCA), distance similarity and density factors through the application of DBScan. To facilitate the interpretation of clustering results and enable further analysis using model checking we apply a pattern mining approach aimed at generating high-level classificatory descriptions of the clusters' behaviour in temporal logic. We illustrate the power of our approach through the analysis of two case studies: multiple knockdown of the Mitogen-activated protein-kinase (MAPK) pathway, and selective knockout of Planar Cell Polarity (PCP) signalling in Drosophila wing.

1 INTRODUCTION

Systems biology is an interdisciplinary field which aims at a system-level understanding of biological systems. It focuses on the study of complex interactions (networks) rather than their individual molecular components. A biological system typically comprises large numbers of different elements which can interact selectively and nonlinearly to produce complex coherent behaviours. This complexity makes it impossible to intuitively understand the behaviour of the system.

Mathematical models enable us to investigate the dynamic behaviour of such systems by allowing the construction of *in-silico* experiments to test various hypotheses and predict behaviours which can subsequently be tested in vitro and/or in vivo. Among them, Petri Nets (PN) (Gilbert, Heiner, and Lehrack 2007) not only provide a graphical representation of the system but also enable us to model biological systems in both qualitative and quantitative settings. PN simulation can generate time series data of the concentration of each molecular species to be used for a behavioural comparison of different variants of models created by altering one or more parameters. As the number of these variants increases the analysis can be too challenging to be handled manually and the development of tools for the automatic analysis and characterization of model behaviours is becoming essential. In this paper we present a new clustering framework for the analysis of time series data generated by computational models of biological systems in order to detect analogous behaviours of different models.

Unsupervised clustering is helpful to identify groups of similar behaviours; however it does not provide an explicit description of the characteristics of each class and the semantic meaning of each cluster is left to subsequent analysis, often performed manually. To facilitate this interpretation, we have developed a procedure to automatically generate semantic descriptions of the time series belonging to each cluster, based on the temporal logic language PLTLc (Probabilistic Linear-time Temporal Logic with constraints

978-1-4673-4781-5/12/$31.00 ©2012 IEEE

– see Donaldson and Gilbert (2008)). The method can be thought of as a Temporal Logic (TL) feature generation method to identify time series with similar behaviours. The problem of selecting the TL features which identify time series with similar behaviours has been addressed in Batal et al. (2011). In that work the most relevant features are selected starting from a complete TL description of the time series. We instead address the problem of TL feature generation starting from numerical data.

The paper is structured as follows: Section 2 gives a description of the two case studies that we employed to test the validity of our algorithms. In Section 3 we describe our new clustering methodology and its validation method, while Section 4 describes PLTLc and the automatic generation algorithm. Section 5 presents the results of the tests on the case studies.

2 MOTIVATIONAL CASE STUDIES

In this paper we illustrate the potential power of our approach by applying it to analyse models of two biological exemplars, the Mitogen-activated protein kinase (MAPK) pathway and Planar Cell Polarity (PCP) signalling in Drosophila wing. The first is a benchmark case study which illustrates the complexities introduced by multiple variants of a single-scale model and we used it to demonstrate the power of our approach. The second case study integrates spatial scales ranging from the intra-cellular to the tissue level and has been used to show the applicability of the proposed approach to more complex multiscale models. The Mitogen-activated protein kinase (MAPK) pathway is one of the most significant intracellular signal transduction pathways, which passes information from the cell surface to the nucleus. This signal transduction pathway is involved in a number of core cellular processes, and is constitutively activated by mutations in many human tumours (Hoshino et al. 1999), hence playing a major role in drug target discovery. The Hornberg et al. (2005) model that we use describes the MAPK signal transduction pathway starting from EGF binding at the cellular membrane to the phosphorylation of ERK, and integrates knowledge from previous models (Kholodenko et al. 1999; Schoeberl et al. 2002). It comprises 148 molecular processes (reactions) described with mass-action kinetics, and 103 variable molecular species (proteins & complexes). RNA interference (RNAi) (Morris 2008) is a mechanism which can be used to inhibit the expression of specific genes, and can be used to investigate the functioning of biochemical networks. Using this method it is possible to knockdown in a graded manner one or more genes in the MAPK pathway; however the number of variations to be analyzed rapidly increases with the number of simultaneously knocked down species. The challenge is to cluster the behaviours of a selected read-out species (e.g. doubly-phosphorylated ERK) and then to interpret the behaviour of these clusters.

Planar cell polarity (PCP) is a mechanism which controls the orientation of cells within a sheet of cells and occurs in most cell types and tissues. Defects in PCP in vertebrates are responsible for developmental abnormalities in multiple tissues including the neural tube, the kidney and the inner ear (reviewed in Simons and Mlodzik (2008)). The signalling mechanisms underlying PCP have been studied most extensively in the epithelia of the fruit fly *Drosophila melanogaster* including the wing, the abdomen, the eye, and the bristles of the thorax. The adult *Drosophila* wing comprises approximately 30,000 hexagonal cells each of which contains a single hair pointing in an invariant distal direction. This hair is extruded from the membrane at the distal edge of each the cell during pupal development, at the conclusion of PCP signalling. The proteins involved are thought to mediate the cell-cell communication that is part of the core machinery of PCP signalling and to be involved in establishing the molecular asymmetry within and between cells which is subsequently transformed into distal polarisation of the wing hairs (reviewed in (Strutt 2002)). The mechanism of PCP can be investigated *in-vivo* by studying the disruptive effect on hair polarity by the introduction of certain mutations into areas (clones) in the wing by mutagenesis. When modelling this situation we need to consider a sufficiently large number of cells and we use the Flamingo-Frizzled-Dishevelled complex (FFD) as a proxy for the hairs. In (Gao et al. 2011) we have constructed a multiscale hierarchically coloured Petri net (HCPN) model to generate an in-silico patch (tissue) based on a honeycomb grid of 112 cells. Each cell is sub-divided into one central and 6 sub-membrane virtual compartments in order to model intracellular locality, see Fig.2. The underlying unfolded continuous PN

model comprises 8,624 species (places) and 9,184 reactions (transitions), and thus the system of ordinary differential equations (ODEs) to be analysed comprises 8,624 equations, because each species is modelled by one equation. As in the previous case study, we wish to cluster and interpret the behaviour of the variant models representing the $6 * 112 = 672$ sub-membrane compartments.

3 CLUSTERING MODEL BEHAVIOURS

In this section we describe the clustering technique used to automatically identify sets of homogeneous model behaviours. We assume that the system behaviour is described by (possibly multivariate) time series obtained through simulation. These unlabeled time series could comprise monitoring data collected during different periods $t = 1 \dots T$ from a particular process ($m = 1$) or from more than one process ($m > 1$). A time series $X(t)$ is a series of observations, $x_i(t), i = i \dots m, t = 1 \dots T$, made sequentially through time, univariate when $m = 1$ or multivariate when $m > 1$.

Clustering algorithms consider a set D of n unlabeled data tuples $D = X_j(t), j = 1...n, t = 1..T$, where $X_j(t)$ is the time series (univariate or multivariate) represented by the j_{th} element of D. The purpose of clustering is to identify structure in D by objectively organizing data into homogeneous groups. Here the notion of homogeneity means that the within-group-object similarity and the between-group-object dissimilarity are maximised.

Various algorithms have been developed to cluster different types of time series data, see (Liao 2005). We can roughly divide time series clustering approaches into two categories: those which try to modify the existing algorithms for clustering static data in such a way that time series data can be handled, in this case the main effort consists in the definition of suitable distance/similarity measures for time series, and those which convert time series data into a feature vector of lower dimension so that the existing algorithms for clustering static data can be directly used. Our algorithm follows the latter approach, by using a feature reduction method before applying a density-based clustering method.

The input dataset in our examples comprises raw data generated by simulation runs in the form of time series of a fixed (possibly very large) number of time points. Clustering directly on this raw data would be computational demanding and subject to a high degree of noise. In order to overcome this problem our clustering methodology is based on calculating the degree of similarity using Principal Component Analysis (PCA), distance similarity and density factors.

3.1 Background on Principal Component Analysis

Principal Component Analysis (PCA) is an exploratory multivariate statistical technique for simplifying complex data sets represented by a matrix $X \in \Re^{N \times T}$. It is aimed at finding R new variables (called *Principal Components*), where $R \ll$ the original number of variables T. In the case of time series, T represents the total time and N is the number of elements observed during that time.

Each principal component is a linear combination of the original variables and is derived in such a manner that its successive component will account for a smaller portion of variation in X. The derivation of the new components is based on the covariance matrix C_X of X. Each eigenvector (e_j) provides the component weight a_{ij} of the Y_j component and the corresponding eigenvalues (λ_j) provide the variance of this component. Therefore, we can sort the eigenvalues λ_j in descending order and retain only the first R whose corresponding variables will account for as much as possible of the variance present in the original T-dimensional dataset, while remaining mutually uncorrelated and orthogonal. R is chosen as the minimum value which satisfies $min_R : \frac{\sum_{i=1}^{R} \lambda_i}{\sum_{j=1}^{T} \lambda_j} \geq \tau$, where τ is usually 90% or 95%.

3.2 Density-based Clustering

To the best of our knowledge, three categories of clustering methods, i.e. partitioning methods, hierarchical methods, and model-based methods, have been utilised directly or modified for time series clustering (Golay

et al. 1998; Van Wijk and Van Selow 1999). Partitioning and hierarchical methods suffer from the inability to identify non-spherical clusters, and their performance is usually affected by noisy data. This is particularly true when analysing biological data, as highlighted in (Fersini et al. 2010). Model-based approaches, instead, assume that each time series has been generated by some kind of stochastic model or by a mixture of underlying probability distributions. A visual inspection of our data showed that clusters of common behaviours were characterised by non-spherical shapes. Moreover, our time series exhibit rather smooth behaviour, partly due to being the results of deterministic simulation. Therefore the application of clustering procedures belonging to the above categories did not lead to any useful results as shown in Section 5.

This problem can be overcome by another category of clustering methods, called *Density-based methods*, which analyse the cluster detection space, looking for areas where the density (number of elements or data points) exceeds some threshold. For these reasons we focused on a density-based method called DBScan (Density-Based Spatial Clustering of Applications with Noise) (Ester et al. 1996) which is well suited to cluster data of arbitrary shape. The concept of density is based on two parameters, which define the neighbourhood of a point: the radius Eps of the neighbourhood and the minimum number of points $MinPts$ needed to consider a neighbourhood as *dense*. The neighbourhood is defined as the set of points N_{Eps} which are distant Eps or less from a point p.

In order to find a cluster, DBScan starts with an arbitrary point p; if $N_{Eps}(p)$ has at least $MinPts$ points, then p is labeled as a "core" point and the algorithm iterates over the points in its neighbourhood. In this way, DBScan generates a set of core points connected by their neighbourhood, which is considered as a cluster. If a point is not labeled as "core", but is in the neighbourhood of a core point, then it is labeled as a "border" point and included in the cluster. When all the points have been examined, and a point is not yet part of any cluster then it will be labeled as a "noise" point and treated as an outlier.

Please refer to (Ester et al. 1996) for further details on DBScan.

3.3 Cluster Evaluation

It is generally accepted that a good clustering algorithm should aim at maximising both the separation between the clusters and the compactness within clusters. Nevertheless, different clustering algorithms may lead to qualitatively different results whose significance is difficult to evaluate. The application domain and its specific requirements have to be taken into account for define an appropriate cluster validity evaluation index. Several validity indexes appear in literature which are well suited in cases when the cluster shape is spherical (or hyper-spherical), including the validity index by Dunn (1974) and the Silhouette validation technique (Rousseeuw 1987). They approximate the measure of compactness and separation through the definition of a cluster centre and therefore fail to capture the validity of the results when the dataset contains clusters with arbitrary shape, which is the case in our study. Halkidi and Vazirgiannis (2008) introduced a new validity index, called CDbw (Composed Density between and within clusters) which measures the quality of the clusters by considering multiple representative points per cluster and therefore accommodates non hyper spherical cluster geometry. The validity index CDbw is defined as the product of three main indices, namely compactness, cohesion and separation.

$$CDbw = \text{Compactness} \cdot \text{Cohesion} \cdot \text{Separation}$$

where *Compactness* measures the density within each cluster, *Cohesion* measures the changes of density distribution within clusters, and *Separation* measures the separation of clusters (for more details, refer to (Halkidi and Vazirgiannis 2008)). The higher the CDbw value, the better the quality of clusters. CDbw offers a valuable support in evaluating clustering results and we employ this measure both for tuning the proposed clustering algorithm and for a comparative evaluation of different clustering techniques.

4 TEMPORAL LOGIC DESCRIPTIONS

Temporal Logics are systems of rules and symbols which can be used to represent and reason about propositions qualified in terms of time. In particular, classical propositional and predicate logic, which have

truth-functional logical operators ($\neg$, $\vee$, $\wedge$ $\rightarrow$) can be extended with temporal modal operators. Linear-time Temporal Logic (LTL) (Pnueli 1981) is the fragment of full Computational Tree Logic (CTL*) (Clarke, Grumberg, and Peled 2001) without path quantifiers, implicitly universally quantifying over all paths. LTL has been introduced in a probabilistic setting in (Baier 1998), and extended by numerical constraints over real value variables in (Fages and Rizk 2007). PLTLc (Donaldson and Gilbert 2008) combines both extensions, complemented by the filter construct as used in Probabilistic Computational Tree Logic (PCTL) (Hansson and Jonsson 1994) and Continuous Stochastic Logic (CSL) (Aziz, Sanwal, Singhal, and Brayton 1996). The syntax and semantics of PLTLc have been described in (Donaldson and Gilbert 2008); here we give a short summary of those features which we have used in this research. The temporal operators employed in PLTLc are:

- Next: $X\phi$: ϕ holds at the next state.
- Until: $\phi U \psi$: ψ holds at the current or a future position in time, and ϕ holds until that time.
- Release: $\phi R \psi$: ϕ releases ψ if ψ is true until the first position in time in which ϕ is true (or forever if such a position does not exist).
- Finally: $F\phi$: ϕ eventually holds (somewhere on the subsequent path).
- Globally: $G\phi$: ϕ holds for the entire path from the current state onwards.

LTLc provides the ability to define any function returning a real or integer value. We have chosen to use *d(Variable)*, parameterised by a state, which returns the derivative of the *Variable* in each state, thus increasing and decreasing species values can be expressed: $d(Protein) > 0$ and $d(Protein) < 0$ respectively.

PLTLc enhances LTLc by the inclusion of a probability operator and filter construct. The top-level definition of PLTLc is $\psi ::= \mathbf{P}_{\trianglelefteq x}[\,\phi\,]$, where ϕ is an LTLc expression. Given that $\trianglelefteq \in \{>, \geq, <, \leq\}$ and $x \in [0,1]$, $P_{\trianglelefteq x}$ is any inequality comparison of the probability of the property holding true, for example $P_{\geq 0.5}$. We also permit the expression $P_{=?}$ which returns the value of the probability of the property holding true.

The semantics of PLTLc is defined over a finite set of finite paths through the system's state space – stochastic or deterministic simulations, or time series data recorded in wet lab experiments. In this work we consider only results of deterministic simulations; however these are typically derived from multiple model variants due to the nature of our case studies, each variant generating one trace. Because we do not average over the variant traces but consider them together as a set, we exploit the probabilistic capabilities of PLTLc which permit model checking over sets of traces rather than single traces; the probabilities are used during the evaluation process (see Section 4.1 below).

4.1 Automatic Generation of Temporal Logic Descriptions

In order to semantically decode the clusters obtained containing time series which exhibit similar behaviours, we developed a pattern learning algorithm to automatically generate formal statements in PLTLc characterizing each cluster. To enable the adequate description of a cluster, a PLTLc statement should be *general* enough to describe all the time series in the cluster, and simultaneously *discriminative* enough to distinguish between time series of different clusters.

Let $\phi(C)$ denote a statement for describing a single cluster C, then the optimal statement ϕ_{opt} should maximise $P_{=?}[\phi\,(C)\,]$ while minimising $P_{=?}[\phi\,(\neg C)\,]$, where $\neg C$ denotes the set of time series not belonging to C.We start by defining a set of property patterns, which act as templates needed to describe the behaviour of one time series:

- **Trend**: describes the trend of a time series as a sequence of directions, *increase*, *steady* and *decrease* determined respectively by derivatives $d[X] > 0$, $d[X] = 0$ and $d[X] < 0$. The Trend is a chain of directions connected by the temporal operator *Until* (U). The last direction continues until the end of the time series, therefore to describe it we use the operator *Globally* (G).

$$\phi_1 U(\phi_2 U(\dots U(\phi_{m-1} U(G(\phi_m)))\dots))$$

where m is the number of directions in the time series, and $\phi_j = d(X) \trianglelefteq 0$ (where $\trianglelefteq$ is $\{=, <, >\}$).

- **Time**: identifies the series of m time points when the time series changes its direction.
- **Extrema**: represents the sequence of value and time of occurrence of all the local minima and maxima of a time series. We used the temporal operator *Future* (F) to describe this property.
- **Steady State**: is the value of the time series steady state (if any). A time series has a steady state if its last local trend is steady.

Now we can derive the properties of a cluster C_i of time series by generalizing the properties above in such a way that (almost) all the time series in the cluster can be represented by these properties.

- **Trend(C_i)**: to define the trend $Trend(C_i)$ of a cluster C_i we compute the $Trend(X)$ of all the time series $X \in C_i$, and order them by their frequency. We denote by F_i^0 the most frequent trend in C_i, with F_i^1 the second most frequent and so on. We denote by C_i^j the set of time series $X \in C_i$ such that $Trend(X) = F_i^j$. Trend(C_i) is given by $F_i^0 \vee F_i^1 \vee \dots \vee F_i^j \dots$.
- **Time(C_i)**: time series in C_i having the same Trend may have different Time patterns because the direction changes can occur at different times. Therefore, for each $C_i^j \in C_i$ we compute for each direction change the time interval which contains all the times in which the change may occur. The result is, for each trend F_i^j of C_i, a set of intervals $(\check{t}_l, \hat{t}_l)$, with $l = 1 \dots m$.
- **Extrema(C_i)**: is defined as the sequence of time and value intervals, $(\check{t}_k, \hat{t}_k)$ and $(\check{X}_k, \hat{X}_k)$ respectively, which represent the lower and upper bounds of the k ordered extrema of the time series $X_j \in C_i^0$.
- **Steady State(C_i)**: To define the steady state of a cluster, we consider the range of steady state values of all the time series $X \in C^0$, which we denote by the interval $(\check{s}, \hat{s})$.

The PLTLc description for each cluster C_i is then generated using the following procedure:

1. Consider cluster C_i and the set of remaining clusters $\neg C_i$. We compare the most frequent trend F_i^0 of C_i with the most frequent trend $F_{\neg i}^0$ of $\neg C_i$. If these trends are different, the PLTLc statement is generated using the description of all the trends F_i^j in C_i, connected by the symbol $\vee$.

$$\begin{aligned}\phi_{opt}(C_i) &= F_i^0 \vee F_i^1 \vee \dots \vee F_i^j \dots \\ &= \bigvee_j (\phi_1(F_i^j) U(\phi_2(F_i^j) U(\dots U(\phi_{m_j-1}(F_i^j) U(G(\phi_{m_j}(F_i^j)))\dots))))\end{aligned}$$

where $\phi_l(F_i^j) : d(X) \trianglelefteq 0 \wedge Time \geq \check{t}_l^j \wedge Time \leq \hat{t}_l^j$. Else, continue to step 2.

2. If $F_i^0 == F_{\neg i}^0$, then we consider the times when the time series changes direction, i.e. $Time(C_i)$ and $Time(\neg C_i)$, and we compute two series of l time intervals $(\check{t}_l^{C_i}, \hat{t}_l^{C_i})$ and $(\check{t}_l^{\neg C_i}, \hat{t}_l^{\neg C_i})$ which represent the average of the lower and upper bounds of the values in $Time(C_i)$ and $Time(\neg C_i)$. If their difference is greater than a threshold η_{time} for at least one l, then the PLTLc statement is:

$$\begin{aligned}\phi_{opt}(C_i) &= F_i^0 \vee F_i^1 \vee \dots \vee F_i^j \dots \\ &= \bigvee_j (\phi_1(F_i^j) U(\phi_2(F_i^j) U(\dots U(\phi_{m_j-1}(F_i^j) U(G(\phi_{m_j}(F_i^j)))\dots))))\end{aligned}$$

as in step 1. Else, continue to step 3.

3. In this step we compare Extrema(C_i^0) with Extrema($\neg C_i^0$). Note that, we consider only the extrema of the most frequent trends, which by construction must have the same number of extrema equal to K. For each extremum k, we compute the average value of $(\check{X}_k, \hat{X}_k)$ for C_i and $\neg C_i$. If their difference is greater than a threshold η_{ex} for at least one k, then the PLTLc statement is:

$$\phi_{opt}(C_i) = F(\psi_1) \wedge F(\psi_2) \wedge \dots \wedge F(\psi p) \wedge \phi_1(F_i^0) \dots$$

where $\psi_k : X \geq \check{X}_k \wedge X \leq \hat{X}_k \wedge Time \geq \check{t}_k \wedge Time \leq \hat{t}_k$. Otherwise, continue to step 4.
This procedure ensures a good tradeoff between specificity and generality for the resulting PLTLc statements. Indeed, the all-against-all comparison of trends F^j would be not only computationally demanding but would also generate too general PLTLC statements.

4. In this step we compare Steady State(C_i) and Steady State($\neg C_i$). We compute the average value of ($\check{s}$, $\hat{s}$) for C_i and $\neg C_i$. If the difference is greater than a threshold η_{ss}, we define the PLTLc statements as:

$$\phi_{opt} = \phi_1(F_i^0)U(\phi_2(F_i^0)U(\ldots U(\phi_{m-1}(F_i^0)U(G(\phi_m(F_i^0))))\ldots))$$

where $\phi_m(F_i^0) : d(X) = 0 \wedge X \geq \check{s} \wedge X \leq \hat{s}$. Otherwise, it means that we cannot discriminate between C_i and the other clusters.

Note that the effectiveness of the PLTLc generation algorithm depends on the cluster's quality rather than on the number of time series, because it is affected by the variety in the number and times of the direction changes of the time series. Moreover, the effectiveness is affected by the thresholds η_{time}, η_{ex} and η_{ss}, whose values have to be chosen empirically.

To evaluate $\phi_{opt}(C_i)$ we use the probability $P_{=?}[\phi_{opt}(C_i)]$ that the statement correctly classifies the time series belonging to C_i, and we associate a *confidence level Conf* to this probability value defined as:

$$Conf(\phi_{opt}(C_i)) = \frac{P_{=?}[\phi_{opt}(C_i)]}{1 + max_{j \neq i} P_{=?}[\phi_{opt}(C_j)]}$$

which represents the capability of a statement ϕ to discriminate between $X_l \in C_i$ and $X_l \in C_j$ with $j \neq i$. *Conf* can vary between 0 (for an entirely wrong statement) and 1 (for an entirely correct statement).

In order to evaluate the quality of PLTLc statements, we employed the simulative model checking tool MC2 (Donaldson and Gilbert 2008) to compute the probability that a statement correctly describes the behaviour of each cluster C_i. We developed the machine learning procedure, and performed the evaluation of the model checking results using MATLAB (2010). The model for the first study was directly encoded in MATLAB while the model for the second case study was developed in coloured Petri nets using the Snoopy (Heiner et al. 2012) tool.

5 RESULTS

5.1 EGF-induced MAPK Cascade Model

In order to demonstrate the feasibility of our approach, we first chose the EGF signaling cascade, described in Section 2, analysing the behaviours of the model under different perturbations by inducing diverse knock-downs to the signalling pathway. This case study is thus an example of a uniscale model with many variants which exhibit a set of complex behaviours. We analysed the behaviours of the model under different perturbations by inducing diverse knock-downs to the signalling pathway. The molecular species chosen to knock down are listed in Table 1. We used the initial settings of molecular species and parameter values given in (Hornberg et al. 2005) to set up our PN model. In order to investigate the effect of gene knock-downs, we first manipulated single knock-downs of 11 selected molecular species by decreasing the initial concentration (marking) of a particular species at 11 different levels, ranging from 50% to 100% with a step of 5%, thus obtaining a dataset of 121 time series in total. To reduce computational complexity, double knock-downs were implemented by simultaneously reducing the concentrations of a pair of any two selected species in Table 1 at 3 different levels, 50%, 75% and 100%, resulting in a total of 495 time series. All simulations were run over 6,000 time units reported at 6,000 time points. The time span represented 1 hour of EGF signalling in-vitro.

We applied our analysis to each time series as described in Section 3. By applying PCA we obtained R=3 at 95% of the total eigenvalue sum, and from the representation of the time series in 3-dimensional

space it was clear that both single-knockdown and double-knockdown datasets present non-spherical shapes and a high degree of noise. As discussed in Section 3, partitioning and hierarchical methods work better if the clusters are spherical, but their performance degrades for non-spherical clusters. In Table 2 we give the results of the cluster validation, in terms of the CDbw index, obtained by comparing DBScan with two classical implementation of K-means and agglomerative hierarchical algorithms, with respect to different configurations (parameter settings). In particular, we considered different values of K (number of clusters) for K-means and hierarchical clustering, and for DBScan we considered values of MinPts ranging from 1 to 5 and values of Eps (radius of the neighbourhood) ranging from 0.25 to 5.

DBScan outperforms both the alternative algorithms, and the difference in performance is even more pronounced for the double-knockdown dataset, because the clusters contain more noise and are less spherical-shaped. The best outcomes of DBScan comprises a set of 4 clusters for single-knockdown and 10 clusters for double-knockdown, represented in Fig.1.

We used the algorithm shown in Sec. 4 to generate PLTLc statements which describe the behaviour of the time series in the clusters. The results in Table 3 represent the quality of these statements in terms of *confidence value* and percentage of time series in cluster C_i which fulfill the statements.

Almost all the statements in the single-knockdown dataset have higher confidence values compared to the double-knockdown dataset, which is due to the high degree of noise in the latter. The only exception is $S1$, which wrongly classifies some time series belonging to C_2 and C_4 as belonging to C_1, because the latter consists of time series with heterogeneous behaviours, some of which are similar to those of C_2 and C_4.

The results for the double-knockdown dataset, despite the noise and the higher number of elements, show that a good number of statements unambiguously describe their cluster ($S3$, $S4$, $S5$, $S7$, $S8$, $S9$, $S10$). We note that the $S6$ and $S8$ do not discriminate well between C_6 and C_8, which is because C_6 and C_8 have similar behaviours. The time series in C_1 are highly heterogeneous, and therefore $S1$ is too general and wrongly classifies as belonging to C_1 some of the time series belonging to C_2,C_9 and C_{10}.

For demonstration purposes, we give one example of PLTLc statements generated by our method, and its biological interpretation:

C_2 *single-knockdown*: "Transient to sustained behaviour — the concentration of ERK_{PP} increases, with a slight perturbation between time 2 to 15, reaching a peak between time 25 to 41 after which it remains constantly high."

$P_{=?}[d[ERK_{PP}] > 0 \;\; U(Time \geq 2 \wedge Time \leq 2 \wedge d[ERK_{PP}] < 0 \;\; U(Time \geq 3 \wedge Time \leq 3 \wedge d[ERK_{PP}] = 0$
$U(Time \geq 12 \wedge Time \leq 15 \wedge d[ERK_{PP}] > 0$
$U(Time \geq 25 \wedge Time \leq 41 \wedge d[ERK_{PP}] = 0 \wedge G(d[ERK_{PP}] = 0))))))]$

The members of this cluster exhibit behaviours which display sustained signalling effects. Because the magnitude and duration (transient versus sustained) of ERK activation decides which type of cellular response is stimulated by MAPK signalling (Cook, Aziz, and McMahon 1999) (Marshall 1995), the knockdowns in this group may provide a potential approach to switch between different cellular processes controlled by the MAPK cascade.

Table 1: List of molecular species chosen to knock down.

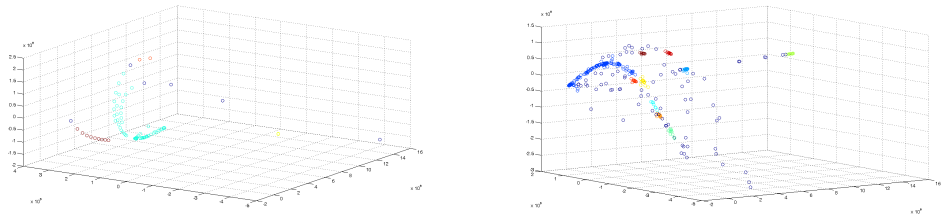
Molecular species			
EGFR	RasGDP	Shc	Phosphatase1
Raf	Phosphatase2	MEK	ERK
SOS	Phosphatase3	Grb2	

Table 2: Results of DBScan, K-Means and Hierarchical clustering algorithms in terms of CDbw, for Single- and Double- Knockdown datasets.

Single-Knockdown							Double-Knockdown						
DBScan			K-Means		Hierarchical		DBScan			K-Means		Hierarchical	
MinPts	Eps	CDbw	K	CDbw	K	CDbw	MinPts	Eps	CDbw	K	CDbw	K	CDbw
1	2.35	0.24	2	0.01	2	0	1	0.95	0.19	5	0.01	5	0.00
2	0.85	0.14	3	0.09	3	0.19	2	0.85	0.23	6	0.01	6	0.00
3	0.35	0.14	4	0.07	4	0	3	0.65	0.34	7	0.02	7	0.00
4	0.35	0.14	5	0.04	5	0	4	0.65	0.34	8	0.00	8	0.03
5	0.45	0.13	6	0.03	6	0.14	5	0.65	0.33	9	0.00	9	0.05
			7	0.06	7	0				10	0.01	10	0.04
			8	0.06	8	0.12				11	0.01	11	0.00

Table 3: Evaluation of the quality of the generated PLTLc statements. The performance is measured in terms of confidence value and percentage of time series recognized.

	Single-Knockdown				Double-Knockdown									
PLTLc Statements	S1	S2	S3	S4	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
$P_{\phi_{Opt}}(C_i)$	1	1	0.82	1	1	1	1	1	1	1	0.73	0.91	1	0.98
$Conf_{\phi_{Opt}}(C_i)$	0.67	0.99	0.82	1	0.61	0.65	1	1	1	0.52	0.73	0.91	1	0.98



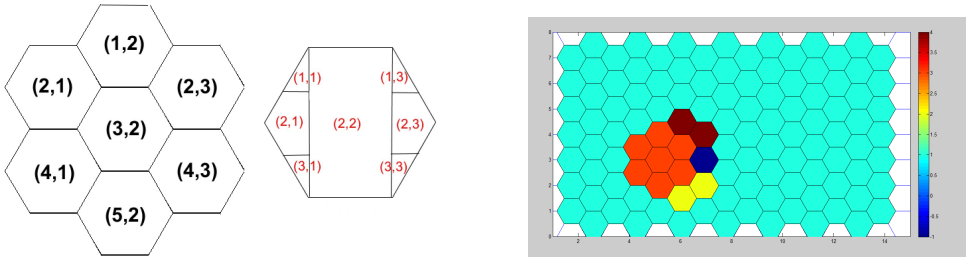
(a) Single-Knockdown dataset (b) Double-Knockdown dataset

Figure 1: DBScan clustering results with the best performance in terms of CDbw index.

5.2 Planar Cell Polarity

In the following, we demonstrate the power of our approach in handling multiscale models by using Planar Cell Polarity (PCP) as an example, where we wish to model the influence of changes at the genetic level on changes in physiology at higher levels.

We used our HCPN model to generate an in-silico tissue (patch) consisting of 112 hexagonal cells. In particular, we modeled the effect on neighbouring wild-type cells of a mutant clone with cells lacking the molecular species Frizzled (Fz) by completely knocking out the concentration and transport of the corresponding places in our model. We produced a mutant clone of Fz- inside our in-silico patch, comprising seven cells. We used the concentration of Flamingo-Frizzled-Disheveled (FFD) as our target species in order to analyse the behaviour of each of the six cell membrane compartments, and all simulations were run over 500 time units reported at 500 time points. The original T variables represent all time points from all six time series (3000 variables in total), and we applied PCA to generate a new set of R variables summarising the information of the multiple time series. In order to identify cells with similar behaviours in all six cell membrane compartments we applied DBScan with different settings of MinPts values (ranging from 1 to 5) and Epsilon values (ranging from 0.05 to 2). The solution, which contains 5 clusters, has been obtained by using the performance measure CDbw as described in Sec. 3.3.



(a) (b)

Figure 2: (a) Left: A patch of cells is structured as an hexagonal grid (black coordinates refer to cell positions), and Right: each cell is subdivided into compartments (red coordinates refer to compartments). (b) The clustering results identify the mutant clone and its effect on neighbouring wild-type cells.

To compare our analytical results with images obtained from bio-laboratory, there was a need to provide a clear visual representation of our clustering results which assigns a single colour to cells within one cluster along the tissue. We used a hexagonal grid to represent the tissue and coloured each cell according to the cluster which it belongs to. Our approach is capable of identifying cells in the mutant clone (in red), as well as neighbouring cells distally adjacent to the clone (see Figure 2). These distally adjacent cells exhibit three types of behaviours according to their positions.

The primary results show that there are compartments sharing some time series with similar behaviour. In order to generate PLTLc descriptions of each compartment we decided to compute a new set of clusters considering each of the six compartment separately and then apply the algorithm given in Sec. 4.

By analysing the results we discovered some important properties:

- Cells in the mutant clone are assigned to one cluster and all compartments exhibit the same behaviour, where the concentration of FFD equals to zero all the time.
- In clusters comprising cells distally adjacent to the mutant clone, differences mainly occur in compartment (1,1) (proximal top compartment) & (3,1) (proximal bottom compartment), see Fig. 2a, right. Compared to wild-type cells, the cells in the yellow cluster differ only for compartment (1,1), the cells in the brown cluster differ only for compartment (3,1), and the cell in the blue singleton cluster differs for both compartment (1,1) and (3,1).

For demonstration purpose we give the PLTLc statement describing one specific compartment:

Non-wild-type behaviour: "The concentration of FFD increases from time zero, reaches its peak between time 30 and 31, and then becomes steady till the end".

$P_{=?}[d[FFD] > 0\ U(Time \geq 30 \wedge Time \leq 31 \wedge d[FFD] = 0\ \wedge G(d[FFD] = 0)))]$

6 CONCLUSION

In this paper we have described a machine learning approach for the analysis and interpretation of the simulation results of a system biology model which combines unsupervised clustering and the automatic generation of semantic descriptions of clusters. The method can be considered as a TL feature generation method to identify time series with similar behaviour and automatically generate logic statements to describe them semantically. The validity of these statements is evaluated using a model checking technique and a confidence index.

This methodology has been successfully validated on two case studies, a uniscale model of a cell and a multiscale model of a tissue composed of many cells. In the first model, our technique allowed us to automatically analyse the behaviour of the cell when subjected to different perturbations, and we showed that our algorithms maintain high performance as the number of time series grows. The second case study demonstrated the applicability of our methodology to a multiscale model, where we need to characterise

the behaviour of many different time series at once. In both cases, the PLTLc statements produced to describe the clusters show high discriminative power.

REFERENCES

Aziz, A., K. Sanwal, V. Singhal, and R. K. Brayton. 1996. *Verifying Continuous-Time Markov Chains*, Volume 1102, 269–276. New Brunswick, NJ, USA: Springer Verlag.

Baier, C. 1998. *On Algorithmic Verification Methods for Probabilistic Systems*. Habilitation thesis, University of Mannheim.

Batal, I., H. Valizadegan, G. F. Cooper, and M. Hauskrecht. 2011. *A Pattern Mining Approach for Classifying Multivariate Temporal Data*, 358–365. IEEE.

Clarke, E., O. Grumberg, and D. Peled. 2001. *Model Checking*. MIT Press 1999, third printing.

Cook, S. J., N. Aziz, and M. McMahon. 1999. "The Repertoire of Fos and Jun Proteins Expressed during the G1 Phase of the Cell Cycle Is Determined by the Duration of Mitogen-Activated Protein Kinase Activation". *Mol. Cell. Biol.* 19:330–341.

Donaldson, R., and D. Gilbert. 2008. *A Model Checking Approach to the Parameter Estimation of Biochemical Pathways*, 269–287. Springer Volume 5307.

Dunn, J. C. 1974. "Well-Separated Clusters and Optimal Fuzzy Partitions". *Journal of Cybernetics* 4:95–104.

Ester, M., H.-P. Kriegel, J. Sander, and X. Xu. 1996. *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*, 226–231. AAAI Press.

Fages, F., and A. Rizk. 2007. *On the Analysis of Numerical Data Time Series in Temporal Logic*, 48–63. LNCS/LNBI 4695, Springer.

Fersini, E., E. Messina, F. Archetti, and C. Manfredotti. 2010. "Combining Gene Expression Profiles and Drug Activity Patterns Analysis: A Relational Clustering Approach". *Journal of Mathematical Modelling and Algorithms* 9 (3): 275–289.

Gao, Q., F. Liu, D. Gilbert, M. Heiner, and D. Tree. 2011, September. *A Multiscale Approach to Modelling Planar Cell Polarity in Drosophila Wing using Hierarchically Coloured Petri nets*, 209–218. ACM digital library.

Gilbert, D., M. Heiner, and S. Lehrack. 2007. *A unifying framework for modelling and analysing biochemical pathways using Petri nets*, 200–216. CMSB'07. Berlin, Heidelberg: Springer-Verlag.

Golay, X., S. Kollias, G. Stoll, D. Meier, A. Valavanis, and P. Boesiger. 1998. "A new correlation-based fuzzy logic clustering algorithm for fMRI". *Magnetic Resonance in Medicine* 40:249–260.

Halkidi, M., and M. Vazirgiannis. 2008. "A density-based cluster validity approach using multi-representatives". *Pattern Recognition Letters* 29:773–786.

Hansson, H., and B. Jonsson. 1994. "A Logic for Reasoning about Time and Reliability". *Formal Aspects of Computing* 6 (5): 512–535.

Heiner, M., M. Herajy, F. Liu, C. Rohr, and M. Schwarick. 2012, June. *Snoopy – a unifying Petri net tool*, Volume 7347 of *LNCS*, 398–407. Springer.

Hornberg, J. J., B. Binder, F. J. Bruggeman, B. Schoeberl, R. Heinrich, and H. V. Westerhoff. 2005. "Control of MAPK signalling: from complexity to what really matters". *Oncogene* 24 (36): 5533–42.

Hoshino, R., Y. Chatani, T. Yamori, T. Tsuruo, H. Oka, O. Yoshida, Y. Shimada, S. Ari-i, H. Wada, J. Fujimoto, and M. Kohno. 1999. "Constitutive activation of the 41-/43-kDa mitogen-activated protein kinase signaling pathway in human tumors". *Oncogen* 18:813–822.

Kholodenko, B., O. Demin, G. Moehren, and J. Hoek. 1999. "Quantification of short-term signaling by the epidermal growth factor receptor". *JBC* 274:30169–30181.

Liao, T. W. 2005. "Clustering of time series data: a survey". *Pattern Recognition* 38 (11): 1857–1874.

Marshall, C. J. 1995. "Specificity of receptor tyrosine kinase signaling: transient versus sustained extracellular signal-regulated kinase activation". *Cell* 80:179–185.

MATLAB 2010. *version 7.10.0 (R2010a)*. Natick, Massachusetts: The MathWorks Inc.

Morris, K. 2008. *RNA and the Regulation of Gene Expression: A Hidden Layer of Complexity*. Caister Academic Press.

Pnueli, A. 1981. "The Temporal Semantics of Concurrent Programs". *Theor. Comput. Sci.* 13:45–60.

Rousseeuw, P. J. 1987. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis". *Journal of Computational and Applied Mathematics* 20:53 – 65.

Schoeberl, B., C. Eichler-Jonsson, E. Gilles, and G. Müller. 2002. "Computational modeling of the dynamics of the MAP kinase cascade activated by surface and internalized EGF receptors". *Nat. Biotech.* 20:370–375.

Simons, M., and M. Mlodzik. 2008. "Planar cell polarity signaling: From fly development to human disease". *Annu. Rev. Genet* 42:517–540.

Strutt, D. I. 2002. "Asymmetric localization of frizzled and the establishment of cell polarity in the drosophila wing". *Mol. Cell* 7:367–375.

Van Wijk, J. J., and E. R. Van Selow. 1999. *Cluster and Calendar Based Visualization of Time Series Data*, 4–. INFOVIS '99. Washington, DC, USA: IEEE Computer Society.

AUTHOR BIOGRAPHIES

DANIELE MACCAGNOLA (daniele.maccagnola@gmail.com) received a bachelor degree in Computer Science and a master degree in Bioinformatics at the University of Milano-Bicocca where, from March 2012, he is a research fellow. His main interests are in data mining, clustering and their application to systems biology.

ENZA MESSINA (messina@disco.unimib.it) is a Professor in Operations Research at the University of Milano-Bicocca (Department of Informatics Systems and Communications), where she founded the research Laboratory MIND (www.mind.disco.unimib.it). She holds a PhD in Computational Mathematics and Operations Research from the University of Milano. Her research activity is mainly focused on the development of models and methods for decision making under uncertainty and statistical relational models for data analysis.

QIAN GAO (qian.gao@brunel.ac.uk, web page http://people.brunel.ac.uk/~cspgqqg) is a PhD student of Systems Biology at School of Information Systems, Computing and Mathematics, Brunel University, UK. Her interests centre around System Biology, Bioinformatics and their applications in biomedical and health sciences. Her current research focuses on the development and application of computational techniques to model and analyse multiscale systems.

DAVID GILBERT (david.gilbert@brunel.ac.uk, web page http://people.brunel.ac.uk/~csstdrg) is a Professor of Computing and Co-Director of the Centre for Systems and Synthetic Biology at Brunel University, UK. He holds a PhD in computational logic from Imperial College. David's research interests lie in the development and application of computational techniques to model and analyse biological data, with a focus on multiscale systems, and approaches to support the design and development of synthetic biological systems.