Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 52-26 (2019) 52–57

Literature and data-driven based inference of signalling interactions using time-course data

Enio Gjerga*, **, Panuwat Trairatphisan**, Attila Gabor*, **, Julio Saez-Rodriguez *, **

*RWTH Aachen University, Faculty of Medicine, Joint Research Centre for Computational Biomedicine (JRC-COMBINE), Aachen, Germany

**Heidelberg University, Faculty of Medicine, and Heidelberg University Hospital, Institute for Computational Biomedicine, Bioquant, Heidelberg, Germany

Corresponding author: Julio Saez-Rodriguez (julio.saez@bioquant.uni-heidelberg.de)

Abstract: Cellular activity and responses to stimuli are governed through an elaborated communication process called cell signalling. The modelling of signalling mechanisms has the potential to help us understand the regulatory processes determining cellular behaviour. One approach to derive models of signalling networks is from data alone. Another one is to use prior knowledge networks (PKN's) derived from literature or experts' knowledge to build models that are trained to data. Both approaches have limitations. Data-driven methods can infer many false-positive interactions. Literature-constrained methods, on the other hand, are limited to model only known interactions. To overcome these limitations, within a logic ordinary differential equations (ODE) formalism, we have developed Dynamic-Feeder. The framework identifies and incorporates new possible links to the network and then it evaluates their effects based on how the models predict the data. Dynamic-Feeder combines data-driven inference methods with general literature-based knowledge of proteins interaction networks (PIN's). We illustrate our method with a published case study using phosphoproteomic data upon perturbation of breast cancer cell lines.

© 2019, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Signaling Networks, Dynamic Models, Pathway Resources, Protein Interactions.

1. INTRODUCTION

Cells rely on a system of signal processing and transmission networks to perform and coordinate their basal activities. The transmission process is governed through chemical signals which can be mediated by proteins or other smaller molecules. At the protein level, the signal propagation relies on post-translational modifications and protein-protein interactions. Considering the variety of proteins and their modifications, signalling networks are generally large, complex and dynamic systems.

One way to deal with this level of complexity is to build computational models that represent a simplification of the signalling system. The model parameters are finely tuned to best describe the observed relationship between the inputs and the outputs(Janes & Lauffenburger 2013). By inputs, we refer to the perturbations we add to the system (i.e. extracellular ligand and inhibitors of specific intracellular proteins) that produces an observed response within the cells (i.e. measured activity or abundance of proteins or other molecules). These models can then be used to predict other perturbation effects with relevant therapeutic implications.

Many computational tools have been developed to model cell signalling networks. Logic-based models, in particular, are a suitable variant due to their efficiency and simplicity for describing biochemical networks without the need to include details about the biochemistry of each interaction. These methods can provide a qualitative (Chaouiya et al. 2012; Müssel et al. 2010; Batt et al. 2012; Dorier et al. 2016) or quantitative (Stoll et al. 2012; De Landtsheer et al. 2017; Di Cara et al. 2007; Trairatphisan et al. 2014) description of the system by modelling a list of interactions as encoded in a prior knowledge network (PKN). A PKN is typically derived from literature, often summarized in dedicated databases. On this line, we have developed CellNOpt (Terfve et al. 2012) which is a method used for building and training predictive logic models of signalling networks to data.

Modellers often rely on their own knowledge or on pathway resources to decide which interactions to be included in the PKN. However, the main problem with methods that use PKNs is that the knowledge we have about the interactions in the system might be incomplete or even wrong. Furthermore, exploring the literature and large databases containing thousands of interactions and manually identifying a few relevant mechanisms is impractical. To address these issues, Eduati et al. (Eduati et al. 2012) developed a method called CNORFeeder that is implemented in the CellNOpt pipeline (Eduati et al. 2012). CNORFeeder identifies interactions from the data and merges them into the PKN to link the model inputs to perturbed nodes in the network. However, this approach is limited in its applications to only the simple logic Boolean formalism and to two time-point (before and after perturbation) datasets. Furthermore, it does not take into consideration how well the original PKN based Boolean model fits the data. In this paper, we present an extension of

2405-8963 © 2019, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2019.12.235

this method developed within the CellNOpt framework, called Dynamic-Feeder. The method overcomes the limitations of CNORFeeder by inferring new possible functional interactions missing in the PKN for dynamic models casted as logic-based Ordinary Differential Equations (ODEs) from time-series data. We show a step-by-step summary of the Dynamic-Feeder pipeline and apply it to a study from the HPN-DREAM Breast Cancer challenge (Hill et al. 2016) where time-resolved phospho-proteomic data from breast cancer cell lines were used to identify causal influences in signalling networks.

2. METHODS

2.1 CellNOpt

CellNOpt is a free R package, which trains a PKN to data, in order to identify functional proteins and interactions in a specific context. The data used to refine the networks comes from low-to-medium throughput targeted proteomic experiment that offers superior levels of quantitative accuracy and reproducibility (Bensimon et al. 2012) compared to large-scale label-free or un-targeted data acquisition techniques. Furthermore, depending on the quality and temporal resolution of the data, CellNOpt features different logic formalisms describing the system with increasing levels of details: from Boolean models for perturbation data to logic-based ordinary differential equation (ODE's) models (MacNamara et al. 2012). In this paper, we will focus on the logic ODE's since the Dynamic-Feeder pipeline has been designed specifically for this kind of formalism.

The logic-ODE formalism has been implemented in CNORode, an add-on package of CellNOpt. CNORode can handle time-course data in a continuous manner. It derives and trains a system of logic ordinary differential equations (ODEs) to account for continuous values of state and time. The set of ODE's for each species *i* in the PKN are defined as in (1).

$$dx_i / dt = \tau_i (B_i(s(x_{1,i}), s(x_{2,i}), \ldots, s(x_{N,i})) - x_i) \quad s \in [0,1] \tag{1}$$

Where x_i represents the activity state of node $i \in \{1,2,\ldots,n_s\}$ (where n_s is the number of species in the network); while B_i represents the continuous activation function of node *i* based on the activity levels of it's *N* regulators and their logic combination as encoded in the PKN (Wittmann et al. 2009). τ_i is the life-time of the species *i* and we can we refer to it as the responsiveness of the species *i*. Higher values of τ_i on this case means that the protein is more responsive to it's upstream activators. The regulation level of a node *i* from one of it's regulators *j* is defined by a sigmoidal transfer function $s(x_{j,i})$ as described in (2).

$$s(x_{j,i}) = 1 - \frac{(1-x_{j,i})^{n_{j,i}} / [(1-x_{j,i})^{n_{j,i}} + k_{j,i}^{n_{j,i}}]}{1/(1+k_{j,i}^{n_{j,i}})} \tag{2}$$

Besides τ, the other continuous Hill parameters *n* and *k* (representing respectively the degree of cooperativity between interactors sharing an edge in the PKN and the strength of interaction) are also optimized for each edge since they represent pathway dynamic. CNORode relies on CVODES library for the simulations, while the continuous parameters are optimized by using MEIGO (Egea et al. 2014) optimization toolbox.

Fitting of model to data is formalized as an optimization problem. Specifically, we aim to identify the set of model parameters that minimizes the mean square error (MSE) between our model predictions and the scaled data for all measured nodes (Saez-Rodriguez et al. 2009). Each fitted parameter represents specific dynamic properties of the nodes and the edges of the PKN (i.e. strength of interactions, response, etc.). The goal is to minimize the sum of model predictions $x \in [0,1]$ and measured scaled values $m \in [0,1]$. To account for model sparsity, an *L1* penalty term can be introduced to the objective function on the parameters. The balance between goodness of fit and model sparsity can be adjusted by a parameter λ (Eduati et al. 2017):

$$\min(\frac{1}{N}\sum_{i=1}^{n_s}\sum_{z=1}^{n_e}(m_{i,z} - x_{i,z})^2 + \sum_{i=1}^{n_s}\lambda^{\tau}\tau_i + \sum_{l=1}^{n_r}\lambda_k k_l + \sum_{l=1}^{n_r}\lambda_n n_l) \tag{3}$$

Where *i* and *l* are the index sets of species (n_s being the total number of nodes in the PKN) and reactions (n_r being the total number of interactions in the PKN). The index $z \in \{1,2,\ldots,n_e\}$ refers to a specific experimental condition in which a set of nodes is being stimulated or inhibited. We can assign a specific penalty parameter value λ for each edge or node in the PKN and control which network components can be more prioritized compared to others (i.e. based on the number of literature mentions). To reduce the computational complexity of our optimization problem and relieve identifiability issues, the PKN can undergo pathway compression to remove all the un-measured and un-perturbed species without impairing the logical consistency of the network. In addition, all possible logic AND gates between pairs of interactors (incoming to each node in the PKN) (Saez-Rodriguez et al. 2009; Terfve et al. 2012).

2.2 OmniPath – Integrating pathway resources

While assembling a PKN, users can rely on many available resources, typically databases of signalling pathways. However, these databases are maintained by different groups and information can be curated differently. This can lead to inconsistency in the direction and sign of an interaction across various resources thus making it unclear for the user which one to use. We use OmniPath (Türei et al. 2016) to identify potential missing links. OmniPath integrates 55 publicly available resources, and it includes information about how many times a specific interaction has been reported in the literature. The number of evidence from literature about a specific interaction can be used as prior information about how we would penalize these interactions.

2.3 Integration of CellNOpt to add missing links – CNORFeeder

CNORFeeder (Eduati et al. 2012) can be used to propose potential missing interactions in the prior knowledge in order to obtain new models with a better fit to the data. However, the method is limited in its application to the simple Boolean formalism. Hence, it cannot capture links, which may describe quantitative differences in the dynamic behavior of the system.

2.4 Dynamic-Feeder

In order to extend the CNORFeeder application to time-series data, we have developed the Dynamic-Feeder method. The way the problem is approached is similar to CNORFeeder, as described in the previous section. The new Dynamic-Feeder pipeline (**Figure 1**) is described next.

2.4.1 Estimation of initial set of parameters

The first step of the Dynamic-Feeder pipeline consists of performing a simple dynamic analysis of the initial PKN with CNORode. We first fit a PKN to observed data and obtain the set of continuous model parameters, which best fits the observations. From here on we refer to these tuned parameters as the initial set of optimal parameters.

2.4.2 Identifying poor fits

In each experimental condition, we are stimulating or inhibiting the activity of specific nodes in the PKN while observing their effects on the measured nodes. After the initial analysis, we identify a list of measurements, which are poorly fitted across each of the experimental conditions. We use the mean squared error (*MSE*) as a metric to identify the bad fits. For each measurement across each experimental condition, we can compute *MSE* across all time points in order to get an insight into the quality of fit for each measurement. In this case, an *MSE* threshold value can be set in order to define when a measurement has not been fitted well. A measurement is considered to be poorly fitted when it's *MSE* value is higher than the specified threshold (see Section 3.2).

2.4.3 Integration with PKN

Poorly fitted nodes indicate missing interactions from the PKN. After identifying the measurements that are poorly fitted, we can then use various resources to identify a set of possible signed links connecting with the shortest paths the measured nodes and the proteins, which have been perturbed on that specific experimental condition. We particularly make use of OmniPath because of the reasons as described in section 2.2. We can constrain the search with a maximal path length parameter set by the user. Also, we can control the sparsity of the integrated network by the threshold parameters (over the MSE values) of what we want to consider as a poorly fitted measurement. The higher the threshold parameter we use, the sparser the network. From the list of all the shortest paths identified, we then only retain those ones which have a higher overlap with the original PKN (common interactions). On the way we identify and integrate potential missing links, lies also the difference of Dynamic-Feeder to CNORFeeder: in CNORFeeder, all possible interactions linking cues (the perturbed components of the network) to the regulated measurements were integrated while on Dynamic-Feeder we are more focused on identifying and adding those links which can potentially correct the misfits. This has a computational advantage, especially when considering the complexity of the dynamic modelling of signalling networks.

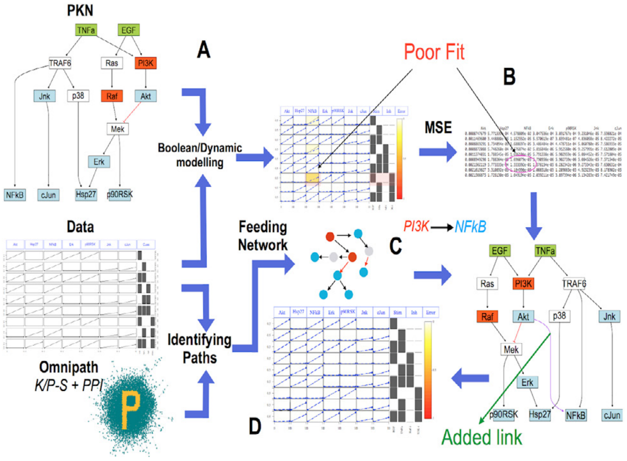

Figure 1.: Dynamic-Feeder pipeline. **A)** Initial training of the PKN to data. **B)** Identifying poorly fitted measurements. **C)** Identifying possible missing links from interaction resources (activation of NFkB by PI3K through Akt). **D)** Integrating missing links to PKN and running final training to data results to better fits.

2.4.4 Training the integrated PKN

Once we have identified a set of possible links as described in the previous section, we integrate it to the PKN and then train the new model. In this case, similar to CNORFeeder, we penalize the addition of the new links to the system by introducing another penalty factor to the objective function for each parameter associated with the new nodes and links. The integrated network is then trained to the data. The new set of parameter values are identified for each link and node in the integrated PKN. The higher penalty factor for the new members in the PKN will discourage the addition of new links, which do not provide a sensible contribution to the improvement of the overall fit.

3. RESULTS

The Dynamic-Feeder method was applied to a case study from the HPN-DREAM Breast Cancer challenge (Hill et al. 2016). The data was scaled in the range between 0 and 1 across all the time-points (5min, 15min, 30min, 60min, 120min, and 240min) with a basal level set to 0.5 in the control measurement at time-point 0. An activity value less than 0.5 means that the measurement is at a lower activity level compared to the basal (0 being the lowest). An activity value of more than 0.5 means that the measurement is at a higher activity level compared to the basal (1 being the highest). The initial structure of the PKN used to train the HPN-DREAM data was generated by combining interactions from Reactome (Fabregat et al. 2018) as explained in (Razzaq et al. 2018). For computational efficiency, the PKN (having originally 120 interactions and 37 nodes, of which 21 are measured) was further reduced to contain only those measurements that map in OmniPath and by excluding the non-identifiable nodes through compression. The resulting network (**Figure 2**) contains 58 interactions connecting 14 measured proteins (of whom 2 are also inhibited), 6 are stimulated while 4 are not measured.

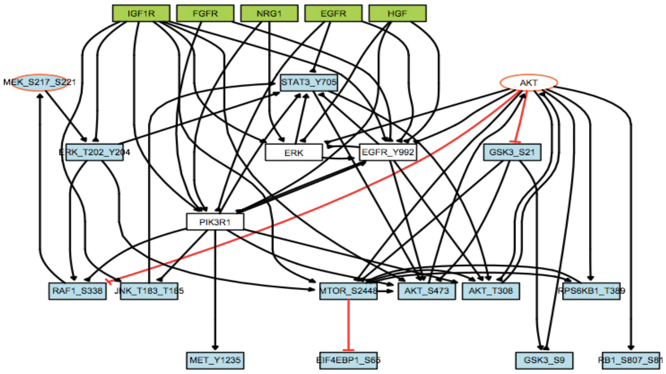

Figure 2: Initial PKN. Stimulated species are represented as green nodes, red nodes are inhibited, blue nodes are measured and white nodes are un-measured and un-perturbed. Black edges represent activatory interactions, while red edges represent inhibitory interactions.

3.1 *Training of the original PKN*

As described in the pipeline, we start by first training the network to data. Even considering the small size of the model, we allow training for 2 hours to ensure that the initial solution we obtain has reached the best possible fit (**Figure 3**). We fix the parameters *n=3* while only optimizing the other two parameters *k* and τ for each edge and node in the PKN. For each of them, we apply an *L1* regularization penalty factor $\lambda_k = 0.01$ and $\lambda_\tau = 0.1$ respectively, to induce sparsity in the model. These specific values of λ correspond to the best accuracy in estimating the parameters as according to a small in-silico case-study explained in (Eduati et al. 2017). The initial training provided a fitting score with a sum of squared residuals *RSS=24.36282*. In order to better visualize the importance of the inferred parameter values, we map the estimated values of k and τ to their corresponding network components as in (**Figure 4**).

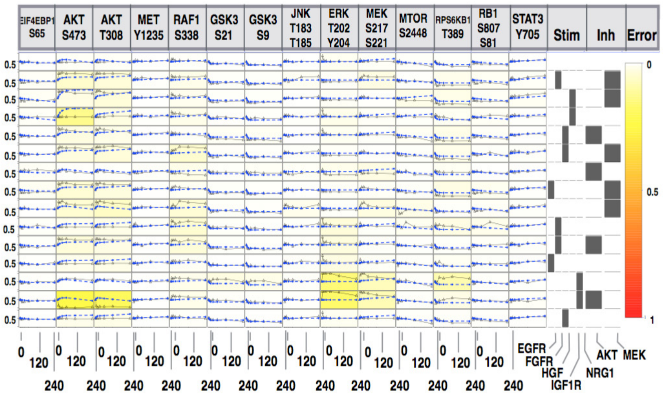

Figure 3: Solution obtained by training PKN to data. In blue dashed lines we have the simulations. In solid black lines, the measured data is shown across each time-point. In the columns, we label the measurements, while in the rows we label the experimental condition where we perturb none, one or a combination of cues.

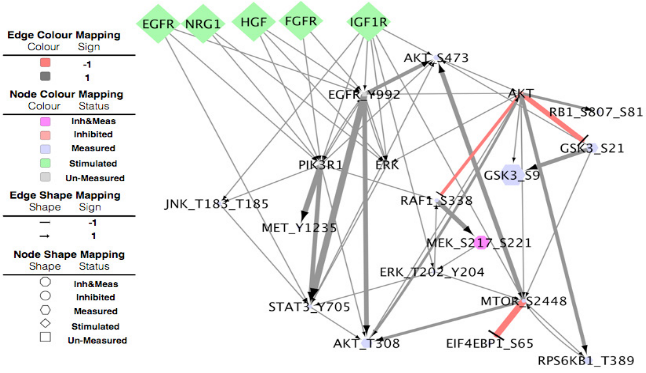

Figure 4: Mapping the initial parameter set as network features. Thicker edges correspond to higher values of *k* (meaning stronger interactions) while the bigger the size of each node; the higher the corresponding inferred parameter (meaning higher responsiveness).

3.2 *Training of the integrated network*

By comparing the simulations to data for the specific set of initial parameters that we have obtained, we are able to identify measurements that were poorly fitted. Based on that we can then identify the list of interactions we can integrate to the PKN. We have screened through all the combination of different parameters: Fitting error or RMSE threshold (the worst *5%*, *10%* and *20%* measurement fits at each condition); Maximal length of paths to search on the database connecting the perturbed cues to the poorly fitted measurements (maximal path length of *2*, *3*, *4* and no path length limit were screened); The multiplier penalty factor over the new links (λ_k multiplied by a factor of *5*, *10*, *50* and *100* for each newly integrated link). *48* different integrated models were then generated and optimized by applying the different parameters mentioned above. The optimized models were then evaluated to identify the best one based on the Akaike's Information Criterion Score (*AIC*) (Akaike 1992). We have computed the Akaike

Information Criterion (AIC) scores for each of the models as:

$$AIC=2P+M*ln(RSS/M) \quad (4)$$

P is the number of functional parameters (the inferred k and τ parameters greater than 0); while M is the number of observations. According to the Akaike model selection criterion, the model with the lower AIC score is considered to be the best one and thus selected as the preferred model.

The Feeder parameters corresponding to the best model in terms of AIC scores were: worst 5% of fits, path length limit of 4 and penalty factor of 5 times for the new integrated links compared to the edges present in the original PKN.

As seen (**Figure 5**), we obtain a model with a better fit than the previous one and with an overall *RSS* score of *RSS=23.24489*. However, the *RSS* score alone might not be a good indicator for telling if indeed our integrated model is better than the original one. For a better estimator of the relative quality of each of the models, we again rely on the AIC score to tell if the new integrated model is better than the original one obtained by training the PKN to data. AIC scores for both of the models yield AIC_{PKN}*=-5972.012* and AIC_{Feed}*=-6032.017,* respectively. Since $AIC_{Feed}<AIC_{PKN}$, we consider the model we obtain from the integrated network as the best one. We map the new estimated parameter values as in (**Figure 6**). The best Feeder network contains *21* new interactions and *4* new species.

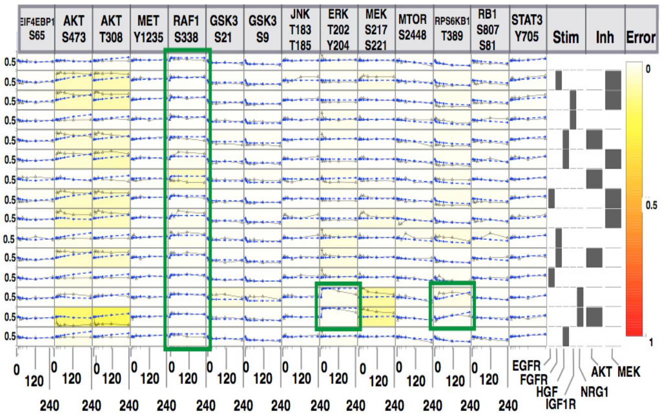

Figure 5: Solution obtained by training integrated network to data. In the green box, we highlight how the simulations fits better the observed dynamic behaviour of *RPS6KB1_T389* and *ERK_T202_Y204* at the experimental conditions in which *NRG1* is being stimulated. There is also an improvement of RAF1_S338 measurement across all the experimental condition compared to the fit of the original PKN.

In particular (as highlighted in the green box in **Figure 5**), we observe that the main improvement in the fitting cost has resulted by adding the NRG1 to RPS6KB1_T389 and ERK_T202_Y204 activatory interactions to the PKN, which reflects the activation of the RPS6KB1_T389 ERK_T202_Y204 in the presence of neuregulin (NRG1).

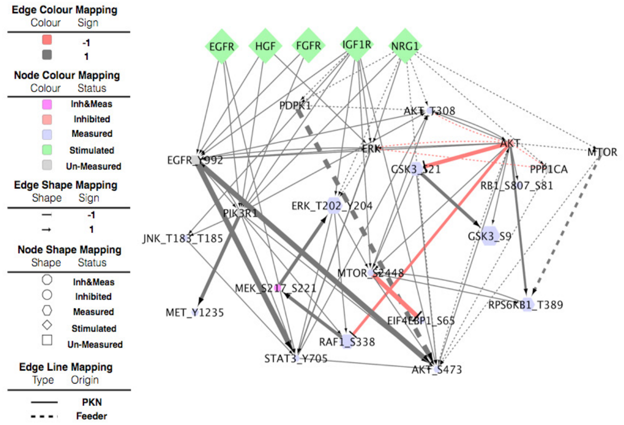

Figure 6: Mapping parameter set as network features in the integrated network. Dashed edges correspond to the newly integrated links.

4. DISCUSSION

In this paper, we present an approach that identifies new possible interactions coming from signalling pathway resources that improve the fit of logic ODE models to the data with respect to the original PKN. This method deals with the incompleteness of the PKN by identifying and integrating new links into the model that help to improve the fit to the data. This is achieved by combining information from perturbation analysis and other resources of protein interactions. We have used OmniPath as a comprehensive collection of pathway resources. From OmniPath, we then identify the shortest path connecting each pair of perturbed network components to poorly fitted measurements. Each new link is then weighted with a higher penalty factor than the ones from the PKN in order to remove the false positives and avoid the over-fits. This approach extends previous efforts by allowing for efficient identification of missing signalling pathways in a context when we do dynamic modelling of the data.

We have applied the method to a real-case published dataset, used in the HPN-DREAM Breast Cancer challenge. We were able to identify a possible activatory connection between NRG1 and RPS6KB1_T389 and ERK_T202_Y204 from this data. Further experimental validations would be necessary to confirm this finding.

Our method has several limitations. One limitation of the method is that it needs PPI (protein-protein interaction) resources from where to find potential missing pathways to integrate into the PKN and we cannot infer any interaction which is yet unknown. Future progress in that direction would be to integrate the Dynamic-Feeder pipeline with some purely data-driven methods of reverse engineering of signalling pathways like SELDOM (Henriques et al. 2017). Another limitation is that Dynamic-Feeder currently searches for the shortest paths in the database that connect the cues with the poorly fitted measurements, and then selects those with the highest overlap with the initial PKN. However, there could be alternative paths between other nodes downstream the cues towards the poorly fitted measurements that could improve the fits. This was done in the original Feeder (Eduati et al. 2012) and it could be implemented in our Dynamic-Feeder pipeline. Addressing this issue is also a scope of our future work regarding the Dynamic-Feeder pipeline.

5. FUNDING & ACKNOWLWDGEMENTS

EG was supported by the European Union's Horizon 2020 research and innovation program (675585 Marie-Curie ITN ''SymBioSys'') and is a Marie-Curie Early Stage Researcher. PT received funding from the TransQST project, an Innovative Medicines Initiative 2 Joint Undertaking under grant agreement No 116030. This Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation programme and EFPIA

6. REFERENCES

Akaike, H., 1992. Information Theory and an Extension of the Maximum Likelihood Principle. *Springer Series in Statistics*, pp.610–624. Available at: http://dx.doi.org/10.1007/978-1-4612-0919-5_38.

Batt, G. et al., 2012. Genetic network analyzer: a tool for the qualitative modeling and simulation of bacterial regulatory networks. *Methods in molecular biology* , 804, pp.439–462.

Bensimon, A., Heck, A.J.R. & Aebersold, R., 2012. Mass spectrometry-based proteomics and network biology. *Annual review of biochemistry*, 81, pp.379–405.

Chaouiya, C., Naldi, A. & Thieffry, D., 2012. Logical modelling of gene regulatory networks with GINsim. *Methods in molecular biology* , 804, pp.463–479.

De Landtsheer, S. et al., 2017. FALCON: a toolbox for the fast contextualization of logical networks. *Bioinformatics* , 33(21), pp.3431–3436.

Di Cara, A. et al., 2007. Dynamic simulation of regulatory networks using SQUAD. *BMC bioinformatics*, 8(1), p.462.

Dorier, J. et al., 2016. Boolean regulatory network reconstruction using literature based knowledge with a genetic algorithm optimization method. *BMC bioinformatics*, 17(1), p.410.

Eduati, F. et al., 2017. Drug Resistance Mechanisms in Colorectal Cancer Dissected with Cell Type-Specific Dynamic Logic Models. *Cancer research*, 77(12), pp.3364–3375.

Eduati, F. et al., 2012. Integrating literature-constrained and data-driven inference of signalling networks. *Bioinformatics* , 28(18), pp.2311–2317.

Egea, J.A. et al., 2014. MEIGO: an open-source software suite based on metaheuristics for global optimization in systems biology and bioinformatics. *BMC bioinformatics*, 15, p.136.

Fabregat, A. et al., 2018. The Reactome Pathway Knowledgebase. *Nucleic acids research*, 46(D1), pp.D649–D655.

Henriques, D. et al., 2017. Data-driven reverse engineering of signaling pathways using ensembles of dynamic models. *PLoS computational biology*, 13(2), p.e1005379.

Hill, S.M. et al., 2016. Inferring causal molecular networks: empirical assessment through a community-based effort. *Nature methods*, 13(4), pp.310–318.

Janes, K.A. & Lauffenburger, D.A., 2013. Models of signalling networks - what cell biologists can gain from them and give to them. *Journal of cell science*, 126(Pt 9), pp.1913–1921.

MacNamara, A. et al., 2012. State-time spectrum of signal transduction logic models. *Physical biology*, 9(4), p.045003.

Müssel, C., Hopfensitz, M. & Kestler, H.A., 2010. BoolNet--an R package for generation, reconstruction and analysis of Boolean networks. *Bioinformatics* , 26(10), pp.1378–1380.

Razzaq, M. et al., 2018. Computational discovery of dynamic cell line specific Boolean networks from multiplex time-course data. *PLoS computational biology*, 14(10), p.e1006538.

Saez-Rodriguez, J. et al., 2009. Discrete logic modelling as a means to link protein signalling networks with functional analysis of mammalian signal transduction. *Molecular systems biology*, 5, p.331.

Stoll, G. et al., 2012. Continuous time Boolean modeling for biological signaling: application of Gillespie algorithm. *BMC systems biology*, 6, p.116.

Terfve, C. et al., 2012. CellNOptR: a flexible toolkit to train protein signaling networks to data using multiple logic formalisms. *BMC systems biology*, 6, p.133.

Trairatphisan, P. et al., 2014. optPBN: an optimisation toolbox for probabilistic Boolean networks. *PloS one*, 9(7), p.e98001.

Türei, D., Korcsmáros, T. & Saez-Rodriguez, J., 2016. OmniPath: guidelines and gateway for literature-curated signaling pathway resources. *Nature methods*, 13(12), pp.966–967.

Wittmann, D.M. et al., 2009. Transforming Boolean models to continuous models: methodology and application to T-cell receptor signaling. *BMC systems biology*, 3, p.98.