

Petri Nets with Fuzzy Logic (PNFL): Reverse Engineering and Parametrization

Robert Küffner*, Tobias Petri, Lukas Windhager, Ralf Zimmer

Institut für Informatik, Ludwig-Maximilians-Universität, München, Germany

Abstract

Background: The recent DREAM4 blind assessment provided a particularly realistic and challenging setting for network reverse engineering methods. The *in silico* part of DREAM4 solicited the inference of cycle-rich gene regulatory networks from heterogeneous, noisy expression data including time courses as well as knockout, knockdown and multifactorial perturbations.

Methodology and Principal Findings: We inferred and parametrized simulation models based on Petri Nets with Fuzzy Logic (PNFL). This completely automated approach correctly reconstructed networks with cycles as well as oscillating network motifs. PNFL was evaluated as the best performer on DREAM4 *in silico* networks of size 10 with an area under the precision-recall curve (AUPR) of 81%. Besides topology, we inferred a range of additional mechanistic details with good reliability, e.g. distinguishing activation from inhibition as well as dependent from independent regulation. Our models also performed well on new experimental conditions such as double knockout mutations that were not included in the provided datasets.

Conclusions: The inference of biological networks substantially benefits from methods that are expressive enough to deal with diverse datasets in a unified way. At the same time, overly complex approaches could generate multiple different models that explain the data equally well. PNFL appears to strike the balance between expressive power and complexity. This also applies to the intuitive representation of PNFL models combining a straightforward graphical notation with colloquial fuzzy parameters.

Citation: Küffner R, Petri T, Windhager L, Zimmer R (2010) Petri Nets with Fuzzy Logic (PNFL): Reverse Engineering and Parametrization. PLoS ONE 5(9): e12807. doi:10.1371/journal.pone.0012807

Editor: Mark Isalan, Center for Genomic Regulation, Spain

Received: March 31, 2010; **Accepted:** June 18, 2010; **Published:** September 20, 2010

Copyright: © 2010 Küffner et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Funding: This work was supported by Helmholtz Alliance on Systems Biology, Project CoReNe (http://www.helmholtz.de/en/joint_initiative_for_innovation_and_research/initiating_and_networking/helmholtz_alliances/systembiologie/helmholtz_alliance_on_systems_biology/networks_in_detail/corene_control_of_regulatory_networks_with_focus_on_non_coding_rna/). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: Robert.Kueffner@bio.ifi.lmu.de

Introduction

The inference of biological networks based on gene expression measurements is a complex task. A range of approaches have been developed for that purpose, which is in turn reflected by a range of corresponding reviews [1–7]. Basic principles to derive relationships between genes or proteins include ordinary differential equations (ODE) [8–10], mutual information [11] and Bayesian networks [12].

Predictions from the available methods are currently quite unreliable as shown in several comparative studies on *in silico* networks [13–16]. For instance, precisions of less than 30% have been observed in [14] for all approaches investigated. This might be due to the fact that most methods were developed to exploit either (static) interventional datasets such as knockout experiments or dynamic datasets such as time courses, but not both [4].

Whether the incorporation of a broad range of datasets can increase the reliability of network reconstruction is explored by the DREAM competitions that conduct blind assessments of network reverse-engineering approaches [17]. The *in silico* part of DREAM4 (2009) provided time course datasets together with

complex knockout, knockdown and multifactorial perturbation datasets.

We present a network inference approach based on Petri Nets with Fuzzy Logic (PNFL) [18]. Similar to ODEs but in contrast to Bayesian or mutual information networks, PNFL enables a simulation of the models. In contrast to the more detailed ODEs, PNFL employs a simpler rule based discrete modeling system. The simulation is important for the investigation and refinement of mechanistic network models in order to capture the dynamic behavior of systems in addition to their topology. In case of DREAM4, we simulate to re-generate the provided datasets. The objective of our inference approach is the reconstruction of models by optimizing the agreement between all of the datasets provided in the challenge and those generated by PNFL. Heterogeneous datasets can thus be exploited and scored in a unified way.

In the following, we briefly summarize the DREAM4 setting, introduce Petri Nets and PNFL and outline our approach to simulate and reconstruct PNFL models. Subsequently, we describe the results we obtained in the DREAM4 *in silico* size ten challenge.

Methods

Setting of the DREAM4 *in silico* size ten challenge

Problem statement. The *in silico* part of DREAM4 aims at the reconstruction of gene regulatory networks where effects are propagated via directed transcription factor (TF, i.e. the effector protein) \rightarrow target gene relationships. TFs are synthesized from their corresponding genes and can thus be themselves the targets of other TFs. Other kinds of relationships (e.g. alternative splicing, protein modification, transport, metabolic reactions) were not considered.

The task is the automated reverse engineering of the directed topology of five different networks with ten nodes per network. The topology to be predicted merges genes and their products (i.e. the TFs) into single nodes. All networks contain cycles, but no self loops. No direct information on the edges is given. Instead, networks are to be inferred from the provided gene expression datasets (see below) alone. In a bonus round, participants used their reconstructed networks to simulate dual knockout perturbations. The problem statement, evaluation and datasets are described in more detail on the DREAM website (<http://wiki.c2b2.columbia.edu/dream/index.php/D4c2>).

Evaluation. After the challenge, submissions were evaluated against the true topology based on the area under the precision-recall curve (AUPR) and the area under the receiver-operator characteristics curve (AUROC). We will focus our discussion on the AUPR. Roughly speaking, an AUPR of 50% means that for each correctly predicted edge an erroneous edge is predicted as well. The sign of the edges (activation vs. inhibition) is not considered in the DREAM4 evaluation. Dual knockout predictions were compared against the true equilibrium values via the mean squared error (MSE). The evaluation is described in more detail in [19].

Gene expression datasets. The approach for dataset generation was developed by Marbach et al. [20,17]. Five time course (TC) datasets were provided. At the beginning of the TC, strong perturbations were applied to the basal transcription levels of about a third of all genes. Halfway through the TC the perturbation was removed so that the network relaxed to the wild type (WT) equilibrium state (5 TC * 20 measurements * 10 genes = 1000 values). All other datasets contained equilibrium gene levels only. Ten single gene knockout (KO), knockdown (KD) and multifactorial (MF) perturbations (3 * 10 perturbations * 10 genes = 300 values) were provided. Compared to the wild type (WT), basal transcription levels of KO and KD target genes were reduced to 0% and 50%, respectively. MF datasets were generated by applying moderate perturbations to the basal activation levels of all genes in the network. Thus, MF datasets could be regarded as transcriptional variations between different individuals. Given gene levels were scaled to be in the range [0, 1].

Petri Nets

The application of Petri net theory for modeling and analysis of biological networks is well established in the field of systems biology [21–24]. Petri nets are graph representations of networks consisting of two types of nodes: *places*, representing entities like proteins, genes, metabolites etc, and *transitions*, representing reactions or, in general, state changes of entities. The state of an entity is defined by the *tokens* that represent the *marking* of the according place and the overall system state by the marking of the Petri net. Directed edges (*arcs*) connect places to transitions (input arcs) or transitions to places (output arcs). These arcs not only depict which entities influence reactions or are influenced by them, but they also exactly define the effects of a reaction, e.g. by

specifying the amount of substrate consumed and the amount of product produced during a reaction (the *firing* of a transition). For a detailed description of classical Petri nets see [25]. In addition, there exists a wide variety of extensions of Petri nets [26]. A Petri Net with Fuzzy Logic (PNFL) can be defined as an instance of a hybrid functional Petri net (HFPN) [27].

In PNFL models of gene regulatory networks, the activity of a target gene t is controlled by a single transition that discharges into a single output place (see Fig. 1). The marking of this place represents t 's numerical gene level l_t . The relationship between an effector place and a target place is called an *effect*. It is mediated by an *effect arc* connecting an effector-gene place to the transition that controls the target place.

Transitions are always enabled as each place always contains a valid (real-valued) token. Firing a transition removes the old marking on the target place via a target place-transition arc (Fig. 1). After quantifying the effect strength based on the effector gene level (via function c , see eq. 1–3), a new marking is assigned to the target place by the output function o (eq. 4–6). The marking on effector places remains unchanged (test arcs).

A transition, its output place and their connecting arcs can be replaced by hexagonal nodes (Fig. 1) to simplify the representation. In this reduced form, only hexagons and their connecting effect arcs remain as all transitions and places are replaced. Effect arcs will be attached to or detached from transitions during the network reconstruction process (section Reconstruction) thus connecting different hexagonal nodes in the reduced Petri net representation.

Modeling of gene regulatory relationships with PNFL

The evaluation of effects using fuzzy logic involves a three-step procedure that consists of fuzzification, the application of effector rules and defuzzification.

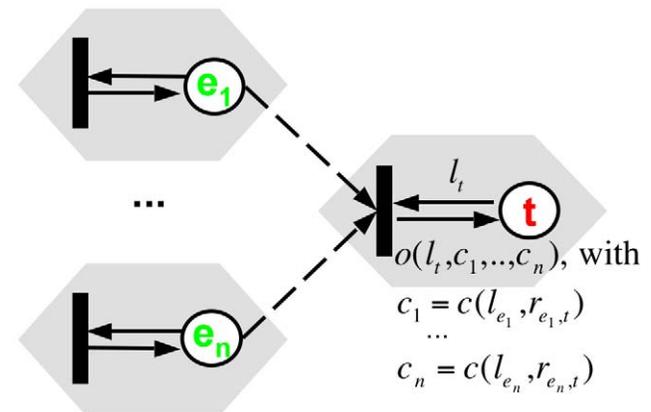


Figure 1. Petri Nets with Fuzzy Logic (PNFL). In Petri nets, states such as effector (e) or target (t) gene levels are represented by places and are depicted as circles. State changes are represented by transitions and are depicted as boxes. Effect arcs (i.e. effector place-transition arcs) define the effectors influencing a target gene via the transition. Firing transitions leaves the marking of the effector places unchanged (test arcs, dashed). After the application of rule tables $r_{e,t}$ to effector gene levels l_e (function c , eq. 1–3), the target gene levels l_t are updated by the output function o (eq. 4–6). In Fig. 6, Fig. 8 and Fig. 10, we represent a transition and its output place as a simplified hexagonal node. The reconstruction determines the topology (=effect arcs) and the parametrization (=rule tables and combination operators) of PNFL models.

doi:10.1371/journal.pone.0012807.g001

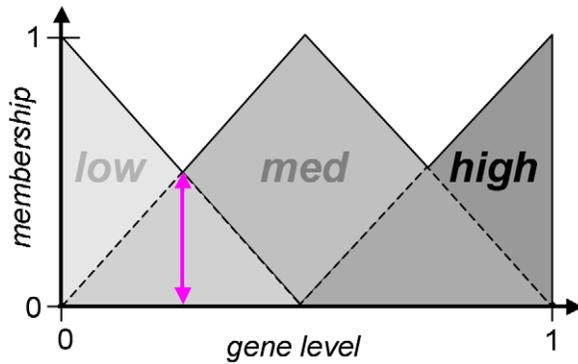


Figure 2. Fuzzification and defuzzification. We use triangular membership functions to fuzzify the continuous gene levels of an effector e into fuzzy sets. As shown by the magenta arrow, a continuous gene level of $l_e = 0.25$ is fuzzified into the fuzzy value $\langle L(low, l_e) = 0.5, L(med, l_e) = 0.5, L(high, l_e) = 0.0 \rangle$. This can be reversed by defuzzification without loss of information.
doi:10.1371/journal.pone.0012807.g002

Fuzzification. In a first step, the continuous gene level $l_e \in [0, 1]$ of an effector e is transformed (fuzzified) into the fuzzy value $\langle L(low, l_e), L(med, l_e), L(high, l_e) \rangle$ by triangular membership functions (eq. 1, Fig. 2):

$$L(low, l_e) = \begin{cases} 0, & \text{if } (l_e > 0.5) \\ 1 - 2 \cdot l_e, & \text{otherwise} \end{cases}$$

$$L(high, l_e) = \begin{cases} 2 \cdot l_e - 1, & \text{if } (l_e > 0.5) \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$L(med, l_e) = 1 - L(low, l_e) - L(high, l_e)$$

Such membership functions are called *fuzzy sets* [28]. Contrary to a classical set, where an object is either contained in the set or not (two-valued logic, $\{0, 1\}$), a fuzzy set assigns a degree of membership from the interval $[0, 1]$ to each object. Thus, the fuzzy value resulting from the fuzzification of a gene level l_e with respect to three fuzzy sets $S \in \{low, med, high\}$ can be interpreted as a fuzzy discretization.

Application of effector rules. Based on the discretization defined by fuzzy sets, the properties of regulatory relationships are modeled by rule tables $r_{e,t}$ (Fig. 3) in analogy to Boolean network models. Rule tables (as used in DREAM4) define three levels of effect strength for both activation (+++, ++, +) and inhibition (---, --, -). A rule table $r_{e,t}: S \rightarrow S$ maps each effector set $E \in S$ to a corresponding target set $T \in S$. The application of a rule by the

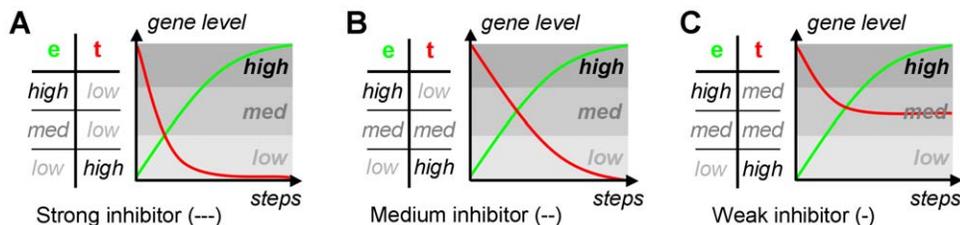


Figure 3. Rule tables. Given fuzzy effector gene levels, we describe the behavior of the targets by rule tables. Rule tables define sign and strength of effects. Fully active strong (---, A) or medium (--, B) inhibitors result in low target activity, which is in contrast to weak inhibitors (-, C). The corresponding strong (+++), medium (++) and weak (+) activator rule tables are constructed by exchanging *high* by *low* and *low* by *high* in the target column.
doi:10.1371/journal.pone.0012807.g003

sum-product logic [29] results in a fuzzy rule consequent C :

$$C(T, l_e, r_{e,t}) = \sum_{E \in S} \begin{cases} L(E, l_e), & \text{if } (r_{e,t}(E) = T) \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Applying eq. 2 to all sets $T \in S$ results in a fuzzy value $\langle C(low, l_e, r_{e,t}), C(med, l_e, r_{e,t}), C(high, l_e, r_{e,t}) \rangle$ describing e 's effect on the target gene t , i.e. it is a fuzzy discretization of the proposed effect.

Defuzzification. By center of gravity defuzzification we obtain a continuous rule consequent c :

$$c(l_e, r_{e,t}) = \frac{0 \cdot C(low, l_e, r_{e,t}) + 0.5 \cdot C(med, l_e, r_{e,t}) + 1 \cdot C(high, l_e, r_{e,t})}{C(low, l_e, r_{e,t}) + C(med, l_e, r_{e,t}) + C(high, l_e, r_{e,t})} \quad (3)$$

with the centers of gravity at 0, 0.5 and 1. Note that due to our choice of fuzzy sets and rule tables the value of the denominator always equals to one. An example calculation involving eq. 1–3 is shown in Fig. 4.

Combination of effects. If several effectors regulate a target gene, their combined effect on the target can be modeled by logical operations [30]. We model two kinds of dependent regulation by the minimum of the effects (*AND* operator) or the maximum of the effects (*OR* operator). The average (*MEAN*) models the independent regulation of a target by its effectors. Dependent and independent regulation are described in Fig. 5. The combination logic currently used in PNFL allows only a single operator (either *AND*, *OR* or *MEAN*) to be selected per target gene regardless of the number of effectors, see eq. 4 and eq. 6.

PNFL simulation

Before simulation, gene levels are initialized to their wild type levels as provided by DREAM4. Let t be a gene targeted by n effectors e_1, \dots, e_n . In each simulation step, updates u of the levels of all genes are computed from the continuous rule consequents $c_j = c(l_e, r_{e,t})$:

$$u(c_1, \dots, c_n) = \begin{cases} 1, & \text{if } (n = 0) \\ c_1, & \text{if } (n = 1) \\ op(c_1, \dots, c_n), & \text{otherwise} \end{cases} \quad (4)$$

with $op \in \{AND, OR, MEAN\}$. Subsequently, u is applied to the gene levels l_t of all target genes via the output function o (eq. 5) at once.

$$l_t = o(l_t, c_1, \dots, c_n) = \begin{cases} 0, & \text{if } (\beta_t = 0) \\ \alpha \cdot \beta_t \cdot u(c_1, \dots, c_n) + (1 - \alpha) \cdot l_t, & \text{otherwise} \end{cases} \quad (5)$$

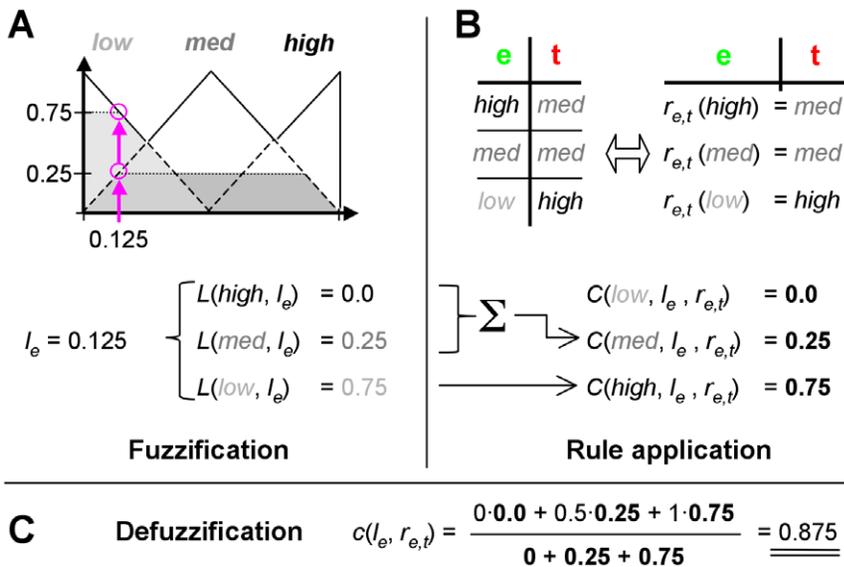


Figure 4. Fuzzy effect calculation example. In this example, the gene level of effector e is $l_e = 0.125$. It is transformed (fuzzified, panel A) into the fuzzy gene level L by application of eq. 1. In panel B, the rule table $r_{e,t}$ (Fig. 3C) is applied to describe the influence of e onto its target gene t by the rule consequent C . C is derived by eq. 2, yielding the fuzzy value $\langle 0, 0.25, 0.75 \rangle$ (panel B). The real valued influence of e onto t , $c(l_e, r_{e,t}) = 0.875$, is calculated by defuzzification (panel C). Such a calculation is performed for all effectors of the target gene t individually. The influences are combined by eq. 4 or eq. 6 (not shown here, see text). doi:10.1371/journal.pone.0012807.g004

The scaling parameter α (Table 1) aligns the PNFL generated time courses to the provided time courses. The transcription rate parameter β_t tunes the transcription rate of gene t , with $\beta_t = 1$ for the wild type transcription rate.

Knockout, knockdown and double knockout data. Gene perturbations are simulated by reducing the transcription rate β_t . In case of a knockout or knockdown simulation, β_t of the perturbed gene t is set to 0 or 0.5, respectively. Similarly, double knockout simulations can be performed.

Time course data. Time course datasets were provided by DREAM4 to show the impact of strong gene perturbations (about a third of all genes) on a network as well as the relaxation to the wild type equilibrium state after removing the perturbations.

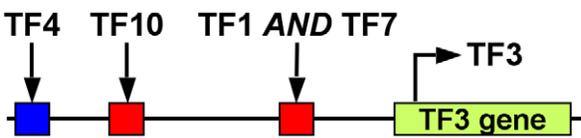


Figure 5. Combinatorial gene regulation. The regulatory logic of different transcription factors (TFs) regulating a target gene used in DREAM4 was disclosed after the challenge. TFs are assumed to bind to cis-regulatory modules (CRMs) to regulate the expression of target genes. Individual CRMs act as enhancers (red) or repressors (blue) of gene regulation. The bound states of different CRMs (e.g. by TFs 4 and 10) are mutually independent. A complex of TFs regulating a given CRM can be represented as AND operator. TFs 1 and 7 are mutually dependent to form the complex and regulate the gene. In turn, a complex of TFs controlling a repressing CRM can be implemented by the OR operator (not shown). The effects of several CRMs on the activity of the target gene are averaged (MEAN operator). In contrast to the arbitrary combination of operators in the DREAM4 approach, PNFL selects only a single operator (AND, OR or MEAN) per target gene (see Methods and Results). The depicted regulation of gene 3 was taken from network 5 (see Fig. 10A). doi:10.1371/journal.pone.0012807.g005

A perturbation is represented as an additional (hidden, i.e. unobserved) node in the network. During reconstruction, we infer perturbation targets together with effector targets, as both were not disclosed in the challenge. Initially, we use eq. 6 instead of eq. 4 for all genes t directly affected by the perturbation. For a time course i , eq. 6 includes the perturbation term $c(l_{p_i}, r_{p_i,t})$ with $l_{p_i} \equiv 1$, where p_i is an additional perturbation effector with corresponding rules $r_{p_i,t}$. The perturbation is disabled halfway through the time course via switching back to eq. 4 thereby allowing the network to return to its wild type state. The reconstructed networks thus consist of 15 variables: 10 genes and 5 perturbation variables for the 5 different time courses.

$$u_i(c_1, \dots, c_n, c(l_{p_i}, r_{p_i,t})) = \begin{cases} c(l_{p_i}, r_{p_i,t}), & \text{if } (n=0) \\ op(c_1, \dots, c_n, c(l_{p_i}, r_{p_i,t})), & \text{otherwise} \end{cases} \quad (6)$$

Multifactorial data. DREAM4 also provided equilibrium values for multifactorial (MF) perturbations. Here, the basal transcription levels of all genes in the network were perturbed, but to a lesser degree compared to the time course perturbations. In contrast to the time courses, we do not compute additional rule consequents for the MF data. Instead, we test how well MF target gene levels can be generated if the PNFL wild type rules are applied to the provided MF effector gene levels. The MF target gene levels are thus approximated, as the basal activation changes are not reflected in the PNFL rules.

Differences between the PNFL and DREAM MF gene levels can be due to three reasons: (1) the inferred effects or their parametrization are inadequate: this should be corrected by the reconstruction method, (2) noise and (3) the MF changes to the basal transcription levels. Reasons (1) and (2) apply equally to all of the datasets. For reason (3) we did not account for, so deviations will be somewhat larger than for the other datasets. Therefore, we use lower weights for MF data in the objective function (Table 1).

Table 1. Parameters used for PNFL based reconstruction.

Parameter descriptions	Equation	Values/Lists of values ^a
Rule tables r : effect strength	2	(M), (W, S), (W, M, S) ^b
Combination operators	4, 6	(MEAN), (OR, AND), (OR, AND, MEAN)
Update ratio α	5	0.4, 0.5, 0.6
Regularization parameter reg	7	0.005, 0.002, 0.001, 0.0005, 0.0002
Weight time course w_{TC}	7	1
Weight knockout w_{KO}	7	8
Weight knockdown w_{KD}	7	6
Weight multifactorial w_{MF}	7	4
Simulated annealing parameter k	8	0.02

^aParameters from *lists* are randomly selected for ensemble predictions (see Submission).

^bDegrees of effect strength, W=weak=(+,-), M=medium=(+,-,-) and S=strong=(+,-,-,-).

doi:10.1371/journal.pone.0012807.t001

Reconstruction

Overview. We construct PNFL models by inferring and parametrizing relationships between genes via appropriate rule tables. Starting from a randomly initialized PNFL model our reconstruction approach (Fig. 6) proceeds via four steps: (1) The topology and parametrization of the initial network are modified by the application of moves. (2) After each move, data is simulated by PNFL and (3) compared to the original data via an objective function. Finally, (4) we use a simulated annealing protocol to decide if a given move should be accepted or rejected. The network optimization thus targets at the best possible

agreement between the DREAM4 provided and PNFL simulated datasets.

Note that the networks discussed here always include 10 genes and the PNFL models always contain one place and one transition for each gene. Topological changes of the PNFL models only involve attachment or detachment of input arcs.

Move set and move probabilities. Starting from a population of randomly initialized networks, the reconstruction proceeds one network modifying move at a time. Each move modifies a single target gene. After a move, data is generated by PNFL and compared against the DREAM data (Fig. 6). We implemented moves on individual networks that add or remove effects (i.e. effect arcs), switch the effect combination logic $op \in \{AND, OR, MEAN\}$ and increase or decrease the effector strength via selecting the corresponding rule tables $r \in \{+++, ++, +, -, --, ---\}$. Each network in the population evolves both independently by the moves mentioned before but also by a set of crossover moves. The crossover moves copy effect strength, combination logic or effects between two individuals.

During reconstruction, particular moves are selected from the move set with a move probability that is proportional to the past move acceptance probability for that move.

Objective function. The quality of the reconstructed networks is evaluated by an objective function *dist*. It is based on the Pearson correlation coefficients ρ_t of the target genes t and a regularization term. Lower values of *dist* indicate a better agreement between the DREAM dataset vectors \mathbf{x}_t and the PNFL dataset vectors \mathbf{y}_t and thus better PNFL models. The vectors \mathbf{x}_t and \mathbf{y}_t are formed by the concatenation of all four kinds of datasets (10 knockout, 10 knockdown, 100 time course, 10 multifactorial values per gene). An additional vector $\mathbf{w} = (w_{KO}, \dots, w_{KO}, w_{KD}, \dots, w_{KD}, w_{TC}, \dots, w_{TC}, w_{MF}, \dots, w_{MF})$ weights the data points with dataset specific weights (Table 1). All three vectors

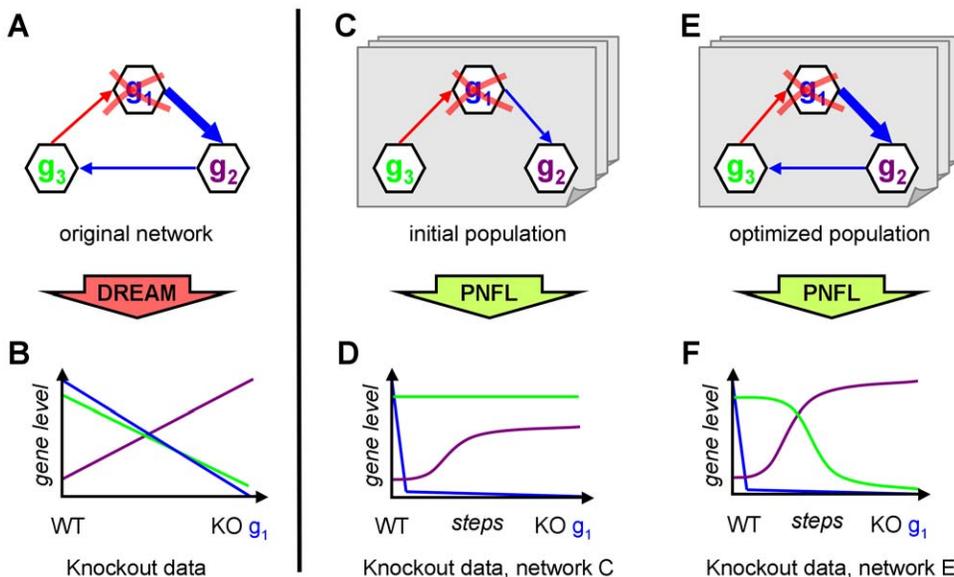


Figure 6. Overview network reconstruction. To reconstruct the original network (A) we mimic the DREAM4 data generation process (A→B). The knockout (KO) of gene 1 is depicted as an example data set in the lower panels. Our reconstruction starts from a randomly initialized population (C) and proceeds through network changing moves. After each move, data is generated by PNFL (D) and compared against the DREAM data (B). We implemented moves on single networks in the population and crossover moves that copy features between pairs of networks. Thereby, favourable features are propagated throughout the population, which eventually leads to improved networks (E) and corresponding datasets (F). Note that - in contrast to the PNFL simulation (D,F) - only equilibrium values were given for knockout experiments in DREAM4 (B). Edges denote effect strength (thickness) and sign (activation = red, inhibition = blue).
doi:10.1371/journal.pone.0012807.g006

are of length 130. The weighted ρ_i is calculated based on the weighted covariance and the weighted mean (not shown). A model coefficient ρ is calculated as the average of the gene coefficients ρ_i . In addition, we introduced a regularization parameter reg . It allows us to control $|Network|$, i.e. the number of edges (= effect arcs) in the models.

$$dist = [1 - \rho(x,y,w)]^2 + reg \cdot |Network| \quad (7)$$

Note that DREAM4 provided only equilibrium values for the knockout, knockdown and multifactorial datasets. Only for the time course datasets gene levels for different measurements were available and are used for the calculation of $dist$.

Simulated annealing. We employ simulated annealing to decide if a network changing move is accepted or rejected. That is, we always accept moves that improve the network with respect to

the objective function $dist$. We accept inferior networks with a probability p calculated from the Boltzmann distribution parametrized by k (Table 1). Essentially, moves that only slightly increase $dist$ are accepted more frequently, especially if the temperature T is high. T decreases linearly during the

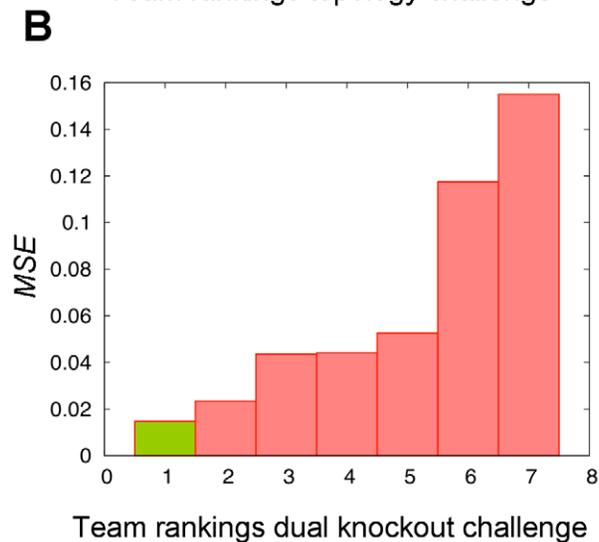
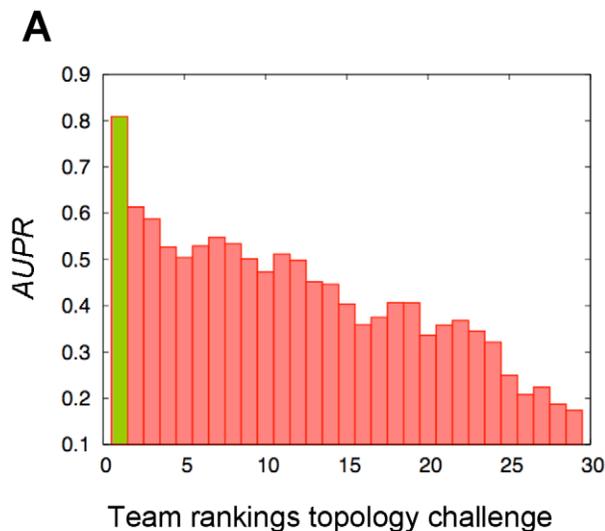
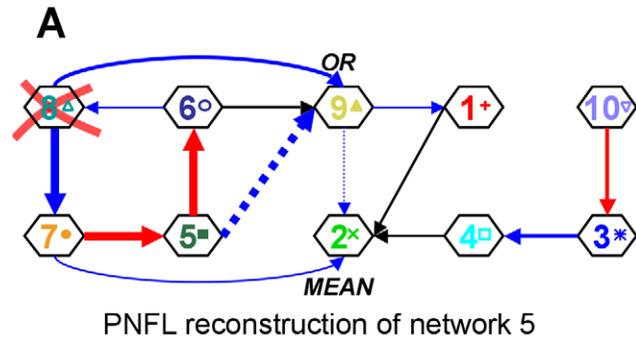


Figure 7. Evaluation of the *in silico* challenge comprising five networks of ten genes. Panel A shows the prediction performance of the directed unsigned topology as the area under the precision recall curve (AUPR). In a bonus challenge, steady-state level predictions of dual knockout experiments were evaluated by the mean squared error (MSE, panel B). Our performance is shown in green. doi:10.1371/journal.pone.0012807.g007

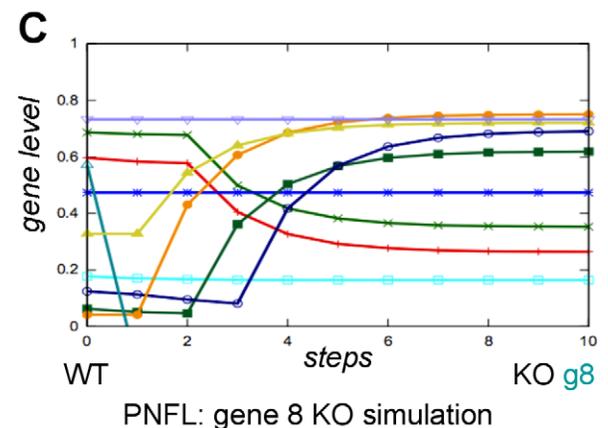
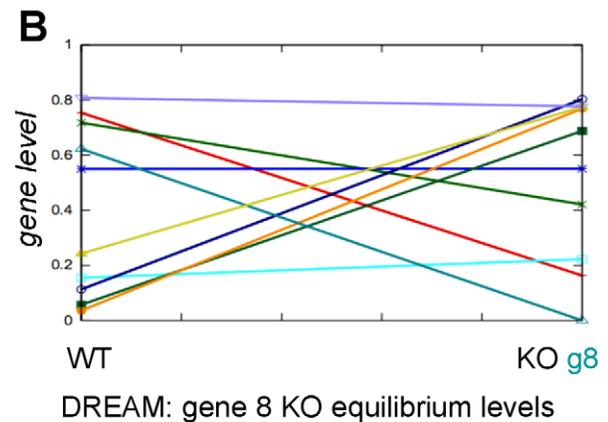


Figure 8. PNFL reconstruction of network 5 (AUPR=76%). DREAM4 evaluated our predictions (panel A) in terms of correct (colored solid), missed (black) and surplus (dotted) edges. For simulation, we also infer three levels of effect strength (edge thickness) for both activation (red) and inhibition (blue). Targets regulated by multiple effectors are parametrized by the kind of regulation, i.e. dependent (AND, OR) vs. independent (MEAN). Incorrect predictions are more frequent when effector gene levels are low in the wild type (e.g. genes 4, 5, 6 and 9). In panels A and B we compare the provided DREAM data to the PNFL simulation for the knockout of gene 8. doi:10.1371/journal.pone.0012807.g008

reconstruction run from one to zero.

$$p = e^{-k \cdot \Delta f / T}, T \in [1..0] \quad (8)$$

Submission. The DREAM4 submission format required to rank effects by their prediction confidence. We therefore chose a consensus approach to predict an ensemble of networks. Consensus prediction approaches have been successfully applied to network reconstruction before [31]. We carry out 100 reconstruction runs with different parameter settings (Table 1) and random seeds. Ranking is based on the effect prediction confidence calculated as the fraction of networks that included the given effect. This automated approach was the same for each of the submitted nets. No manual processing was performed. For the visualization and description of effects or networks, we assume an effect to be predicted if the prediction confidence is 50% or above.

Results

Size of the model space

The number of possible models m depends on the number of genes $g = 10$, the number of time courses $tc = 5$, the number of rule tables $r = 6$ (Fig. 3) and the number of combination operators $|op| = 3$ (eq. 4). The number of models m depends on the number of time courses tc because each time course introduces an additional perturbation variable. As we do not restrict the number of effectors, a gene can be affected by zero, one or up to $n = g - 1 + tc = 14$ variables, i.e. 9 other genes (as self interactions were not allowed) and 5 perturbation variables (eq. 6).

$$m = \left[1 + n \cdot r + |op| \cdot \sum_{k=2}^n \binom{n}{k} \cdot r^k \right]^g \quad (9)$$

In the given setting, the size of the model space is $1.2 \cdot 10^{123}$. Thus, a heuristic search strategy is necessary to detect high scoring networks.

Reconstruction run time

The most time expensive steps are the simulations needed to calculate the objective function after each move. Each move requires 35 simulations, i.e. 5 time courses, 10 knockouts, 10 knockdowns and 10 multifactorials. A typical reconstruction run consists of 2500 moves (1ms per move and network) on a population of 25 individual networks and takes about a minute on

a single processor core. During the run, the individuals in the population usually converge to a single network with only minor variations (data not shown).

Relative contribution of the different datasets

The contribution of the different datasets to the prediction of networks is given by dataset specific weights. The weights were derived manually based on randomly generated PNFL models. The relative contributions of the individual datasets amount to KO = 29% ($w_{KO} \cdot 10$ data points = 80; compare eq. 7 and Table 1), KD = 21% ($w_{KD} \cdot 10 = 60$), TC = 36% ($w_{TC} \cdot 100 = 100$) and MF = 14% ($w_{MF} \cdot 10 = 40$). While the combination of KO+KD accounts for half of the total dataset weights, the largest individual portion stems from the TC data.

DREAM4 evaluation results

The overall results of the *in silico* size ten challenge as reported by the DREAM organizers are depicted in Fig. 7. The network topology was predicted by 29 different teams. In terms of the AUPR, our PNFL based reconstruction approach (81% AUPR averaged over 5 networks) outperformed the second best team by 20 percentage points. Our approach performed best on four of the five networks and second best on the remaining network. In an additional challenge, steady state gene levels in response to double knockout mutations were predicted. This evaluated the ability to predict the behavior of networks under previously unseen experimental conditions. Only 7 teams participated in the double knockout predictions (Fig. 7B) where PNFL also was the top performer.

Reconstruction of network 5

Our reconstruction of network 5 (Fig. 8A) achieved an AUPR of 76%. The panels B and C in Fig. 8 compare the provided data (DREAM) to the PNFL simulation for the knockout of gene 8. Genes up (e.g. gene 7) or down regulated (e.g. gene 1) are captured correctly in the PNFL simulation. To simplify the representation of networks, transitions and the corresponding output places (compare Fig. 1) are merged into single nodes depicted as hexagons.

Network 5 demonstrates the utility of the multifactorial data (Fig. 9) for network reconstruction. According to personal communication at the joint RECOMB/DREAM conference 2009, several participants neglected to utilize this kind of data. The four-gene cycle (genes $5 \rightarrow 6 \rightarrow 8 \rightarrow 7 \rightarrow 5$, Fig. 8A) in network 5 is an example for a difficult network motif that our approach predicts correctly only if the multifactorial data is included.

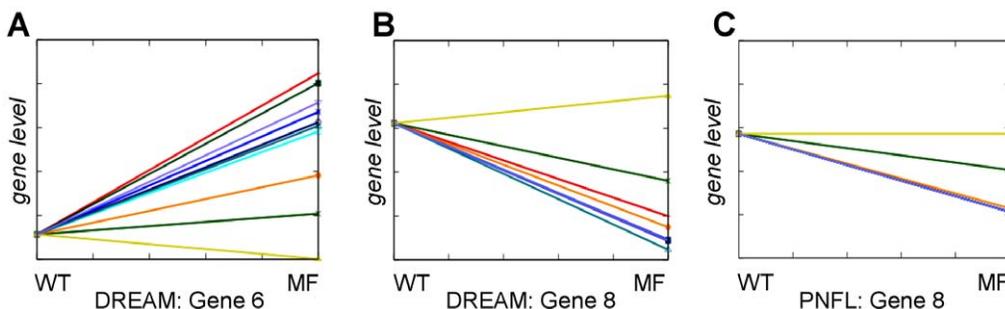


Figure 9. Generation of multifactorial (MF) data for an effect in network 5. In network 5, gene 6 is the only effector for gene 8 (see Fig. 8). Effectors are initialized by the provided MF gene levels (A). Subsequently, individual PNFL transitions are applied to compute the MF gene levels for the targets (C). The objective function compares the target gene levels of the provided MF data (B) to the PNFL outputs (C). doi:10.1371/journal.pone.0012807.g009

Incorrect predictions were more likely when the effector gene levels were low in the wild type (e.g. genes 4, 5, 6 and 9 in network 5). Here, predictions frequently contain shortcuts with respect to the true topology (Fig. 8, correct: $5 \rightarrow 6 \rightarrow 9$, predicted $6 \leftarrow 5 \rightarrow 9$; correct: $9 \rightarrow 1 \rightarrow 2$, predicted: $1 \leftarrow 9 \rightarrow 2$; see also Fig. 10, correct: $9 \rightarrow 10 \rightarrow 3$, predicted: $10 \leftarrow 9 \rightarrow 3$). This leads to two errors: As the effect $6 \rightarrow 9$ can not be directly observed in the given data it is missed as it is already ‘explained’ by an incorrectly predicted

surplus effect (here: $5 \rightarrow 9$). Such a missing observation can for instance be due to knockouts or knockdowns exhibiting no substantial effect because of low wild type gene levels.

Reconstruction of network 1

The reconstruction of network 1 (Fig. 10A) achieved a very high AUPR of 92%. Here, we predicted 14 out of 15 effects correctly. For a correct reproduction of time course data (e.g. time course 2 in Fig. 10, compare panels B and C) we also infer perturbation target genes. According to our reconstruction, the perturbation p_2 in time course 2 affects genes 3 and 7.

Network 1 was selected to demonstrate the capability of PNFL to represent oscillating network motifs. Oscillations require cycles that seem to pose no particular difficulty for the PNFL based reconstruction. Each of the three nested cycles contained in network 1 (genes $3 \leftrightarrow 4$, $3 \leftrightarrow 7$, $3 \rightarrow 7 \rightarrow 4 \rightarrow 3$) was resolved correctly. In addition, genes 3, 4 and 7 were recognized as an oscillation generating network motif. The removal of the perturbation triggers oscillations for instance in gene 7, which was picked up clearly in the PNFL simulation (Fig. 10, panels B and C).

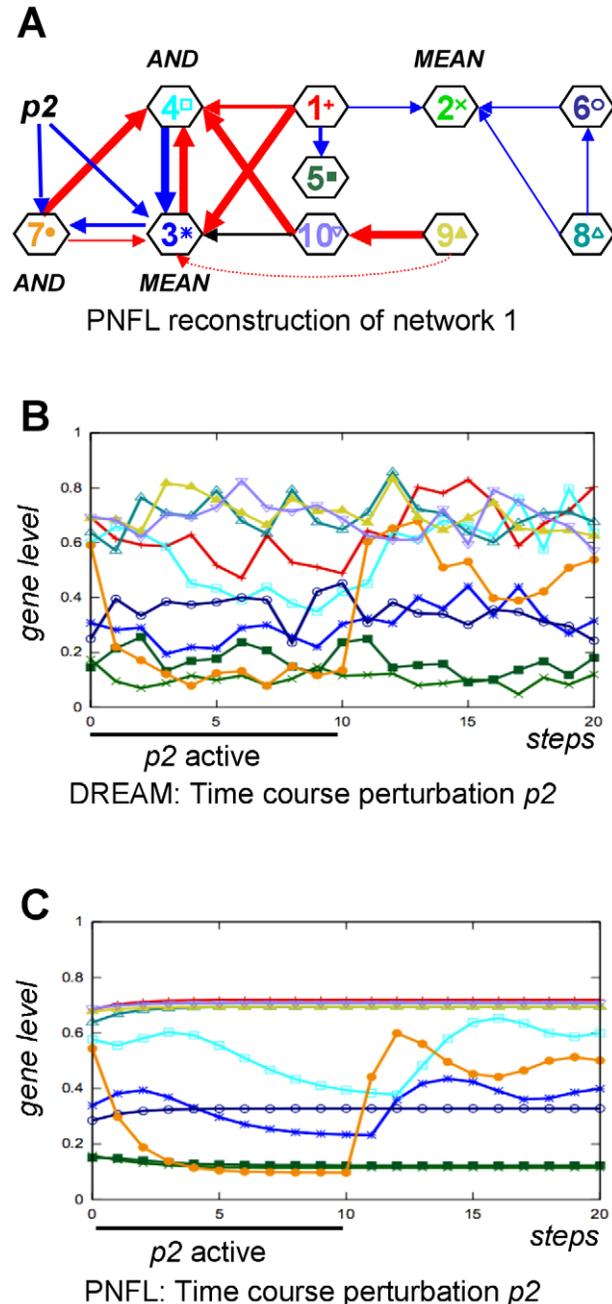


Figure 10. PNFL reconstruction of network 1 (AUPR=92%). Shown is our reconstruction of network 1 (A) and the data of time course 2 as provided by DREAM (B) or simulated by PNFL (C). Time course data shows how the network responds to the application and removal of perturbations. In addition to effector targets (eq. 4), we also predict perturbation targets (eq. 6). According to our reconstruction, perturbation p_2 in time course 2 affects genes 3 and 7. doi:10.1371/journal.pone.0012807.g010

Validation of effect signs

The validation described in this and the following subsections is based on supplementary material posted after the completion of DREAM4 (<http://gnw.sourceforge.net/resources/DREAM4%20in%20silico%20challenge.zip>). It for instance enables the validation of the signs of the effects in the models, i.e. if a target is activated or inhibited by a given effector. Effect signs are determined by the effector rule tables (see Application of effector rules) selected during PNFL reconstruction. Sign predictions can only be evaluated for correct effector-target predictions. Here, the signs were predicted correctly in 100% of the cases.

Validation of the regulatory logic

Logical operations are used by both PNFL and DREAM4 to combine the effects of multiple effectors on a given target gene. Thereby, dependent (*AND*, *OR*) and independent (*MEAN*) kinds of regulation are distinguished (see Fig. 5). In the DREAM4 setting, arbitrary combinations of these operators are possible. For instance, the activation state of gene 3 in network 1 (Fig. 5) is described by a combination of *AND* and *MEAN* operators. This is not possible by PNFL as it currently allows only one operator per target gene. Note that this does not apply to the effect signs, which are assigned to each effect separately by PNFL as well as DREAM4.

A rigorous comparison between PNFL and DREAM4 logic is not possible if more than one operator is involved as in Fig. 5. We then consider PNFL and DREAM4 regulatory logics as approximately equal if the operator that combines the majority of terms is predicted by PNFL. In the example of Fig. 5, *MEAN* (combining three terms, i.e. the three CRMs) would be correct whereas *AND* (combining two terms, i.e. TF1 and TF7) and *OR* (not used here) would be incorrect. According to this, our predictions are correct in 13 out of the 18 targets (72%) regulated by multiple effectors. Three of the five mismatches are explained by topological errors. Here, the corresponding target genes are connected to single effectors in the PNFL models and to multiple effectors in the DREAM4 network. Thus, if the predicted topology permits the inference of the regulatory logic it is correct in 87% (= 13/15) of the cases.

Validation of time course perturbation targets

The time courses emerge from perturbations that affect a specific subset of target genes. The PNFL based simulation of the

time courses thus required the prediction of the targets of a perturbation (see Time course data). The evaluation of our prediction performance on inferring the time course perturbation targets resulted in an AUPR of 73%. The performance difference to the prediction of effector targets (AUPR of 81%) is due to the fact that each perturbation corresponds to a single time course. The remaining four time courses (and all of the KO, KD and MF datasets) do not provide any information with regard to the targets of a selected perturbation.

Discussion

We presented a method for network reconstruction that uses Petri Nets with Fuzzy Logic (PNFL) for modeling and simulation. This approach was the best performer (Fig. 7) in the *in silico* size ten challenge of the 2009 DREAM4 assessment of reverse engineering methods. Why did it work so well?

Our approach optimizes models to achieve the best possible agreement between PNFL generated datasets and the datasets provided in the DREAM4 challenges. To get the most out of the data, we employ specific simulation approaches for each of the available datasets. This allows us to exploit and score heterogeneous datasets in a unified way. We further reduce the model complexity severely to avoid the risk of overfitting. Ideally, only a single network should be able to reproduce the data. The model space is still huge, requiring a heuristic, population based search strategy. It avoids local minima traps and thus improves the convergence of networks and also the agreement between PNFL and DREAM datasets. The resulting PNFL models accurately predict the network behavior even under new experimental conditions not seen during model building. This was demonstrated in the double knockout challenge (Fig. 7B).

Incorrect predictions might result when the effector gene levels are low in the wild type. Knockout experiments, for instance, provide only little topological information in such cases. This is particularly frequent in network 2 (data not shown) where different network topologies generate similar data and our reconstruction does not converge to a single network. Indeed, no team achieved a good prediction performance for network 2. In case of genes with

low wild type expression, over-expression instead of knockout experiments should be performed.

Several of the participating teams focused on the knockout (KO) datasets and neglected to exploit the time course (TC), multifactorial (MF) and knockdown (KD) data in their reconstruction (personal communication at the joint RECOMB/DREAM conference 2009). We found that only the combination of all provided datasets enabled us to predict particularly difficult network motifs. An example is the unusual four-node cycle in Fig. 8A that is predicted correctly only when using the MF data (Fig. 9). In general, cycles and nested cycles pose no particular difficulty to our PNFL based approach (e.g. Fig. 10A). The time course shown in Fig. 10 demonstrates that our reconstruction also resolves and recognizes oscillating network motifs.

The reconstruction of PNFL models reliably determines a range of mechanistic details that go beyond the graph topology evaluated in DREAM. Our models distinguish activation from inhibition, dependent from independent regulation as well as strong, medium and weak degrees of effect strength. Such intuitive assertions are sufficient to specify, visualize and thus comprehend executable models and their parameters. This is a characteristic feature of fuzzy logic modeling [18]. Similarly intuitive notions are more difficult if not impossible to obtain from ODE, mutual information or Bayesian models. Nevertheless, both PNFL and ODE enable the detailed simulation of models. Simulation models can facilitate an iterative cycle of model improvements based on the comparison between *in silico* and laboratory experiments.

Acknowledgments

We acknowledge Florian Erhard for the stimulating discussions on the PNFL system. We thank all reviewers for their suggestions, which substantially improved the manuscript.

Author Contributions

Conceived and designed the experiments: RK TP LW RZ. Performed the experiments: RK. Analyzed the data: RK TP LW RZ. Wrote the paper: RK. Implemented the software: RK.

References

- de Jong H (2002) Modeling and simulation of genetic regulatory systems: a literature review. *J Comput Biol* 9(1): 67–103.
- Brazhnik P, de la Fuente A, Mendes P (2002) Gene networks: how to put the function in genomics. *Trends Biotechnol* 20(11): 467–72.
- Bansal M, Belcastro V, Ambesi-Impombato A, di Bernardo D (2007) How to infer gene networks from expression profiles. *Mol Syst Biol* 3: 78.
- Markowitz F, Spang R (2007) Inferring cellular networks—a review. *BMC Bioinformatics* 8 Suppl 6: S5.
- Li H, Xuan J, Wang Y, Zhan M (2008) Inferring regulatory networks. *Front Biosci* 13: 263–75.
- Hecker M, Lambeck S, Toepfer S, van Someren E, Guthke R (2008) Gene regulatory network inference: data integration in dynamic models—a review. *Biosystems* 96(1): 86–103.
- Sima C, Hua J, Jung S (2009) Inference of gene regulatory networks using time-series data: a survey. *Curr Genomics* 10(6): 416–29.
- de la Fuente A, Brazhnik P, Mendes P (2002) Linking the genes: inferring quantitative gene networks from microarray data. *Trends Genet* 18(8): 395–8.
- Gardner TS, di Bernardo D, Lorenz D, Collins JJ (2003) Inferring genetic networks and identifying compound mode of action via expression profiling. *Science* 301(5629): 102–5.
- Nachman I, Regev A, Friedman N (2004) Inferring quantitative models of regulatory networks from expression data. *Bioinformatics* 20 Suppl 1: i248–56.
- Margolin A, Nemenman I, Basso K, Wiggins C, Stolovitzky G, et al. (2006) ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context. *BMC Bioinformatics* 7(Suppl 1): S7.
- Friedman N, Linial M, Nachman I, Pe'er D (2000) Using Bayesian networks to analyze expression data. *Journal of Computational Biology* 7(3): 601–620.
- Soranzo N, Bianconi G, Altafini C (2007) Comparing association network algorithms for reverse engineering of large-scale gene regulatory networks: synthetic versus real data. *Bioinformatics* 23(13): 1640–7.
- Hache H, Lehrach H, Herwig R (2009) Reverse Engineering of Gene Regulatory Networks: A Comparative Study. *EURASIP J Bioinform Syst Biol* 2009: 617281.
- Michoel T, De Smet R, Joshi A, Van de Peer Y, Marchal K (2009) Comparative analysis of module-based versus direct methods for reverse-engineering transcriptional regulatory networks. *BMC Syst Biol* 3: 49.
- Zou C, Denby KJ, Feng J (2009) Granger causality vs. dynamic Bayesian network inference: a comparative study. *BMC Bioinformatics* 10: 122.
- Marbach D, Prill RJ, Schaffter T, Mattiussi C, Floreano D, et al. (2010) Revealing strengths and weaknesses of methods for gene network inference. *Proc Natl Acad Sci U S A* 107(14): 6286–91.
- Windhager L, Zimmer R (2008) Intuitive Modeling of Dynamic Systems with Petri Nets and Fuzzy Logic. German Conference on Bioinformatics, Lecture Notes in Informatics P-136: 106–115.
- Prill RJ, Marbach D, Saez-Rodriguez J, Sorger PK, Alexopoulos LG, et al. (2010) Towards a rigorous assessment of systems biology models: the DREAM3 challenges. *PLoS One* 5(2): e9202.
- Marbach D, Schaffter T, Mattiussi C, Floreano D (2009) Generating realistic *in silico* gene networks for performance assessment of reverse engineering methods. *J Comput Biol* 16(2): 229–39.
- Sackmann A, Heiner M, Koch I (2006) Application of Petri net based analysis techniques to signal transduction pathways. *BMC Bioinformatics* 7: 482.
- Heiner M, Koch I, Will J (2004) Model validation of biological pathways using Petri nets—demonstrated for apoptosis. *Biosystems* 75(1–3): 15–28.
- Chen L, Qi-Wei G, Nakata M, Matsuno H, Miyano S (2007) Modelling and simulation of signal transductions in an apoptosis pathway by using timed Petri nets. *Journal of Biosciences* 32(1): 113–127.
- Marwan W, Wagler A, Weismantel R (2008) A mathematical approach to solve the network reconstruction problem. *Mathematical Methods of Operations Research* 67(1): 117–132.

25. Murata T (1989) Petri nets: Properties, analysis and applications. *Proceedings of the IEEE* 77(4): 541–580.
26. Chaouiya C (2007) Petri net modelling of biological networks. *Brief Bioinformatics* 8(4): 210–219.
27. Matsuno H, Tanaka Y, Aoshima H, Doi A, Matsui M, et al. (2003) Biopathways representation and simulation on hybrid functional Petri net. *In Silico Biol* 3(3): 339–404.
28. Zadeh LA (1965) Fuzzy sets. *Information and Control* 8: 338–353.
29. Mendel LM (1995) Fuzzy logic systems for engineering: a tutorial. *Proceedings of the IEEE* 83(3): 345–377.
30. Istrail S, Davidson EH (2005) Logic functions of the genomic cis-regulatory code. *Proc Natl Acad Sci U S A* 102(14): 4954–9.
31. Marbach D, Mattiussi C, Floreano D (2009) Combining multiple results of a reverse-engineering algorithm: application to the DREAM five-gene network challenge. *Ann N Y Acad Sci* 1158: 102–13.