Genome Biology

Genome Biology
Biology for the post-genomic era

MetaReg: A platform for modeling, analysis and visualization of biological systems using large-scale experimental data

Genome Biology 2008, **9**:R1 doi:10.1186/gb-2008-9-1-r1

Igor Ulitsky (ulitskyi@tau.ac.il)
Irit Gat-Viks (gat-viks@molgen.mpg.de)
Ron Shamir (rshamir@tau.ac.il)

ISSN	1465-6906
Article type	Software
Submission date	4 July 2007
Acceptance date	2 January 2008
Publication date	2 January 2008
Article URL	http://genomebiology.com/2008/9/1/R1

MetaReg: A platform for modeling, analysis and visualization of biological systems using large-scale experimental data

Igor Ulitsky[*,1], Irit Gat-Viks[*,1,2] and Ron Shamir[1,3]

[1]School of Computer Science, Tel Aviv University
[2]Computational Molecular Biology Department, Max Planck Institute for Molecular Genetics, Berlin, Germany.
[3] Corresponding author, rshamir@post.tau.ac.il
E-mail addresses: ulitskyi@post.tau.ac.il, gat-viks@molgen.mpg.de, rshamir@post.tau.ac.il

* These authors contributed equally to this work

MetaReg (http://acgt.cs.tau.ac.il/metareg/application.html) is a computational tool that models cellular networks and integrates experimental results with such models. MetaReg represents established knowledge about a biological system, available today mostly in informal form in the literature, as probabilistic network models with underlying combinatorial regulatory logic. MetaReg enables contrasting predictions with measurements, model improvements and studying what-if scenarios. By summarizing prior knowledge and providing visual and computational aids, it helps the expert explore and understand her system better.

1. Rationale

Given the recent accumulation of high throughput biological data, the task of integrating and analyzing large-scale datasets is a major challenge. A variety of computational modeling approaches have been developed for the analysis of such datasets, such as clustering [1, 2] and topological interaction network models [3, 4]. While these approaches give a broad, low resolution picture of cellular processes, many biologists are interested in a specific subsystem, and wish to use the results from experiments in order to refine the current knowledge on the system. This analysis of data in the context of the available knowledge is often performed in an informal manner: The researcher sketches a diagram of a relevant subsystem according to the current knowledge. This diagram summarizes and organizes the available knowledge, and assists the expert in analyzing the predicted state of the system in various possible experiments. The predictions are then compared to experimental measurements, and in case a discrepancy is found, additional experiments are performed, and the diagram is iteratively refined.

In case of complex biological systems and massive amount of data, manual construction of the model, state predictions, comparison with data and systematic model refinements are impractical, and automatic computational methodologies must be employed [5, 6]. To address the need for such analysis workflow, we developed MetaReg, an integrative tool for analysis of steady-state high throughput data in the context of specific biological system. The theoretical foundations of the MetaReg methodology and algorithms are outlined in Section 4 (for a complete description, see [7]). While making some gross simplifying assumptions over real biological systems behavior, the model was demonstrated to be highly effective on several systems [7-9]. MetaReg enables easy conversion of the current qualitative knowledge on a particular subsystem into a mathematical model, including logical relations among the biological components. The system is represented by a probabilistic graphical model called a Bayesian network [10], which allows distinguishing between regulatory relations that are known at a high level of certainty and those that are more speculative. Given the model, MetaReg predicts the level of each variable under any given genetic perturbation or environmental stimuli. Moreover, MetaReg allows incorporation of high throughput data, and graphical comparison between model predictions and measurements. The most advanced MetaReg capability is suggesting model refinements by systematically seeking changes that increase the fit between model predictions and experimental measurements.

2. MetaReg Application

2.1 MetaReg core functionality

Figure 1 illustrates the key features of MetaReg application and its workflow. The basic workflow begins with model construction and its initial analysis through simulations. Once a current-knowledge model is established, it can be used to predict component values under any experimental treatment (e.g., genetic perturbation, growth environment). Next, we compare these predictions to the values observed in the actual experiments under the same treatments, and highlight the discrepancies between them graphically. MetaReg can also automatically refine the model in order to reduce such discrepancies. Screenshots of the main windows from the application are shown in **Figure 2**. A comprehensive manual of the application is available online (http://acgt.cs.tau.ac.il/metareg/manual/).

2.2 Model construction

The first step in utilizing MetaReg is the construction of the biological system model on the *model canvas* (**Figure 2A**). A MetaReg model consists of a set of biological variables and their regulatory logics. The *variables* represent different biological entities (mRNA, protein, metabolite, etc.). Each variable may attain several discrete *states* (three states by default), representing, for example, the transcript level of an mRNA, or the activity level of an enzyme. The state of a variable *v* is influenced by the states of the variables that are connected to *v* by incoming edges. These variables are called the *regulators* of *v*. Most importantly, every variable is assigned a discrete *logic*, which defines its state given the states of its regulators. For example, if variable A has two activators B and C, its logic might be *Max(B,C)*. We assume all the logics represent steady-state regulatory relations, and thus the model represents the steady-state behavior of the biological system. Every logic is associated with a probability that indicates the certainty in the prior biological knowledge. For example, if a logic is known with high certainty, it will be assigned with a high probability (e.g., 90%), and alternative logics will have low probabilities.

The application offers several tools to help in model construction. Variables can be selected from and automatically linked to known databases such as SGD [11] and NCBI Gene [12] (**Figure 2B**). Each variable can be attributed with links to relevant journal publications from PubMed, enabling further model curation. The application provides several gadgets for logic definition, including scripting, a tabular editor and a logic wizard (**Figure 2C**) for hierarchical construction of complex logics. The type of each regulation, activation (→), repression (—┤) or other (–O) is automatically deduced based on the logic of the regulatee (the regulated variable). The model canvas is fully interactive, including capabilities for manual or automatic variable positioning and highlighting of different sets of variables, such as all the metabolites or all the cycles in the model.

2.3 Model simulation

In order to view the behavior of the model in response to different experimental treatments, simulations can be performed. Given a particular experimental treatment, the possible system states are computed as described in [8]. A *system state* is an assignment of states to all the variables in the model. The user can dynamically design an experimental treatment scenario and visually analyze the system state on the model canvas. If the model contains cycles, several system states might be feasible, and the user can navigate among them.

2.4 Data integration

The application can integrate *observations* (measurements) from multiple studies. The measured biological components are automatically matched to the model variables. For example, gene expression data are automatically matched to the corresponding mRNA variables, and protein measurements are matched to the corresponding protein variables. As part of the data import, the user must specify the *experimental treatment* used in each experiment, including the environmental stimulations and genetic perturbations performed in each particular experiment. For example, if the experiment was performed in surplus of nitrogen and on a yeast strain where Leu3 is knocked out, the experimental treatment is "Leu3=0; Nitrogen=2", where Leu3 and Nitrogen are model variables. Once the data are imported, it is possible to visualize all measured variables under each of the experiments in a single data matrix (**Figure 2D**, see below), or to view the measurements of a specific experiment projected on the model canvas (**Figure 2A**).

2.5 Comparing predictions with observations

In order to evaluate the model, the *predicted* levels of each variable are compared to its *observed* levels under each experiment. MetaReg provides a prediction engine that infers probabilistically the expected level of each variable in each experiment, given the network model and the experimental treatment (see [7]). MetaReg supports two visualization tools to compare these predictions with the observations, both designed to highlight cases of discrepancies, which are often the starting point of further research. First, the observed and the predicted values for a single experiment can be projected side by side on the model canvas (**Figure 2A**). The second visualization tool provides a comprehensive view of the discrepancies across all the experiments, in which each cell contains color-coded representation of the observed and the predicted values, along with a representation of the discrepancy between them (**Figure 2D**). This view allows simple detection of discrepancy "hot-spots" in which the model fails to explain the data.

2.6 Model refinement

Our methodology enables refinement of the model to obtain better fit between model predictions and observations. The input of the refinement process is the target variable and a set of regulators. MetaReg searches among all possible regulatory logics and outputs the most significant one. The suggested logic can be further edited by the user (**Figure 2E**). This way the user can test hypotheses about variable regulation.

3. Case study: Leucine biosynthesis in *S. cerevisiae*

3.1 Modeling and Simulations

We present a model for leucine biosynthesis and related signaling pathways in *Saccharomyces cerevisiae*. Building on literature reports, we constructed a detailed model of known regulatory relations in this system. The model contains 47 variables (nodes) and 67 regulations (arcs). The model is available from our web site (http://acgt.cs.tau.ac.il/metareg/application.html). Leucine is an essential branched-chain amino acid generated from pyruvate via α-ketoisovalerate (KIV), α-IPM and β-IPM in a linear pathway in which nine catalyzing enzymes are involved (Ilv2, Ilv3, Ilv5, Leu9, Leu4, Leu1, Leu2, Bat1, Bat2). The regulation of leucine production is controlled by several known mechanisms [13]:

(1) Transcription regulation of several leucine biosynthetic enzymes via the general regulatory pathway of amino acid biosynthesis. Starvation for any amino acid induces the translation of Gcn4 via Gcn2. Gcn4 is a transcriptional activator of enzymes which catalyze several amino acid biosynthesis pathways, including the leucine biosynthetic pathway.

(2) The control of several catalyzing enzymes is regulated by the transcriptional activator Leu3. The activity of Leu3 is regulated by α-IPM, an intermediate of the pathway acting as a co-inducer. When α-IPM is present, Leu3 acts as activator; when α-IPM is absent, Leu3 acts as repressor [13]. Hence, α-IPM serves as a sensor of leucine production.

(3) The enzymatic activity of Leu4 is subject to two major controls by metabolites. The first is feedback (end product) inhibition by leucine. At high levels of leucine, Leu4 activity is inhibited, and causes a reduction in the production of the pathway. The second control is inactivation by CoA, a product of the reaction catalyzed by Leu4 and a central energy metabolite in the mitochondria. This control serves as a link between the metabolic process and the energy metabolism context.

In **Figure 3**, we present a diagram of our model. It includes the leucine biosynthetic pathway, the catalyzing enzymes and their transcription control. The state of internal leucine depends on the leucine transport into the cell and on the yield of the leucine biosynthetic pathway. The transport is facilitated via amino acid permeases (Bap2, Bap3, Gap1, Tat1) that are regulated by Gcn4, Leu3, and the TOR signaling pathways. The model includes four environmental stimulators: 'NH_3' (ammonium), 'rapamycin', 'leucine', and 'amino acids', which indicates availability of all amino acids except leucine that are needed to represent the environmental conditions enforced on the system. The model graph contains many cycles. For example, the general nitrogen control regulation (e.g. Gcn4 → biosynthetic enzymes → leucine biosynthesis pathway → internal amino acids → Gcn2 → Gcn4), the leucine-specific transcriptional regulation via Leu3 (Leu3 → biosynthetic enzymes → leucine biosynthesis pathway →α-IPM → Leu3), and autoregulation of Leu3 TF on the LEU3 gene transcription (LEU3 <-> Leu3). The variables that are part of cycles in the model are highlighted in **Figure 3**.

We used three states for each mRNA variable: state '0' represents reduced transcription level compared to the wild-type, state '1' represents the wild-type transcription level when cells are grown on YPD medium, and state '2' represents increased transcription level. Similarly, each protein has three states reflecting its activity level (high = '2', medium = '1', low = '0'). The modeling of Leu3 is a special case, since we had to represent its dual role as activator and repressor. We used state '0' for its repressive mode, state '1' represents no effect (e.g., in *leu3* mutant), and state '2' indicates the Leu3 activator mode. For example, a simulation of the system behavior in leucine starvation is shown in **Figure 3**.

3.2 Data Preparation

We integrated expression profiles from four datasets that contain treatments pertinent to our model: (a) seven profiles in rapamycin treatment after 15, 30, 60 and 120 minutes of incubation and in amino acid deprivation after 1, 1.5 and 2 hours of incubation [14]. (b) six profiles in histidine starvation and various Gcn4 perturbations [15]. (c) six profiles of chemostat growth in nitrogen limiting conditions with and without Leu3 perturbation [16] (d) six profiles in nitrogen depletion after 8, 12 and 24 hours of treatment and in amino acid and adenine starvation after 1, 2 and 4 hours of treatment [17]. A complete description of the profiles, the experimental treatments under which they were obtained and the data preprocessing, is available in **Supplement A**.

3.3 Evaluation of the model in accordance to data

We executed the prediction engine of MetaReg on the data for each of the profiles described above. The matches and mismatches between the predictions and the observations are displayed by the discrepancy matrix in **Figure 4b**. While there is a good match for the majority of the components and conditions, the matrix reveals several major discrepancies between the model and the microarray experiments:

- The leucine biosynthetic genes LEU1, LEU2, LEU4 and BAP2 show unexpected expression decrease in the *leu3* mutant strain (**Figure 4B**, column 16-18). The reduction was surprising since Leu3 is known to act as a repressor in these experiments.
- In *gcn4* mutant strains, we observe an increase in the levels of the leucine biosynthetic genes BAT1, ILV2, ILV3 and ILV5 following 3AT treatment (histidine starvation) (**Figure 4A**, columns 11-12). In our model the effect of general amino acid control on these genes is mediated solely by Gcn4. Since Gcn4 is absent in these experiments, our model does not predict such an increase, and a discrepancy appears (**Figure 4B**, column 12).
- For LEU3, we observed an increase in expression in two *gcn4* mutant strains and in nitrogen limitation experiments (**Figure 4B**, row 11, columns 11-15). According to the literature, LEU3 mRNA is activated by either Gcn4 or Leu3 TFs. As no amino acid shortage occurs in these experiments, neither Gcn4 nor Leu3 are expected to be active, hence the model predicts low level of LEU3 mRNA, in contradiction to the observed increase.
- Following a rapamycin treatment we observed a consistent decrease in the levels of four biosynthetic genes: BAT1, ILV3, ILV5 and LEU1. The effect of rapamycin on the biosynthetic genes is known to be mediated by the TOR pathway through Gcn4 [18]. It is thus expected that under rapamycin treatment, Gcn4 will be active, while Leu3 will not be active. Consequently, the levels of the leucine biosynthesis genes (LEU1, ILV3, ILV5, BAT1) regulated by Gcn4 should be alleviated. Surprisingly, we witness a down-regulation of these genes.
- For LEU9, BAT2, BAP3 and TAT1, we could not find any report on their regulation in the literature, and thus their predicted level is constant. Hence, their discrepancies merely reflect the lack of knowledge about them.

3.4 Leucine model refinement

In order to improve the fit of the model's predictions to the observed data we used MetaReg's refinement algorithm. We focus here on two representative examples of model refinement. In these examples we suggest improved logics for the way by which Leu3 and Gcn4 jointly regulate LEU9, BAT2 and LEU2.

- LEU9 and BAT2 have similar expression patterns (**Figure 4A**), but we could not find any report on their regulation in the literature. MetaReg suggests that LEU9 is regulated solely by Leu3 with no definite regulatory role for Gcn4 (**Figure 5A**, LEU9 table, rows 1 and 3). A similar logic is obtained for BAT2. Note that for Leu3, MetaReg's refinement matches its known repressive role: When Leu3 acts as a repressor (Leu3=0), we observed medium/low transcription of LEU9, even though the level of the activator Gcn4 is high (**Figure 5A,** LEU9 table).

- LEU2 expression is known to be affected only by Leu3 [13]. Indeed, the suggested logic (**Figure 5**) shows that the state of Gcn4 does not influence Leu2. As expected, when Leu3 should act as activator (Leu3='2') there is high transcription (LEU2='2'). However, we do not detect the expected repressive effect of Leu3 on its targets. When Leu3 should act as repressor (Leu3='0'), we observe medium LEU2 transcription (LEU2='1') instead of the expected low transcription.

Figures 5B-C illustrate the refinement process. During refinement, MetaReg tests the predicted activity levels of the TFs (Gcn4 and Leu3) against the observed level of the mRNA in each experiment (**Figure 5B**), and computes the best logic between the regulators' predicted level and the observations. Consequently, the discrepancies observed for LEU2 and LEU9 in our initial model (before refinement) are drastically reduced after refinement (**Figure 5C**).

In the case of LEU1, BAT1, LEU4, ILV2, ILV3 and ILV5, the results were similar to LEU2 (not shown). For BAP2, BAP3, TAT1 and LEU3, MetaReg did not succeed to derive a high confidence logical relation, due to inconsistent effects that could not be explained by the model. For example, for TAT1, only down-regulation is observed in the data (**Figure 4A**, last row). For BAP3, we observe an inconsistency between two sets of nitrogen depletion experiments in different studies (**Figure 4A**, columns 13-15 vs. 19-21). This probably indicates that each of those genes is regulated by additional elements that are not included in the model.

4. MetaReg’s algorithmic layer

In this section, we briefly outline the algorithmic layer behind the MetaReg application. A full description can be found in[7].

Modeling prior knowledge - Our model consists of variables $X_1 \ldots X_n$, represented by nodes, and regulations among them, represented by arcs. The set of variables that together regulate variable X_i are called its *regulatory unit,* denoted Pa_i. This is the set of nodes that have arcs directed into X_i. Each variable can be in one of several discrete *states*, and its state in any condition is assumed to be determined by its *logic*, i.e., a discrete function of its regulators’ states in that condition. Note that this assumption implies that the relevant conditions are in steady state. In order to model our confidence in the prior knowledge, the logic of a variable X_i is formulated probabilistically as our level of certainty that the variable attains a certain state given the state of its regulatory unit. The uncertainty is modeled by the conditional probability $\theta^i(X_i \mid Pa_i)$. This approach allows us to distinguish between regulatory logics that are known at high level of certainty and those that are more speculative.

The experimental treatment is modeled by fixing the states of each variable that correspond to the environment, and by changing the regulation function priors to reflect the perturbations (e.g., when a gene is knocked out, its level is set to zero under that condition, irrespective of the levels of its regulators).

Data integration - In practice, biological measurements are continuous, and one does not know in advance how to translate them into discrete states. To overcome this, each logical variable X_i is associated with an observed real-valued variable Y_i, and the conditional distribution $\psi^i(Y_i \mid X_i)$ specifies the probability of the variable Y_i to attain a certain observed real value given its state. Hence, $\psi^i(Y_i \mid X_i)$ translates the actual measurements into the discrete model without applying any a-priori discretization to the data. In MetaReg, each ψ is modeled as a mixture of Gaussians.

The complete computational model - Our probabilistic model defines a *Bayesian score,* which evaluates the fit of the model predictions to the data, measured as the log likelihood of the data given the model:

$$\log \Pr(X, Y \mid Model) = \log\left(\frac{1}{Z}\prod_i \theta^i(X_i \mid Pa_i)\cdot\psi^i(Y_i \mid X_i))\right)$$

where Z is a normalization constant. The conditional probabilities θ^i are known from our prior knowledge on the biological system, and ψ are determined by maximizing a likelihood score using an Expectation-Maximization procedure. This model corresponds to a Bayesian network in case of acyclic dependencies, or to a factor graph, in the more general case where the model contains feedback loops.

Computing Model predictions - The *predicted level* is the expected value of a variable X_i given the model and the experimental procedure applied. This is obtained by first computing the posterior states distribution of X_i using a standard probabilistic inference method called Loopy Belief Propagation [19]. This way we obtain a probabilistic average of all its possible system modes. Then, the (continuous) predicted level of X_i is its expectation given θ^i and its states distribution. The comparison of predicted and observed levels (both on the model canvas and in a discrepancy matrix) displays both levels as real values.

Logic refinement

Given a target gene and its candidate regulatory unit, the refinement process searches in the space of discrete regulatory logics in order to achieve a logic with a locally maximum Bayesian score, while fixing the logics of all other variables. Due to an exponential number of possible logics, we apply a greedy heuristic. In case of ties the algorithm chooses randomly among the equally scored improvements. The ψ^i parameters depend strongly on the particular model logics, and thus we re-optimize them using an EM-like procedure during each step of the logical refinement procedure. Note that the refinement process utilizes the Loopy Belief Propagation algorithm, and thus the solution builds on probabilistic averaging of all possible system modes.

5. Discussion

MetaReg provides a framework for modeling and analysis a biological network vis-à-vis high throughput data. A major practical need of molecular biologists today is to generate hypotheses based on network modeling and to iteratively refine the network. MetaReg is designed exactly for this purpose – it allows mathematical modeling of a biological system, interpretation of high throughput data in the context of the prior model, and computational refinement of the model based on the high throughput data. Several other tools with related capabilities, emphasizing visualization or simulations, are being developed, (Table 1). The MetaReg platform is unique in its modeling and refinement capabilities, which fit the needs and workflow of biological investigations. It allows streamlined cycles of probabilistic modeling, laboratory experimentation and systematic refinement.

MetaReg is implemented efficiently, computing predictions and logic refinements within a few seconds for 100 nodes, and within an hour for 6000 nodes (using a network with no more than three regulators per variable, 90% certainty level in all logics, and 100 gene expression profiles). However, the model has practical size limitations: the prediction algorithm run-time increases exponentially with the average number of regulators per variable. Also, for large models with over 300 variables, the automatic layout of the model topology may take several minutes.

MetaReg formalizes the biological system using discrete component states, assuming that the system is in steady state. Clearly these crucial assumptions are a simplification of the biological reality. By making such assumptions, we tried to strike a practical balance between our wish to enable a faithful description of the biological system and the scarcity of accurate knowledge at very high resolution. Indeed, biological processes are inherently temporal, but when the sampling rate (the number and time resolution of experiments) is low relative to the rate of the regulatory mechanisms, we believe that our results here as well as in [7-9] show that the steady state assumption is reasonable.

The accuracy of the prediction and refinement processes may be sensitive to the model size and the certainty in the logics. We have shown previously that the algorithms are highly robust to certainty level on small networks [7]. Indeed, the results shown in the leucine example were obtained using a uniform certainty level of 0.99 for all variables, but we obtained very similar results when using certainty levels of 0.95 and 0.9 (not shown). However, the robustness of our methods to model size and to certainty levels requires further systematic exploration.

A major prerequisite to using MetaReg is formalizing high quality prior knowledge on the pathway of interest. Several efforts to generate databases of curated knowledge on signaling pathway are currently under way (e.g., BioModels [20], Reactome [21] and SPIKE [22]). Thanks to such efforts, it will soon be relatively easy to apply the MetaReg methodology in studying many additional biological systems.

Availability and requirements

Project name: MetaReg
Project home page: http://acgt.cs.tau.ac.il/metareg
Operating system(s): Windows
Programming language: Java for the envelope and C++ for the algorithms.
Other requirements: Java 1.5 or higher
License: free for non-commercial users.
Any restrictions to use by non-academics: License needed.

Acknowledgements

This work was supported in by the EMI-CD project that is funded by the European Commission within its FP6 Programme, under the thematic area “Life sciences, genomics and biotechnology for health”, contract number LSHG-CT-2003-503269. “The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.”

The graphical capabilities of the model layout and related dialogs are deeply based on the implementations of the PIVOT [23] and SPIKE [22] software developed in close collaboration of R. Shamir's group with Y. Shiloh’s group at the School of Medicine, Tel Aviv University. We are especially indebted to Giora Sternberg and Ran Blekhman for their fruitful support. The interactive data matrix display has been developed by Israel Steinfeld for the SIMBA website [24]. We also would like to thank Amos Tanay for helpful discussions and Ewa Szczurek for testing MetaReg and helping to improve it. I. Ulitsky is a fellow of the Edmond J. Safra Bioinformatics Program at Tel-Aviv University.

Author contributions

Igor Ulitsky: developed the tool, performed the analysis and co-wrote the paper
Irit Gat-Viks: conceived the study, developed MetaReg, performed the analysis and co-wrote the paper.
Ron Shamir: conceived and supervised the study and co-wrote the paper.

Table 1 Available tools related to MetaReg.

Tool type	Tools	Description	Relation to MetaReg
Network or model visualization tools	Cytoscape [25] Visant [26] CellDesigner [27] Reviewed in [28]	Tools for constructing visualizations of interaction and regulatory networks. These networks can then be integrated with high-throughput data.	These tools offer powerful visualization aids and other analysis aids, but they do not address regulatory logics and do not offer model evaluation or refinement mechanisms.
Kinetic and continuous modeling tools	Gepasi [29] BioNetS [30] Dynetica [31] PyBioS [32] Reviewed in [33-35]	Tools allowing detailed dynamical modeling with kinetic parameters and differential equations.	These tools can perform detailed model analysis by accurate dynamical simulations, but they cannot discover new mechanisms and rely on detailed mechanistic understanding of the system[1]
Logical modeling tools	BIOCHAM [36] Bionet [37] CellNetAnalyzer [38] GINsim [39]	Tools for modeling regulatory systems using various formalisms, e.g., Boolean, discrete, fuzzy logic etc.	Allow model evaluation through simulations, but are not designed for model evaluation and refinement in accordance with high throughput data.

[1] In several cases these tools have parameter optimization capabilities, but only for the given differential equations included in the model. Hence, these tools lack MetaReg's ability to discover unknown mechanism of reaction and changes in model topology. Furthermore, kinetic and continuous modeling approaches require either detailed mechanistic understanding or known model parameters (e.g., reaction constants), which are commonly unknown, even in well studied systems (e.g., the Leucine biosynthesis pathway studied here).

Figure captions

Fig. 1: Overview of the workflow in MetaReg. **(A)** The available knowledge about the biological system is represented by a mathematical model. **(B)** The model can be manually improved using simulations. **(C)** The model can be integrated with experimental data. For each experiment we modify the model according to the specific treatment and attach the measurements to the model variables. **(D)** MetaReg predicts the variable states based on the experimental treatment and the predictions are visually compared with the measurements. **(E)** The algorithmic engine proposes refinements to the model in order to increase its consistency with the data. The refinement process can be iterated after accepting certain model changes.

Fig. 2: Screenshots of MetaReg core functions. (**A**) The model display. In the main window, the model canvas allows dynamic model layout, model simulation and display of observations (measurements) vs. predictions under a particular experiment for each of the model variables. On the right (top to bottom): satellite view of the current model, variable lookup and variable property viewer. (**B**) Selection of variables from a gene database (NCBI Gene or SGD). (**C**) Formulation of a variable's logic using the wizard. (**D**) The discrepancy matrix, which compares predicted and observed levels for all experiments (columns) and variables (rows). (**E**) A logic suggested by MetaReg's model refinement algorithm. The suggestion can be further edited by the user, and incorporated into the model.

Fig. 3 : MetaReg model canvas view of leucine biosynthesis in yeast during simulation of leucine starvation. The model includes the extracellular stimuli, leucine uptake into the cell by various permeases, the leucine biosynthetic pathway, and its transcriptional regulation by Leu3 and Gcn4. Variable name suffixes indicate variable types: 'm' represents mRNA and 'ap' represents active protein. Arrows indicate the direction of regulation. Arrow types represent either activation (→), repression (—┤) or other (–O) of a variable; for complex logics, the arrow types are an approximation only. The logics of the regulation are not displayed in this view, but are accessible via other windows (**Figure 2C**). The model canvas enables highlighting of different sets of variables. In this snapshot, all the cycles in the model are highlighted in orange. The model is presented here during a simulation of leucine starvation: the values of the extracellular stimuli on variables NH3, amino acids, rapamycin and leucine were fixed to states 2, 2, 0, and 0, respectively. The resulting predicted (simulated) states of all other variables are presented to the left of their nodes.

Fig 4: Comparison of measurements and model predictions on the leucine biosynthesis model. The expression levels (both predicted and observed) are indicated in yellow to blue scale (low to high expression). Discrepancies are indicated in green (observed < predicted) to red (observed > predicted) scale. **(A)** The data collected for the leucine biosynthesis model. Rows correspond to mRNA variables and columns correspond to experiments. Cells are colored according to their observed expression levels. Black cells correspond to cases where the mRNA was not measured. The black strip in the top portion of LEU3m cells indicates that it was perturbed in the respective experiments. **(B)** The discrepancy matrix, highlighting differences between measurements and predictions. Rows correspond to mRNA variables and columns to experiments. Each cell contains two squares colored in expression scale, where the left square indicates observed level and the right indicates the predicted level. The background color intensity indicates the discrepancy between the observed and predicted levels. **(C)** The observed and predicted expression level for nitrogen limitation experiment eight hours after treatment (Gasch et al. 2000, matrices A,B column 20) projected on the model canvas as two colored strips above each variable. The strip right above the variable box represents the predicted level. The strip above it (available only for mRNA variables in this case) represents the observed level.

Fig. 5: Refinement of the leucine biosynthesis model. (A) The refined regulatory logic suggested by MetaReg for LEU2 and LEU9. The regulators of both genes are the transcription factors Gcn4 and Leu3. For each logic, the two columns on the left represent all possible combinations of the regulators' states and the rightmost column is the regulatee's level, colored by an expression scale. Light gray background indicates that the output level predicted for the input combination is not statistically significant. **(B**) Predictions and measurements under specific conditions. MetaReg computes a refined logic based on the regulators' predicted activity level and the observed mRNA level of the regulated gene. As an example, the figure shows the predicted levels of Gcn4 and Leu3 in four conditions along with the measured (top strip) levels of LEU2. The corresponding predicted levels of LEU2 (bottom strip) match the logic suggested by MetaReg for LEU2, as shown in A. **(C)** Discrepancy matrices for LEU2 and LEU9 *before* refinement (LEU9 with a constant level; LEU2 activated by Leu3 only) and *after* refinement (using the logics that appear in A). Clearly, the automatic refinement process reduces the disagreement between the model and the measurements.

References

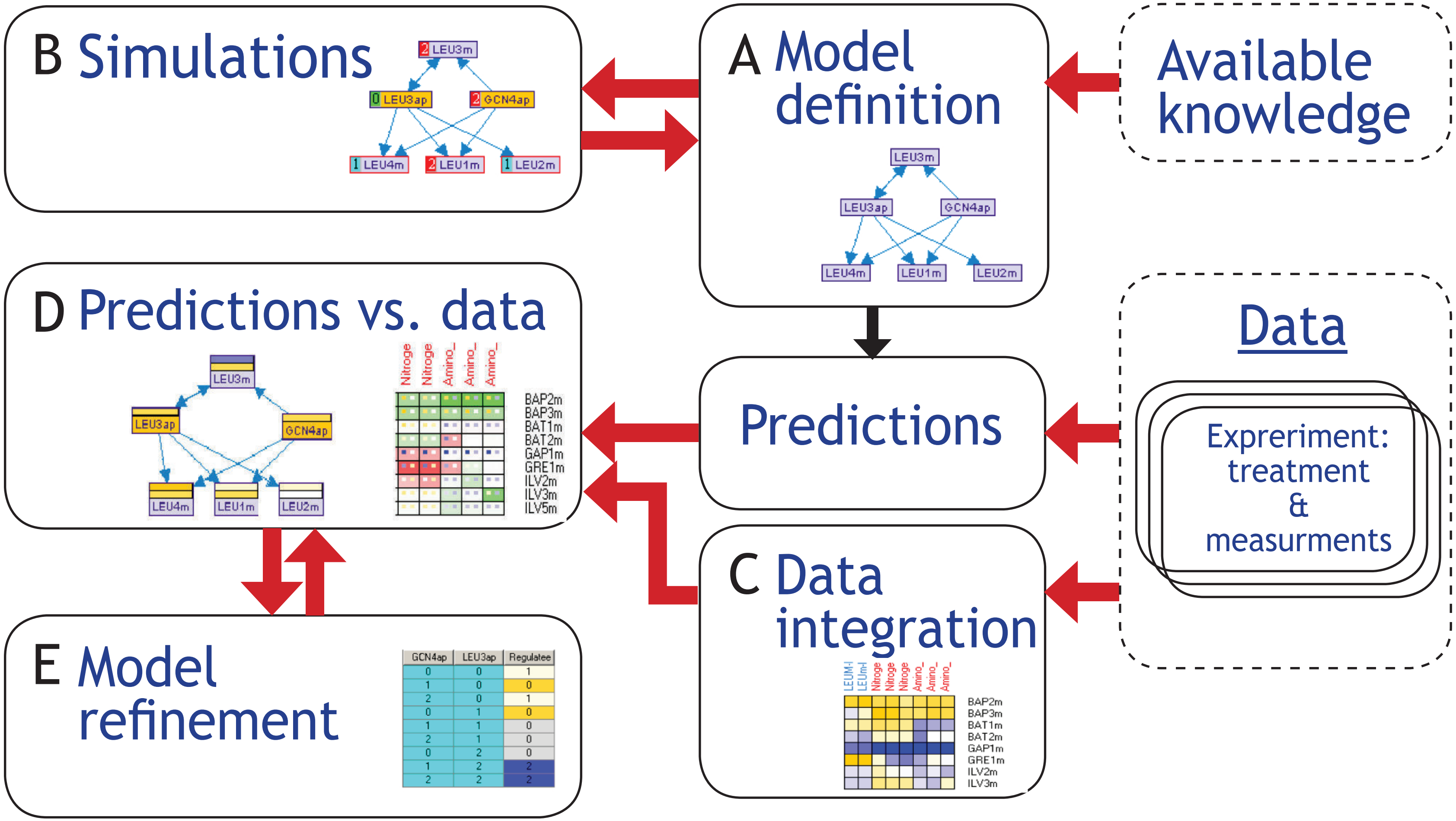
1. Eisen MB, Spellman PT, Brown PO, Botstein D: **Cluster analysis and display of genome-wide expression patterns**. *Proc Natl Acad Sci U S A* 1998, **95**(25):14863-14868.
2. Segal E, Shapira M, Regev A, Pe'er D, Botstein D, Koller D, Friedman N: **Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data**. *Nat Genet* 2003, **34**(2):166-176.
3. Bader GD, Hogue CW: **An automated method for finding molecular complexes in large protein interaction networks**. *BMC Bioinformatics* 2003, **4**:2.
4. Rives AW, Galitski T: **Modular organization of cellular networks**. *Proc Natl Acad Sci U S A* 2003, **100**(3):1128-1133.
5. Klipp E, Nordlander B, Kruger R, Gennemark P, Hohmann S: **Integrative model of the response of yeast to osmotic shock**. *Nat Biotechnol* 2005, **23**(8):975-982.
6. Covert MW, Knight EM, Reed JL, Herrgard MJ, Palsson BO: **Integrating high-throughput and computational data elucidates bacterial networks**. *Nature* 2004, **429**(6987):92-96.
7. Gat-Viks I, Tanay A, Raijman D, Shamir R: **A probabilistic methodology for integrating knowledge and experiments on biological networks**. *J Comput Biol* 2006, **13**(2):165-181.
8. Gat-Viks I, Tanay A, Shamir R: **Modeling and analysis of heterogeneous regulation in biological networks**. *J Comput Biol* 2004, **11**(6):1034-1049.
9. Gat-Viks I, Shamir R: **Refinement and expansion of signaling pathways: the osmotic response network in yeast**. *Genome Res* 2007, **17**(3):358-367.
10. Pearl J: **Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference**: Morgan Kaufmann; 1988.
11. Cherry JM, Adler C, Ball C, Chervitz SA, Dwight SS, Hester ET, Jia Y, Juvik G, Roe T, Schroeder M *et al*: **SGD: Saccharomyces Genome Database**. *Nucleic Acids Res* 1998, **26**(1):73-79.
12. Maglott D, Ostell J, Pruitt KD, Tatusova T: **Entrez Gene: gene-centered information at NCBI**. *Nucleic Acids Res* 2005, **33**(Database issue):D54-58.
13. Kohlhaw GB: **Leucine biosynthesis in fungi: entering metabolism through the back door**. *Microbiol Mol Biol Rev* 2003, **67**(1):1-15, table of contents.
14. Hardwick JS, Kuruvilla FG, Tong JK, Shamji AF, Schreiber SL: **Rapamycin-modulated transcription defines the subset of nutrient-sensitive signaling pathways directly controlled by the Tor proteins**. *Proc Natl Acad Sci U S A* 1999, **96**(26):14866-14870.
15. Natarajan K, Meyer MR, Jackson BM, Slade D, Roberts C, Hinnebusch AG, Marton MJ: **Transcriptional profiling shows that Gcn4p is a master regulator of gene expression during amino acid starvation in yeast**. *Mol Cell Biol* 2001, **21**(13):4347-4368.
16. Boer VM, Daran JM, Almering MJ, de Winde JH, Pronk JT: **Contribution of the Saccharomyces cerevisiae transcriptional regulator Leu3p to physiology and gene expression in nitrogen- and carbon-limited chemostat cultures**. *FEMS Yeast Res* 2005, **5**(10):885-897.
17. Gasch AP, Spellman PT, Kao CM, Carmel-Harel O, Eisen MB, Storz G, Botstein D, Brown PO: **Genomic expression programs in the response of yeast cells to environmental changes**. *Mol Biol Cell* 2000, **11**(12):4241-4257.

18. Rohde JR, Campbell S, Zurita-Martinez SA, Cutler NS, Ashe M, Cardenas ME: **TOR controls transcriptional and translational programs via Sap-Sit4 protein phosphatase signaling effectors**. *Mol Cell Biol* 2004, **24**(19):8332-8341.
19. Kschischang FR, Frey BJ, Loeliger HA: **Factor graphs and the sum-product algorithm**. *IEEE Transactions on Information Theory* 2001, **47**(2):498-519.
20. Le Novטre N, Bornstein B, Broicher A, Courtot M, Donizelli M, Dharuri H, Li L, Sauro H, Schilstra M, Shapiro B: **BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems**. *Nucleic Acids Research*.
21. Vastrik I, D'Eustachio P, Schmidt E, Stein L: **Reactome: a knowledge base of biologic pathways and processes**. *Genome Biol* 2007, **8**(3):R39.
22. Fraser HB, Hirsh AE, Steinmetz LM, Scharfe C, Feldman MW: **Evolutionary rate in the protein interaction network**. *Science* 2002, **296**(5568):750-752.
23. Orlev N, Shamir R, Shiloh Y: **PIVOT: protein interacions visualizatiOn tool**. *Bioinformatics* 2004, **20**(3):424-425.
24. Tanay A, Steinfeld I, Kupiec M, Shamir R: **Integrative analysis of genome-wide experiments in the context of a large high-throughput data compendium**. *Mol Syst Biol* 2005, **1**:2005 0002.
25. Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B, Ideker T: **Cytoscape: a software environment for integrated models of biomolecular interaction networks**. *Genome Res* 2003, **13**(11):2498-2504.
26. Hu Z, Mellor J, Wu J, Yamada T, Holloway D, Delisi C: **VisANT: data-integrating visual framework for biological networks and modules**. *Nucleic Acids Res* 2005, **33**(Web Server issue):W352-357.
27. Kitano H, Funahashi A, Matsuoka Y, Oda K: **Using process diagrams for the graphical representation of biological networks**. *Nat Biotechnol* 2005, **23**(8):961-966.
28. Bell GW, Lewitter F: **Visualizing networks**. *Methods Enzymol* 2006, **411**:408-421.
29. Baigent S: **Software review. Gepasi 3.0**. *Brief Bioinform* 2001, **2**(3):300-302.
30. Adalsteinsson D, McMillen D, Elston TC: **Biochemical Network Stochastic Simulator (BioNetS): software for stochastic modeling of biochemical networks**. *BMC Bioinformatics* 2004, **5**:24.
31. You L, Hoonlor A, Yin J: **Modeling biological systems using Dynetica--a simulator of dynamic networks**. *Bioinformatics* 2003, **19**(3):435-436.
32. Rodriguez-Navarro S, Fischer T, Luo MJ, Antunez O, Brettschneider S, Lechner J, Perez-Ortin JE, Reed R, Hurt E: **Sus1, a functional component of the SAGA histone acetylase complex and the nuclear pore-associated mRNA export machinery**. *Cell* 2004, **116**(1):75-86.
33. Alves R, Antunes F, Salvador A: **Tools for kinetic modeling of biochemical networks**. *Nat Biotechnol* 2006, **24**(6):667-672.
34. Aldridge BB, Burke JM, Lauffenburger DA, Sorger PK: **Physicochemical modelling of cell signalling pathways**. *Nat Cell Biol* 2006, **8**(11):1195-1203.
35. Price ND, Reed JL, Palsson BO: **Genome-scale models of microbial cells: evaluating the consequences of constraints**. *Nat Rev Microbiol* 2004, **2**(11):886-897.
36. Calzone L, Fages F, Soliman S: **BIOCHAM: an environment for modeling biological systems and formalizing experimental knowledge**. *Bioinformatics* 2006, **22**(14):1805-1807.

37. Bosl WJ: **Systems biology by the rules: hybrid intelligent systems for pathway modeling and discovery**. *BMC Syst Biol* 2007, **1**:13.

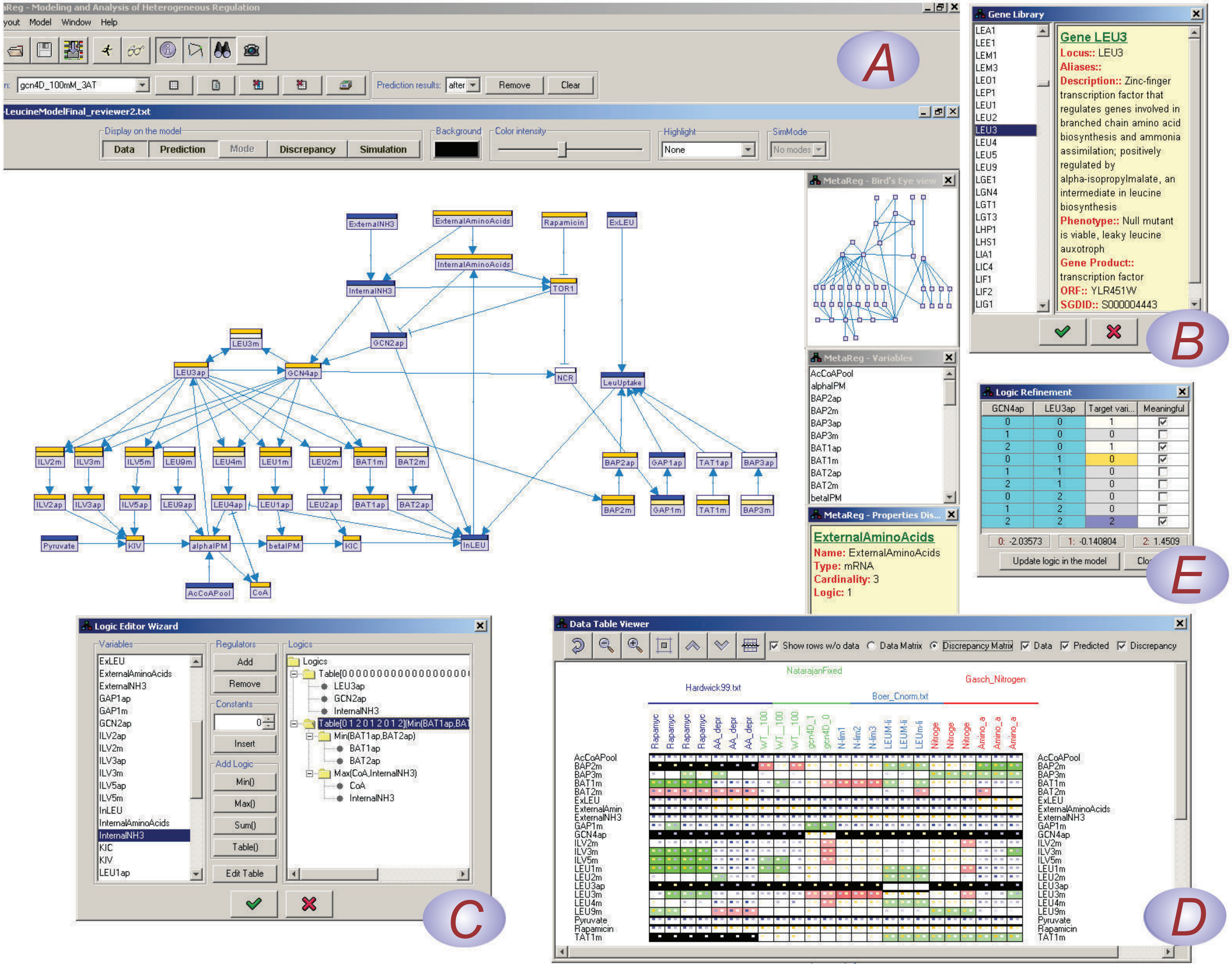
38. Klamt S, Saez-Rodriguez J, Gilles ED: **Structural and functional analysis of cellular networks with CellNetAnalyzer**. *BMC Syst Biol* 2007, **1**:2.

39. Gonzalez AG, Naldi A, Sanchez L, Thieffry D, Chaouiya C: **GINsim: a software suite for the qualitative modelling, simulation and analysis of regulatory networks**. *Biosystems* 2006, **84**(2):91-100.



GCN4ap	LEU3ap	Regulatee
0	0	1
1	0	0
2	0	1
0	1	0
1	1	0
2	1	0
0	2	0
1	2	2
2	2	2

Figure 1



Gene LEU3
Locus:: LEU3
Aliases::
Description:: Zinc-finger transcription factor that regulates genes involved in branched chain amino acid biosynthesis and ammonia assimilation; positively regulated by alpha-isopropylmalate, an intermediate in leucine biosynthesis
Phenotype:: Null mutant is viable, leaky leucine auxotroph
Gene Product:: transcription factor
ORF:: YLR451W
SGDID:: S000004443

ExternalAminoAcids
Name: ExternalAminoAcids
Type: mRNA
Cardinality: 3
Logic: 1

GCN4ap	LEU3ap	Target vari...	Meaningful
0	0	1	☑
1	0	0	□
2	0	1	☑
0	1	0	☑
1	1	0	□
2	1	0	□
0	2	0	□
1	2	0	□
2	2	2	☑

0: -2.03573 1: -0.140804 2: 1.4509

Update logic in the model Clo

Logics
Table[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LEU3ap
GCN2ap
InternalNH3
Table[0 1 2 0 1 2 0 1 2][Min(BAT1ap,BA
Min(BAT1ap,BAT2ap)
BAT1ap
BAT2ap
Max(CoA,InternalNH3)
CoA
InternalNH3

☑ Show rows w/o data ○ Data Matrix ● Discrepancy Matrix ☑ Data ☑ Predicted ☑ Discrepancy

Figure 2

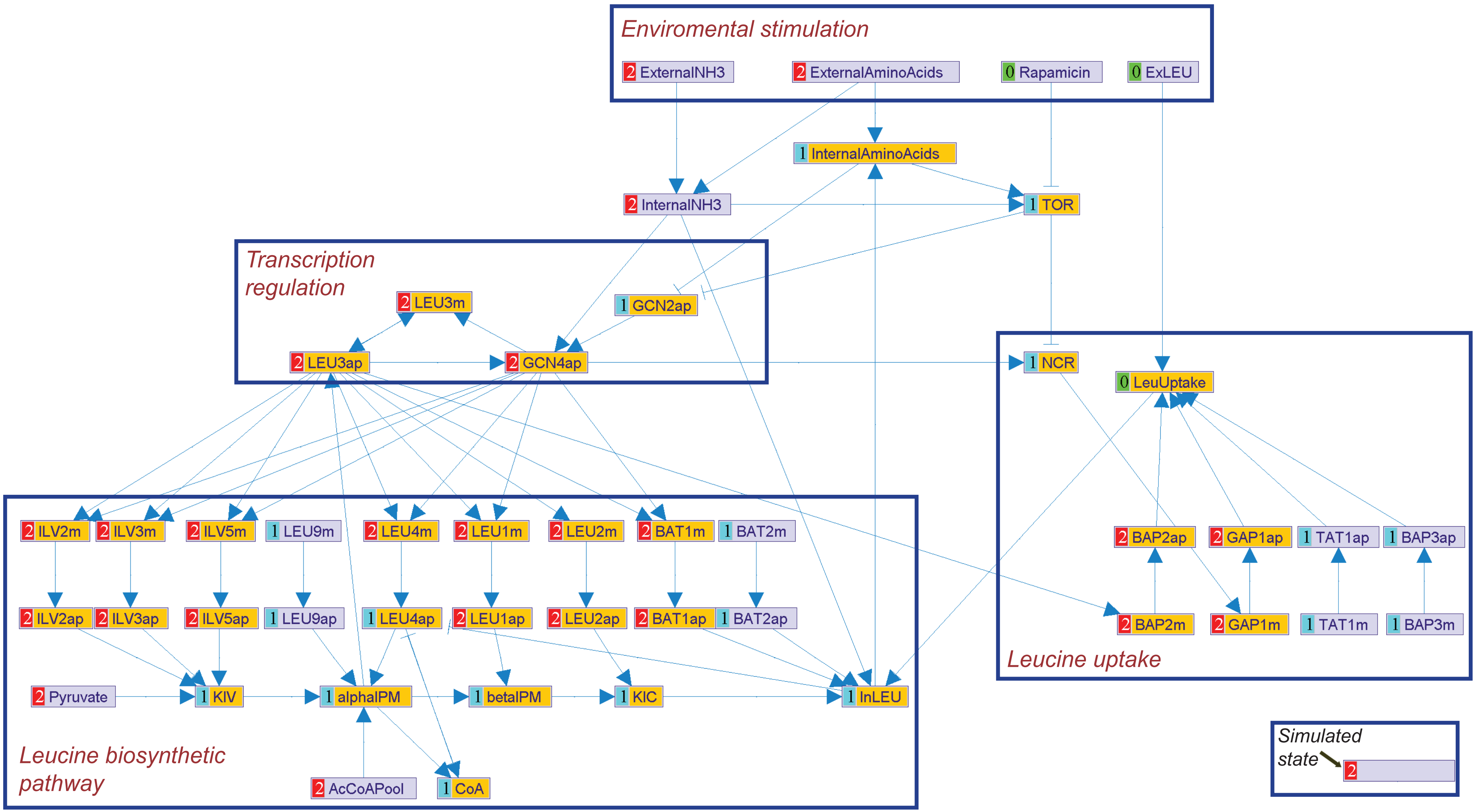
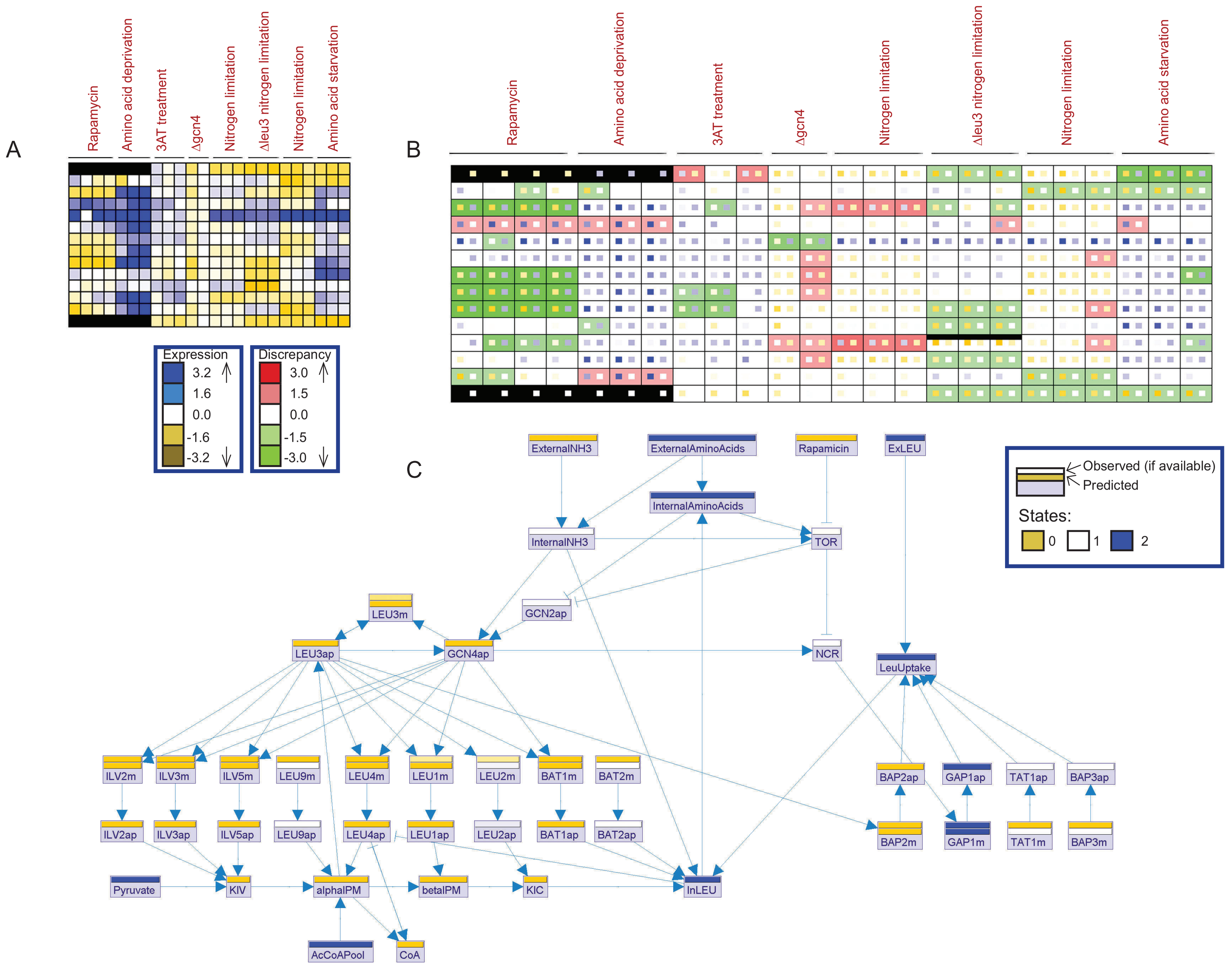
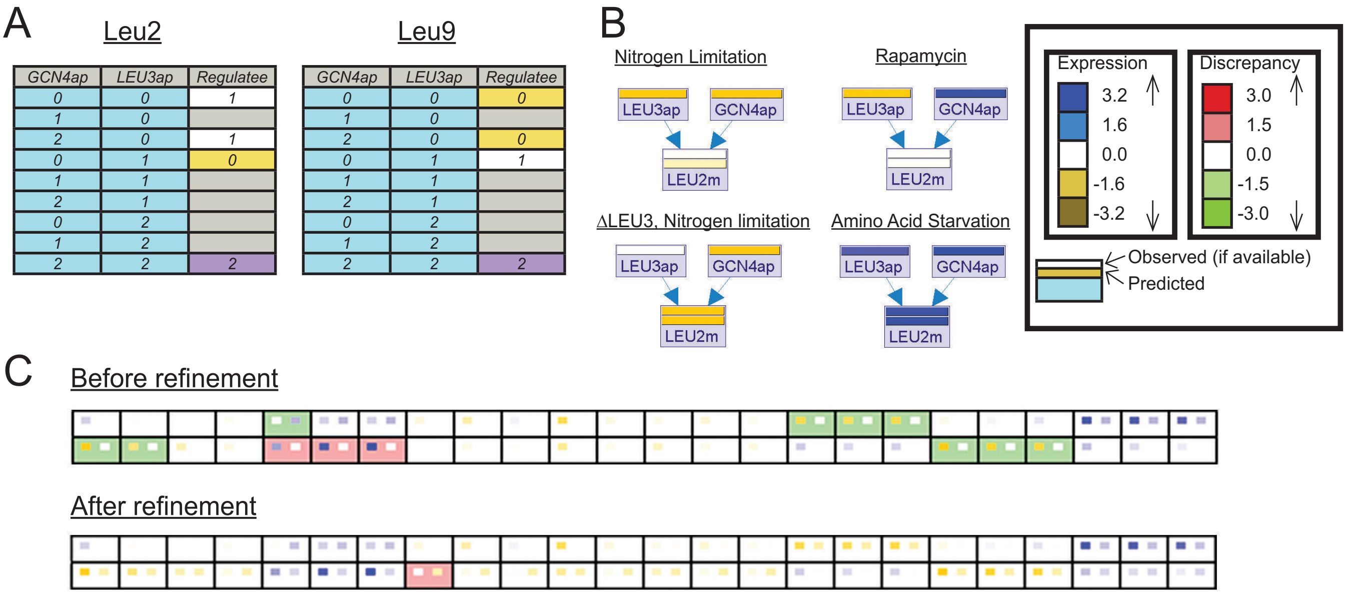

Enviromental stimulation
2 ExternalNH3
2 ExternalAminoAcids
0 Rapamicin
0 ExLEU
1 InternalAminoAcids
2 InternalNH3
1 TOR
Transcription regulation
2 LEU3m
1 GCN2ap
2 LEU3ap
2 GCN4ap
1 NCR
0 LeuUptake
2 ILV2m
2 ILV3m
2 ILV5m
1 LEU9m
2 LEU4m
2 LEU1m
2 LEU2m
2 BAT1m
1 BAT2m
2 ILV2ap
2 ILV3ap
2 ILV5ap
1 LEU9ap
1 LEU4ap
2 LEU1ap
2 LEU2ap
2 BAT1ap
1 BAT2ap
2 Pyruvate
1 KIV
1 alphaIPM
1 betaIPM
1 KIC
1 InLEU
Leucine biosynthetic pathway
2 AcCoAPool
1 CoA
2 BAP2ap
2 GAP1ap
1 TAT1ap
1 BAP3ap
2 BAP2m
2 GAP1m
1 TAT1m
1 BAP3m
Leucine uptake
Simulated state
2

Figure 3



A
Rapamycin
Amino acid deprivation
3AT treatment
Δgcn4
Nitrogen limitation
Δleu3 nitrogen limitation
Nitrogen limitation
Amino acid starvation
Expression
3.2
1.6
0.0
-1.6
-3.2
Discrepancy
3.0
1.5
0.0
-1.5
-3.0
B
Rapamycin
Amino acid deprivation
3AT treatment
Δgcn4
Nitrogen limitation
Δleu3 nitrogen limitation
Nitrogen limitation
Amino acid starvation
C
ExternalNH3
ExternalAminoAcids
Rapamicin
ExLEU
InternalAminoAcids
InternalNH3
TOR
GCN2ap
LEU3m
LEU3ap
GCN4ap
NCR
LeuUptake
ILV2m
ILV3m
ILV5m
LEU9m
LEU4m
LEU1m
LEU2m
BAT1m
BAT2m
BAP2ap
GAP1ap
TAT1ap
BAP3ap
ILV2ap
ILV3ap
ILV5ap
LEU9ap
LEU4ap
LEU1ap
LEU2ap
BAT1ap
BAT2ap
BAP2m
GAP1m
TAT1m
BAP3m
Pyruvate
KIV
alphaIPM
betaIPM
KIC
InLEU
AcCoAPool
CoA
Observed (if available)
Predicted
States:
0
1
2

Figure 4



Leu2

GCN4ap	LEU3ap	Regulatee
0	0	1
1	0	
2	0	1
0	1	0
1	1	
2	1	
0	2	
1	2	
2	2	2

Leu9

GCN4ap	LEU3ap	Regulatee
0	0	0
1	0	
2	0	0
0	1	1
1	1	
2	1	
0	2	
1	2	
2	2	2

Figure 5

Additional files provided with this submission:

Additional file 1: supplementa.doc, 39K
http://genomebiology.com/imedia/8101480151760707/supp1.doc