Lawrence Berkeley National Laboratory

LBL Publications

Title

Protocol for condition-dependent metabolite yield prediction using the TRIMER pipeline

Permalink

https://escholarship.org/uc/item/8f87t1zp

Journal

STAR Protocols, 3(1)

ISSN

2666-1667

Authors

Niu, Puhua

Soto, Maria J

Yoon, Byung-Jun

et al.

Publication Date

2022-03-01

DOI

10.1016/j.xpro.2022.101184

Peer reviewed

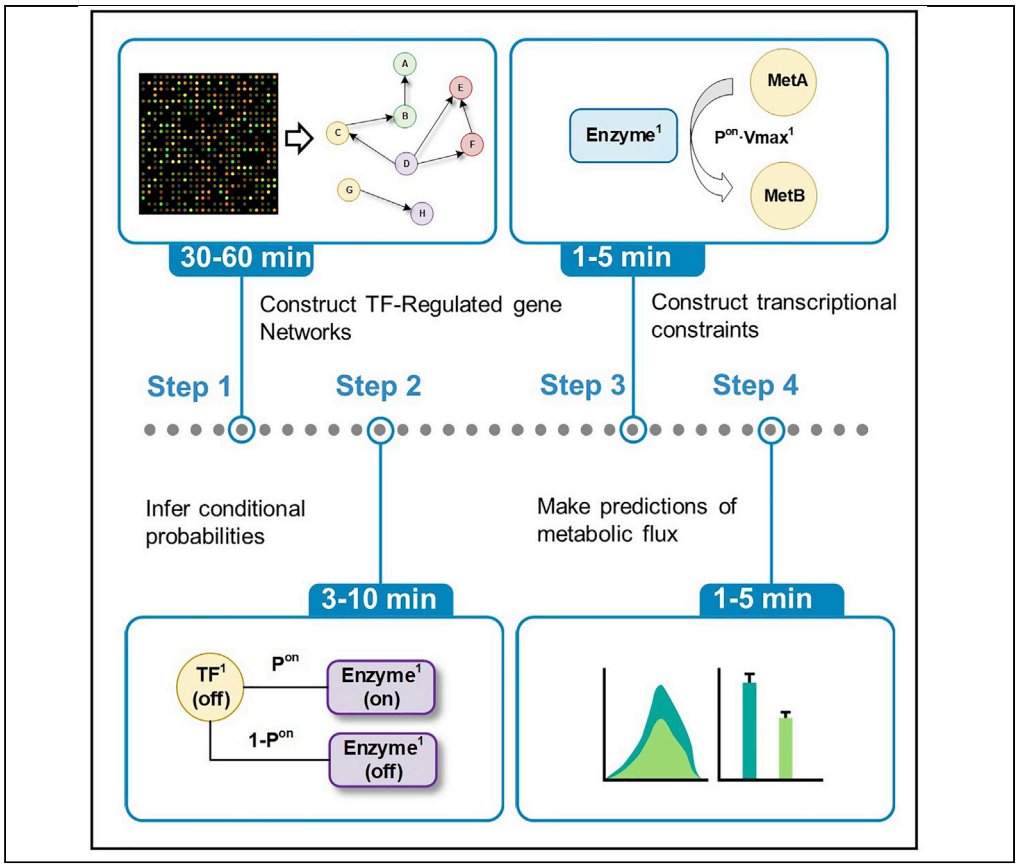
eScholarship.org Powered by the California Digital Library
University of California

STAR Protocols

CellPress
OPEN ACCESS

Protocol

Protocol for condition-dependent metabolite yield prediction using the TRIMER pipeline



This protocol explains the pipeline for condition-dependent metabolite yield prediction using Transcription Regulation Integrated with MEtabolic Regulation (TRIMER). TRIMER targets metabolic engineering applications via a hybrid model integrating transcription factor (TF)-gene regulatory network (TRN) with a Bayesian network (BN) inferred from transcriptomic expression data to effectively regulate metabolic reactions. For *E. coli* and yeast, TRIMER achieves reliable knockout phenotype and flux predictions from the deletion of one or more TFs at the genome scale.

Puhua Niu, Maria J. Soto, Byung-Jun Yoon, Edward R. Dougherty, Francis J. Alexander, Ian Blaby, Xiaoning Qian

ikblaby@lbl.gov (I.B.)
xqian@ece.tamu.edu (X.Q.)

Highlights

TRIMER is a package for transcription-regulated metabolic predictions

Global dependency modeling by Bayesian network enables condition-dependent prediction

We present the step-by-step TRIMER implementation for metabolic engineering

We demonstrate the analyses for *E. coli* and yeast mutants

Niu et al., STAR Protocols 3, 101184
March 18, 2022 © 2022 The Author(s).
https://doi.org/10.1016/j.xpro.2022.101184

STAR Protocols

CellPress
OPEN ACCESS

Protocol

Protocol for condition-dependent metabolite yield prediction using the TRIMER pipeline

Puhua Niu,[1,5] Maria J. Soto,[2] Byung-Jun Yoon,[1,3] Edward R. Dougherty,[1] Francis J. Alexander,[3] Ian Blaby,[2,4,*] and Xiaoning Qian[1,3,6,*]

[1]Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843, USA

[2]US Department of Energy Joint Genome Institute, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

[3]Computational Science Initiative, Brookhaven National Laboratory, Upton, NY 11973, USA

[4]Environmental Genomics and Systems Biology Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

[5]Technical contact

[6]Lead contact

*Correspondence: ikblaby@lbl.gov (I.B.), xqian@ece.tamu.edu (X.Q.)
https://doi.org/10.1016/j.xpro.2022.101184

SUMMARY

This protocol explains the pipeline for condition-dependent metabolite yield prediction using Transcription Regulation Integrated with MEtabolic Regulation (TRIMER). TRIMER targets metabolic engineering applications via a hybrid model integrating transcription factor (TF)-gene regulatory network (TRN) with a Bayesian network (BN) inferred from transcriptomic expression data to effectively regulate metabolic reactions. For *E. coli* and yeast, TRIMER achieves reliable knockout phenotype and flux predictions from the deletion of one or more TFs at the genome scale.

For complete details on the use and execution of this protocol, please refer to Niu et al. (2021).

BEFORE YOU BEGIN

A metabolic network model is a mathematical abstraction of all known biochemical reactions and transmembrane transporters that occur within a living system for *in silico* computational predictions of metabolic dynamic behavior (Yurkovich and Palsson, 2015). Such networks provide a comprehensive view of all characterized metabolic processes by quantifying their metabolic reaction fluxes. Many methods for constraint-based network reconstruction and analysis can be used to simulate reaction fluxes through metabolic networks at the whole-genome scale (Bordbar et al., 2014). Among these, flux balance analysis (FBA) is a well-known technique that adopts linear programming to predict steady-state fluxes. For more accurate and robust prediction of target metabolic behavior under different conditions or contexts (e.g., for mutant strains due to gene deletions in metabolic engineering), these metabolic network models can also be integrated with a set of genetic regulatory rules, which can be modeled as a transcriptional regulatory network involving transcription factors (TFs) that may regulate metabolic reactions. Transcriptional regulation is often integrated via "transcriptional regulatory constraints" with various heuristics in metabolic network models for flux predictions (Covert and Palsson, 2003; Shlomi et al., 2007; Covert et al., 2008).

With the advent of high-throughput profiling technologies, genome-scale gene expression profiles can be easily obtained to help infer genetic regulatory rules. The recently developed hybrid model, Transcription Regulation Integrated with MEtabolic Regulation (TRIMER) (Niu et al., 2021), motivated by PROM (Chandrasekaran and Price, 2010), employs a Bayesian Network inferred from the gene expression data, built with the prior knowledge on regulatory relationships between TFs

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

and genes to probabilistically model TF-gene regulatory rules. The resulting BN can be used to infer the joint probabilities in more general forms of Pr(gene(s)|TF(s)) than the marginal probabilities in forms of Pr(gene|TF) in PROM. These probabilities can subsequently be used to adjust the corresponding reaction flux constraints in the FBA formulation. This flexible probabilistic modeling of TF-gene regulatory relations enables modeling of the global effect of multiple TFs over their regulons, thereby better predicting the metabolic behaviors under different conditions, particularly when the focus is to predict fluxes with different TF Knockout (KO) strategies.

This protocol provides step-by-step explanations of applying TRIMER that integrates both TF-gene regulations and metabolic reaction regulations for *in silico* condition-dependent metabolic behavior modeling targeted for tasks such as knockout phenotype and knockout flux prediction. Specifically, we focus on analyzing genome-scale metabolic network models (e.g., IAF1260 for *E. coli* or IMM904 for *S. cerevisiae*) to predict metabolic fluxes and phenotypes of TF knockout strains for *E. coli* and *yeast*, to help identify knockout strains that overproduce the desired metabolites. The developed TRIMER pipeline can be extended for other organisms when both the corresponding metabolic reaction network model and transcriptomic expression data are available.

The main steps of the TRIMER pipeline comprise (1) learning a Bayesian network from gene expression data, (2) constructing flux constraints in predicting formulation, and (3) predicting metabolic fluxes. We will first present the necessary preparation steps and data requirements before applying TRIMER. Following this, we describe the critical steps for using TRIMER to predict metabolic fluxes under different TF KO conditions. Before introducing detailed implementation steps, we first list the corresponding input data, intermediate output files at each step, and final output results for TRIMER, as listed in Table 1, with corresponding examples in the TRIMER GitHub repository.

Installation of dependent software

Timing: 30–60 min

Table 1. Examples of data inputs and outputs for TRIMER

Inputs	
iAF1260.mat	Metabolic model
Expressionname.mat	gene names
Expressionid.mat	ordered locus IDs of corresponding gene names
Regulator.mat	transcriptional factors
Target.mat	metabolic genes
Regulatornum.mat	indices of transcriptional factors
targetnum.mat	indices of metabolic genes
expression.mat	gene expression data, from microarray as in our examples
ko_tf.mat	transcriptional factors of interest
growth_pos.mat	Indices of metabolic reactions of interest
Outputs	
tegulator_f.mat	transcriptional factors filtered by KS test
target_f.mat	metabolic genes filtered by KS test
data.mat	normalized and binarized gene expression data
bn_learn.bif	learned Bayesian network
rxn_affected_ko.mat	lists of affected metabolic reactions for considered conditions
rxn_prob_ko.mat	lists of regulatory values for affected metabolic reactions
lb_est.mat	lists of new lower bounds for affected reactions
ub_est.mat	lists of new upper bounds for affected reactions
vmax.mat	minimum/maximum fluxes estimated for each metabolic reactions
f.mat	flux solutions of reactions of interest returned from solvers.
v.mat	flux solutions of all the reactions returned from solvers
status.mat	status of solvers

CellPress
OPEN ACCESS

TRIMER package requires both MATLAB and R environments (for specific detailed requirements please refer to the materials and equipment section below). When writing this protocol, we have tested TRIMER with MATLAB 2016 and R version 4.0.3 in the Windows 10 Operating System. In TRIMER, mathematical programming problems are constructed and solved to make predictions about metabolic behaviors. Currently, TRIMER supports two solvers, CPLEX and GLPK, designed for large-scale linear programming (LP), mixed integer programming (MIP), and other optimization formulations. TRIMER also utilizes the bnlearn (Nagarajan et al., 2013) package, an R package for learning the graphical structures of Bayesian networks (BNs), estimating model parameters, and performing inference based on the learned network models.

1. Install CPLEX and connect it to MATLAB
   a. Download CPLEX via https://www.ibm.com/analytics/cplex-optimizer, and install the package.
   b. Add the CPLEX directory to your MATLAB path:

```
>addpath yourCOShome\cplex\matlab\x64_win64
```

2. (Optional) Install and configure GLPK via either of the two ways described below:
   a. Install GLPKMEX, a MATLAB MEX interface of GLPK . Instructions of installation are available at https://github.com/RoyiAvital/GLPKMEX
   b. Install COBRA toolbox, which provides a MATLAB MEX interface of GLPK
      i. Installation instructions of COBRA can be accessed via this link: https://opencobra.github.io/cobratoolbox/stable/
      ii. Add the GLPK directory to your MATLAB path:

```
>addpath yourCOShome\cobratoolbox\external\base

\solvers\glpkmex.
```

3. Install bnlearn in the R environment. Instructions can be accessed via this link: https://www.bnlearn.com/

⚠ CRITICAL: Only approximate inference methods are available in bnlearn. Install the gRain package first if exact inference over Bayesian networks is desired. Instructions about installing gRain are accessible at https://people.math.aau.dk/~sorenh/software/gR/ .

Preparation of metabolic model

Timing: 20–40 min

Any organism with available gene expression data, a TF-gene interaction list and a metabolic model in the standardized data structure can be analyzed by TRIMER. For metabolic models, TRIMER requires them to be a MATLAB structure array and organized in the same way as required by the COBRA Toolbox, which is a popular module for constraint-based reconstruction and analysis of metabolic networks in MATLAB (Heirendt et al., 2019) (a typical metabolic model is shown in Figure 1).

4. While many metabolic model databases are publicly available online, the default choice in TRIMER is the Biochemical Genetic and Genomic (BiGG) knowledge-base where metabolic models in the COBRA format are provided. The database is accessible at: http://bigg.ucsd.edu/models/.

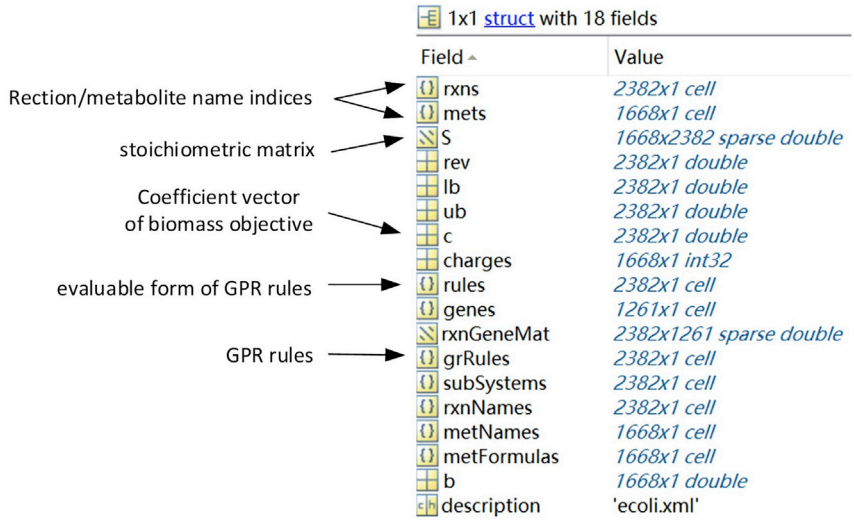


Figure 1. A MATLAB data structure representing the IAF1260 metabolic model for *E. coli*

Before using the metabolic models, users are suggested to run the function `cobra_to_trimer.m`, which checks the contents of COBRA model and converts it to the appropriate form for the CPLEX or GLPK solver. [Troubleshooting 1]

a. In TRIMER, an exemplary model for *E.coli* is provided in the directory:

```
>TRIMER\source_data\Ecoli_model_EcoMac.mat
```

b. Field `.grRules` and `.rules` in the MATLAB array shown in Figure 1 correspond to the same set of GPR (Gene-Protein-Reaction) rules, written in two different forms. Illustrative examples are shown in Figure 2. Only GPR rules written in the way of the right column in Figure 2 can be processed by TRIMER. In field `.rules`, gene names are replaced by indices in the form of x(number), where the numbers represent the ordering of gene names contained in field `.genes`. In addition, logical operators are replaced by recognizable symbols in MATLAB. In case field `.rules` is missing, function `cobra_to_trimer.m` will try to create it based on field `.genes` and `.grRules`.

5. There are also many public databases of gene expression data and annotated TF-gene pairs.

When needed, interactions obtained from inference methods, such as GENIE3 (Huynh-Thu et al., 2010), TIGRESS (Haury et al., 2012), or Inferelator (Bonneau et al., 2006), can also be used to extend the interaction list. For this protocol, the authors use EcoMAC datasets (Carrera et al., 2014) at https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4299492/. The TF-gene pairs in EcoMAC are obtained from the RegulonDBv8.1 database at http://regulondb.ccg.unam.mx/. Gene expression should be normalized and binarized, about which more details can be found in the step-by-step method details section. In TRIMER, exemplary binarized gene expression and TF-gene list can be found in the following two file paths:

```
>TRIMER\source_data\Ecoli_model_EcoMac.mat,

>TRIMER\source_data\EcoMac_data_binary.mat.
```

	model.grRules
	1
10	(b2215) or (b0241) or (b1377) or (b0929)
11	
12	(b4035) and (b4032) and (b4033) and (b4034)
13	(b4036)
14	(b2215) or (b0241) or (b1377) or (b0929)
15	(b2215) or (b0241) or (b1377) or (b0929)
16	(b2215) or (b0241) or (b1377) or (b0929)
17	(b2215) or (b0241) or (b1377) or (b0929)
18	
19	(b2215) or (b0241) or (b1377) or (b0929)
20	(b4213)
21	(b4213)
22	(b4213)

	model.rules
	1
10	(x(612)) | (x(96)) | (x(415)) | (x(309))
11	
12	(x(1153)) & (x(1150)) & (x(1151)) & (x(1152))
13	(x(1154))
14	(x(612)) | (x(96)) | (x(415)) | (x(309))
15	(x(612)) | (x(96)) | (x(415)) | (x(309))
16	(x(612)) | (x(96)) | (x(415)) | (x(309))
17	(x(612)) | (x(96)) | (x(415)) | (x(309))
18	
19	(x(612)) | (x(96)) | (x(415)) | (x(309))
20	(x(1206))
21	(x(1206))
22	(x(1206))

Figure 2. Examples of GPR rules contained in the *E. coli* IAF1260 model in two forms

6. Constructing a TIGER-TRIMER (Jensen et al., 2011) hybrid model requires a Boolean regulatory network, a demo model is provided in the path:

```
>TRIMER\source_data\boolean.mat.
```

The model is converted from the Boolean network model: iMC1010. The conversion code is in the path:

```
>TRIMER\boolean.m.
```

The Boolean model iMC1010 is accessible at this link: https://systemsbiology.ucsd.edu/InSilicoOrganisms/Ecoli/EcoliRegulations.

Configure mathematical programming solver

Timing: 1–3 min

A mathematical programming solver should be specified and configured before metabolic analysis. Following TIGER (Jensen et al., 2011), TRIMER builds a customized MATLAB CMPI (Common Mathematical Programming Interface) for metabolic flux prediction based on the data structure detailed above. This CMPI defines a consistent and friendly interface for mathematical programming solvers, including GLPK and CPLEX for now.

7. Demo code for setting up CPLEX by CMPI is shown below. Users can resort to the MATLAB documents of CMPI for more details about its usage.
8. Demo code for how to set up the solvers is shown below:

```
>options.Display='off';options.MaxTime=100;

>cmpi.set_solver('cplex');

>cmpi.set_option(options);
```

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Raw gene expression data	(Carrera et al., 2014)	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4299492/
Gene interaction list database	RegulonDB v8.1	http://regulondb.ccg.unam.mx/
Other		
Dell Inspiron 7370 Desktop with Windows 10 OS	Dell	NA
Software and algorithms		
MATLAB	MathWorks	https://www.mathworks.com/products/matlab.html
R	The Comprehensive R Archive Network	https://cran.r-project.org/
TRIMER	(Niu et al., 2021)	https://doi.org/10.5281/zenodo.5880036
Bnlearn	(Nagarajan et al., 2013)	https://www.bnlearn.com
Dplyr	The Comprehensive R Archive Network	https://cran.r-project.org/web/packages/dplyr/
Parallel	Part of R	NA
purr	The Comprehensive R Archive Network	https://cran.r-project.org/web/packages/purrr/index.html
R.matlab	The Comprehensive R Archive Network	https://cran.r-project.org/web/packages/R.matlab/index.html
binaryLogic	The Comprehensive R Archive Network	https://cran.r-project.org/web/packages/binaryLogic/index.html
gRain	The Comprehensive R Archive Network	https://cran.r-project.org/web/packages/gRain/index.html
GLPKMEX	GitHub repository	https://github.com/blegat/glpkmex

MATERIALS AND EQUIPMENT

- Hardware and operating systems: The authors used a Dell Inspiron 7370 computer with the Microsoft Windows 10 operating system. This computer has a 500 GB hard drive, an Intel Core i5-8250U CPU @ 1.60 GHz and 8 GB RAM. However, any reasonably up-to-date computer may be used to run all the code and any operating system can be used - Windows, Mac OS, or Unix/Linux.
- R software and required packages: The required packages should be installed for the first time when you run TRIMER. The authors used R (v4.0.3) and the following packages at the indicated versions when writing this protocol:
  - bnlearn (v4.6.1)
  - dplyr (v1.0.6)
  - parallel (v4.0.3)
  - purrr (v0.3.4)
  - R.matlab (v3.6.2)
  - binaryLogic (0.3.9)
  - gRain (1.3.6)
- MATLAB software and required toolboxes: The authors used MATLAB 2016b and the following packages at the indicated versions when writing this protocol:
  - COBRA (v2020)
  - GLPKMEX (v4.62)
- Mathematical programming solver software: The authors used CPLEX version 12.8.0.

STEP-BY-STEP METHOD DETAILS

In this section, a comprehensive step-by-step protocol is laid out for applying TRIMER to a typical metabolic model of *E.coli* iAF1260 based on the EcoMac dataset. Interconnections of the main

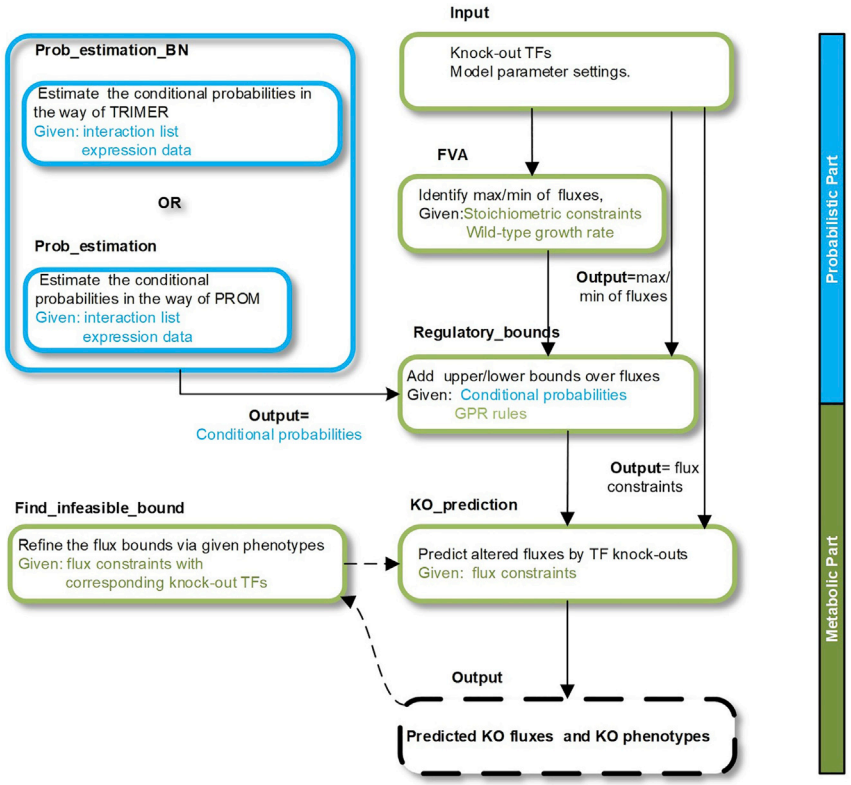


Figure 3. TRIMER workflow

A workflow summarizing the interconnections between the main computational modules in TRIMER (Modified based on two figures in the original publication (Niu et al., 2021), which is an open access article under the CC BY-NC-ND license: http://creativecommons.org/licenses/by-nc-nd/4.0/).

computational modules of TRIMER are shown in Figure 3. All steps described are case-specific, but they can easily be adapted to any gene expression and TF-gene interaction dataset or metabolic model that the users wish to analyze. All the implemented MATLAB functions in TRIMER are well documented. Users can resort to MATLAB's help command for more details about function usages. Demo MATLAB code illustrating the complete procedure of applying TRIMER for biomass and indole flux prediction with TF-knockouts are accessible at the following paths:

```
>TRIMER\flux_biomass.m,

>TRIMER\flux_indole.m.
```

Construct TF-regulated gene network (TRN)

Timing: 30–60 min

We call the set of TF-gene regulations as TF-Regulated gene Network (TRN). In TRIMER, TRN is modeled by a Bayesian network, which is trained from pre-binarized data since TRN concerns 'on\off' states of TFs and genes. Step 2 below can be skipped if users just want to model TRN in the way of PROM (The TRN in PROM is just the filtered TF-gene list).

1. (Under MATLAB environment) Gene expression data and TF-gene interaction list should be preprocessed before being used to learn a Bayesian network. Running the function `data_preprocessing.m` in TRIMER, the following steps will be completed:

a. Quantile normalizes the raw expression data.
b. Binarize the normalized data given a threshold. When writing this protocol, the authors chose the threshold to be 0.33 as suggested in the original PROM paper. In practice, the range of threshold is determined by evaluating PROM's performance. As PROM is computationally efficient and supported in TRIMER, the best threshold range for PROM can be chosen as the threshold for TRIMER.
c. Perform Kolmogorov-Smirnov (KS) test over expression data for each TF-gene pair and prune out the ones that are tested to be independent.
d. Here is an example of using `data_preprocessing.m`:

```
    >%expression ->microarray data

    >%regulator,targets ->TF-gene pair list

    >%expressionid -> list of TF/genes name

    >[expression_b,regulator_f,targets_f]=data_preprocessin
g(expression,expressionid,regulator,targets,'thresh_value',0.33);
```

⚠ CRITICAL: The binarization threshold is a very important hyperparameter, affecting TF-gene pair selection and thereafter Bayesian network modeling of TRN.

2. (Under R environment) In TRIMER, bnlearn is used to learn Bayesian networks based on binary expression data and TF-gene interaction lists. Network learning is comprised of two steps: structure learning and parameter fitting:
   a. Finding the global optimal structure by checking all possible directed acyclic graphs (DAGs) is computationally expensive as the cost grows exponentially with the number of nodes. Two structure learning strategies are implemented in TRIMER: Chow-Liu tree algorithm and Tabu search.
      i. Chow-Liu tree algorithm gives the optimal tree structure.
      ii. Tabu search is a greedy searching method to learn general network structure. To further reduce the computation cost, TRIMER requires TF-gene pair as the searching space of Tabu search.
   b. Demo code for learning a Bayesian network using the EcoMac dataset is provided at the following path:

```
>TRIMER\BN_Module\bnlearn.R.
```

   c. The learned network is saved in the .bif format and will be used for further metabolic analysis in MATLAB. More details about .bif format can be found in the documents provided by the bnlearn package. A demo model is provided at the following path:

```
>TRIMER\BN_Module\learned_network.bif
```

Infer conditional probabilities

Timing: 3–10 min

Once we have learned a BN, we can make inference of all the relevant conditional probabilities which are in the form of Pr(gene(s)|TF(s)) and identified based on the TF-gene interaction list for one or

multiple TFs. These probabilities are then used to compute the regulatory values used for building new flux constraints in the metabolic reaction models.

3. Function `prob_estimation_R.m` allow two ways for computing regulatory values based on conditional probabilities inferred from BNs (Please refer to the TRIMER paper, , for more detail). Given a set of conditions where TF(s) are knocked out, this function returns the indices of reactions affected and the corresponding regulatory values under each KO condition considered. [Troubleshooting 2]
   a. Users can choose either of the two ways by setting the argument `'method'` to be `'CN'` or `'BN'`, corresponding to the one adopted in PROM and the new one supported by TRIMER. It can happen that for a reaction, many genes controlling this reaction are affected simultaneously. This may cause computation problems when using the second way for probability inference. [Troubleshooting 3]
   b. The default choice of performing inference over Bayesian networks is an approximate inference method called logic sampling (please resort to bnlearn documents for more details). For exact inference, package gRain (Højsgaard, 2012) is required besides bnlearn. It should be pointed out that exact inference may be computationally infeasible if the scale of Bayesian network model is large (e.g., hundreds of nodes). [Troubleshooting 4]
   c. TRIMER relies on the annotated TF-gene pairs (also called TF-target interaction list) to determine the affected genes when one TF or multiple TFs are knocked out. Here the TF-gene pair list filtered by KS test during step 1 is utilized again.
   d. Here is the demo code for using `prob_estimation.m`. (exemplary results are shown in Figures 4 and 5):

```
    >%R_path->the directory of R software installed in
    your computer
    >%R_path->the directory of R functions of TRIMER
    package
    >%R_model_path->the path of a BN model saved in .bif
    format.
    >%ko_tf ->knock-out conditions. (list of TFs)
    >%rxn_affected_ko -> the list of reaction affected
    under each
    >%KO condition.
    >%rxn_prob_ko -> the list of conditional
    probabilities for reactions affected under each KO
    condition.
    >R_path=['"C:\Program Files\R\R-
4.0.3\bin\Rscript.exe"'];
    >Rfun_path=[pwd,'\BN_Module'];
    >Rmodel_path=[pwd,'\learned_network.bif'];
    >[rxn_affected_ko,rxn_prob_ko]=prob_estimation_R(trim
er,ko_tf,regulator,targets,R_path,Rmodel_path,'Rfun_path',Rfun
_path,'mode','BN');
```

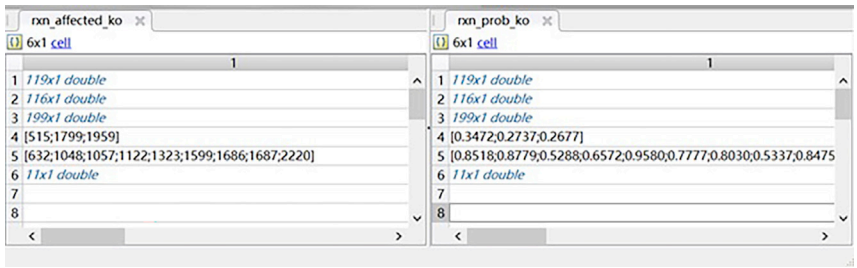


Figure 4. An example list of affected metabolic reactions with the corresponding regulatory values for flux constraints after running prob_estimation.m

Optional: TRIMER supports users to compute the conditional probabilities by relative frequencies of TF-gene state pairs as in PROM by calling the function prob_estimation.m.

Construct transcriptional constraints

Timing: 1–5 min

With the list of affected genes and the associated regulatory values, new flux constraints can be derived for each TF KO condition. In this way, regulatory relationships between TFs and genes are integrated into predicting formulations (e.g., FBA).

4. Regulatory flux constraints are constructed by calling the function `regulatory_bound.m`. New flux constraints are obtained by multiplying the regulatory values computed from inferred conditional probabilities in the previous major step with the max/min flux values estimated via flux variability analysis (FVA) (Mahadevan and Schilling, 2003). For stability, flux constraints with values smaller than a threshold value (e.g., 10^{-6}) set by users are treated as zero.
5. Demo code for building new flux constraints is shown below:

```
>% Threshold value is set to be 1e-6

>% lb_est/ub_est -> constructed flux bounds

>% vmax -> maximum/minimum flux values estimated by FVA

>[lb_est,ub_est,vmax]=regulatory_bound(trimer,ko_tf,rxn_

affected_ko,rxn_prob_ko,'thresh',1e-6);
```

6. TRIMER allows users to refine the regulatory flux bounds by calling the function `regulatory_bound.m`. Setting a minimum growth rate (biomass flux) requirement, this function returns the minimum set of regulatory flux constraints that cannot be satisfied when finding a feasible solution. This helps users to remove or adjust the bounds that over-constrained metabolic models.
7. Demo code for flux constraint refinement is shown below:

```
>% the minimum growth rate is 0.1 times the Wildtype

rate

>% lb_est/ub_est -> constructed flux bounds

>% infeasible -> lists of reaction index
```

```
>[infeasible]=find_infeasible_constrain(trimer,lb_est,ub
_est rxn_affected,'obj_frac',0.1);
```

Make flux predictions

Timing: 1–5 min

With constructed additional flux constraints for each KO condition, flux prediction can be made by optimizing predicting formulations using either GLPK or CPLEX solver.

8. Three kinds of metabolic predicting formulations are implemented in TRIMER, including FBA, sFBA, ROOM. sFBA denotes the formulation adopted in PROM. The main difference between sFBA and FBA is that fluxes are constrained by soft bounds in sFBA instead of hard bounds in the original FBA formulation. Users can easily implement these formulations by calling the function `ko_prediction.m`. [Troubleshooting 5]
9. Demo code for flux prediction is shown below:

```
>%f ->biomass(growth rate) predictions
>%v ->solutions (all reaction flux predictions)
>%states -> types of solution return by solvers
>[f,v,status]=ko_prediction(trimer,lb_est,ub_est,rxn_aff
ected_ko,vmax,'growth_pos',growth_pos,'method','sfba');
```

10. Besides the three formulations, TRIMER provides a set of useful functions that allow users to explore other formulations. Details about function usages can be found in their MATLAB help documents. Figure 6 shows the data structure used to represent all flux constraints. In this protocol, the authors take the implementation of sFBA as an example to show how to use these functions. The implementation can be achieved by the following steps:

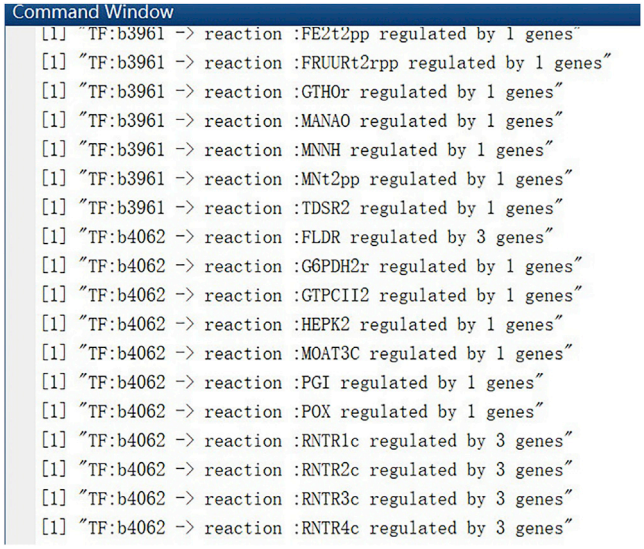


Figure 5. Intermediate displayed output when running the function prob_estimation.m

a. Add binary slack variables α and β:

```
>%add new variable beta and alpha
>var_beta=map(@(x) ['Beta_' x],trimer.rxns);
>var_alpha=map(@(x) ['Aplha_' x],trimer.rxns);
```

b. Initialize bounds of α and β:

```
>trimerS= add_column(trimer,var_alpha,'c',0,0);
>trimerS= add_column(trimerS,var_beta,'c',0,0);
```

c. Add linear constraints: α+v>0 and v-β<0:

```
>lna=T_linalg({{eye(nRxns),trimerS.rxns},{eye(nRxns),va
r_alpha}},'>',trimer.lb(1:nRxns));
>lnb=T_linalg({{eye(nRxns),trimerS.rxns},{eye(nRxns),va
r_beta}},'<',trimer.ub(1:nRxns));
>trimerS=add_matrix_constraint(trimerS,{lna,lnb},{'Alph
a_','Beta_'});
```

d. Initialize flux bounds:

```
>trimerS=change_bound(trimerS,max(trimer.ub(1:nRxns)),'
u',trimerS.rxns);
>trimerS=change_bound(trimerS,min(trimer.lb(1:nRxns)),
'l',trimerS.rxns);
```

e. Estimate the wild-type reaction fluxes:

```
>sol=fba(trimer); f0=sol.val;v0=sol.x(1:nRxns)
```

f. Relax upper bounds of α and β to allow the fluxes of potentially affected reactions to exceed its bounds:

```
>ub_alpha(temprxnpos(v0(temprxnpos)<0))=max(trimer.ub(
1:nRxns));
>ub_beta(temprxnpos(v0(temprxnpos)>0))=max(trimer.ub(1
:nRxns));
>trimerS=change_bound(trimerS,ub_alpha,'u' ,var_alpha)
;
>trimerS=change_bound(trimerS,ub_beta, 'u',var_beta);
```

g. Compute the penalties for exceeding upper/lower flux bounds:

```
>mthresh = 10^(-3); %for computation stability
>vv=abs(vm(temprxnpos));
>vv(vv<mthresh)=mthresh;
>vv=(kappa*(-1)*abs(f0))./ vv;
>weights_alpha(temprxnpos(v0(temprxnpos)<0)) =
vv(v0(temprxnpos)<0);
>weights_beta(temprxnpos(v0(temprxnpos)>0)) =
vv(v0(temprxnpos)>0);
```

h. Change the optimizing objective:

```
>trimerS=change_obj(trimerS, weights_alpha, var_alpha);
```

i. Replace the initial constraints with new regulatory constraints

```
>%lbg/ubg-> new regulatory constraints
>lnpl=T_linalg({{eye(nRxns),trimer.rxns},{eye(nRxns),va
r_alpha}},'>',lbg);
>lnpu=T_linalg({{eye(nRxns),trimer.rxns},{eye(nRxns),va
r_beta}},'<',ubg);
>trimerS=update_matrix_constraint(trimerS,{lnpl,lnpu},{
'Alpha_','Beta_'});
```

j. Make predictions for TF-knockout conditions:

```
>sol=fba(trimerS);
```

The data structures for constraint representations used in TRIMER are compatible with those used in TIGER. Flux constraints constructed by TRIMER can be directly added into the data structures in TIGER. This allows users to build the hybrid TRIMER-TIGER model where a part of flux constraints is constructed by TRIMER and the other part can be constructed by TIGER. This allows the modeling of the feedback effect from metabolites to genes. These feedback regulations cannot be obtained from gene expression data and are often expressed as Boolean rules. Demo code for using the TRIMER-TIGER model to predict flux phenotypes under various growth conditions can be found at the following path:

```
>TRIMER\phenotype.m.
```

EXPECTED OUTCOMES

The main outcome of this protocol is to explain the procedure for using TRIMER to predict metabolic behaviors with TF knockouts. Here we present an example of the pipeline applied to the metabolic model for *E. coli*: iAF1260, following the workflow shown in Figure 3. The procedure begins with

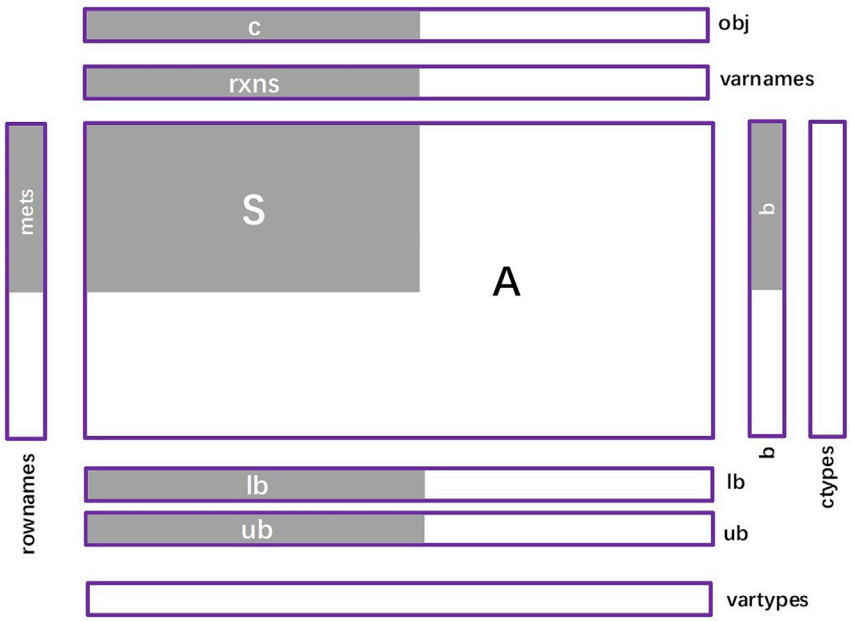


Figure 6. TRIMER metabolic network analysis data structure

The data structure used to represent flux constraints: Gray areas represent the information obtained from the metabolic model; white areas represent the information of additional flux constraints added by users (Modified based on two figures in the original publication (Niu et al., 2021), which is an open access article under the CC BY-NC-ND license: http://creativecommons.org/licenses/by-nc-nd/4.0/).

learning a Bayesian network based on the gene expression data and TF-gene interaction list. Then conditional probabilities can be inferred from the learned BN and used to compute the regulatory values for affected reactions. Based on these regulatory values and maximum/minimum flux values estimated by FVA, new flux bounds can be constructed and added to the flux prediction formulations, such as FBA, sFBA, and ROOM. Finally, metabolic behaviors represented as changes of flux values can be obtained by optimizing these formulations using CPLEX or GLPK. Exemplary data including predicted biomass fluxes for *E.coli* and corresponding intermediate result like regulatory values for affected reactions can be found at the following path:

```
>TRIMER\grRate_KO.m.
```

While this example is case-specific, a similar procedure described in this protocol can be applied to other organisms. Our code and step-by-step descriptions are intended to make the way of applying TRIMER more accessible to non-experts and to serve as a guide to other investigators for combining *in silico* flux simulations with regulatory network modeling.

QUANTIFICATION AND STATISTICAL ANALYSIS

To evaluate the performance of metabolic predictions by TRIMER, Pearson Correlation Coefficient (PCC) is used to evaluate the agreement between predicted fluxes and measured fluxes obtained experimentally. If TF knockout strains are tested under multiple growth conditions, we suggest using the normalized PCC, which is defined as the PCC between experimental flux ratios and predicted flux ratios. The flux ratio is computed by normalizing the knockout mutant strain fluxes by the corresponding wild-type flux in the corresponding growth condition. In this way, the normalized PCC removes the influence due to different growth conditions and can better illustrate how well is the knockout flux predictions (Table 2 show an example of predicted fluxes and experimentally measured ground-truth fluxes with the corresponding PCCs and normalized PCCs).

Table 2. An example of predicted fluxes and experimentally measured ground-truth fluxes under different growth conditions

Knock-out type	Ground-truth	Predicted
'WT+O2'	0.71	0.708
'arcA+O2'	0.686	0.61
'fnr +O2'	0.635	0.547
'arcA fnr +O2'	0.648	0.619
'appY +O2'	0.636	0.708
'oxyR +O2'	0.637	0.708
'soxS +O2'	0.724	0.707
'WT -O2'	0.485	0.481
'arcA -O2'	0.377	0.071
'fnr -O2'	0.41	0.371
'arcA fnr -O2'	0.301	0.16
'appY -O2'	0.476	0.481
'oxyR -O2'	0.481	0.481
'soxS -O2'	0.465	0.481
PCC	-	0.906
Normalized PCC	-	0.841

The unit of fluxes is mmol/gDCW/hr; fluxes are measured under aerobic and anaerobic conditions with glucose as the only substrate (Covert et al., 2004); uptake rates of glucose and oxygen were set to be 8.5 and 14.6 mmol/gDCW/hr when making predictions for the aerobic condition; and 20.8 and 0 14.6 mmol/gDCW/hr for the anaerobic condition, respectively.

LIMITATIONS

In TRIMER, the set of affected genes for a set of TFs considered is determined by the annotated TF-gene pair list. In this protocol, RegulonDB, a gold-standard TF-gene list with strong experimental validation support is used. Such a validated list is important for achieving reasonable prediction performances. However, it should be noted that the gold-standard interactions are not necessarily perfect and may contain false-positive interactions. Similarly, the low-confidence interactions identified from other sources, for example by computational inference methods, could be either false positives or true interactions that have not been validated yet (Chandrasekaran and Price, 2013). This means that potential performance improvement can be still possible with better interaction lists (Wang et al., 2017). However, currently there is no commonly-adopted procedure to refine TF-gene interactions.

For metabolic network modeling, there is no standard operating procedure for determining the uptake rates of nutrients represented by lower bounds of the corresponding reactions. In the absence of *in vivo* uptake rates obtained from time-course metabolomic experiments, it is suggested to use uptake rates measured under wild-type conditions, which may not achieve the best flux predictions.

TROUBLESHOOTING

Problem 1

Error occurs when using `cobra_to_trimer.m` to processGPR rules.

Potential solution

Only when GPR rules are in the standard form shown on the left column of Figure 2, running `cobra_to_trimer.m` is guaranteed to transform them into the MATLAB recognizable form as shown on the right column of Figure 2. Otherwise, users are suggested to manually transform them into the standard form first before running `cobra_to_trimer.m`.

Problem 2

Failure to call the R script for probability inference in the MATLAB environment.

Potential solution

It is likely that the input R environment path is incorrect. Users should specify the installation path of the R environment when using TRIMER. The installation path of the R environment varies depending on the user's computer setup. Here is an example to define the path:

```
>R_path=['"C:\Program Files\R\R-4.0.3\bin\Rscript.exe"'];

>[rxn_affected_ko,rxn_prob_ko]=prob_estimation_R(trimer,ko_tf,regulator,targets,R_path);
```

Problem 3

The running time and required memory to compute flux bound constraints for affected reactions is high.

Potential solution

TRIMER supports two ways of computing the flux bounds. The first relies on estimating probabilities in the form of P(gene=1|TF=0) for each TF-gene pair, which is computationally efficient. In the second way, the probabilities are estimated in the form of P(genes=π|TFs=0) and are inferred by considering each valid gene-TF state pair denoted by π. It is possible that for a given reaction, the number of genes that affect this reaction and are also affected by the TF knockout is large. This means that there are an exponentially growing number of probability values to estimate with respect to the number of genes. Thus, implementing the second way can be computationally expensive. To avoid this computational problem in TRIMER, the default setup to infer conditional probabilities is to estimate P(gene=1|TF=0) for each TF-gene pair when the number of genes is higher than 12. Users can change this setting based on their need and computer setups.

Problem 4

There may be two potential issues when running functions for inferring conditional probabilities: (1) by multiple function calls for inference, the returned results are different; (2) the returned probability values may not satisfy the normalization axiom.

Potential solution

The default inference method in bnlearn is by logic sampling, an approximate inference algorithm, so that the returned results by multiple separate function calls can be different and they may not add up to 1. If these are observed, the user should increase the sample size of logical sampling to achieve more robust estimates. The default choice of the sample size is 5000 x $\log_{10}$ (number of BN model parameters).

Problem 5

MATLAB crashes when using the ROOM formulation to predict fluxes for some knockout conditions.

Potential solution

MATLAB often freezes when mathematical programming solvers (CPLEX or GLPK) are unable to reach optimality, and instead iterate the same solution repeatedly. It is a common phenomenon when solving mix-integer programming problems, such as ROOM. To address this issue, users can change the settings of the solvers to avoid this. For example, users can set the maximum running time of the solver to 10 min.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Xiaoning Qian, xqian@ece.tamu.edu.

Materials availability

This study did not generate new unique reagents.

Data and code availability

The developed package TRIMER is available online in an open-source GitHub repository: https://github.com/niupuhua1234/TRIMER (https://doi.org/10.5281/zenodo.5880036). All the implemented functions in TRIMER are documented with MATLAB's help function. Code for the reported experiments in this paper can also be found in the referred GitHub repository.

ACKNOWLEDGMENTS

The materials presented in this paper are based upon the work supported by the U.S. Department of Energy, Office of Science, Office of Biological and Environmental Research, DE-SC0012704. P.N. and X.Q. are partially supported by the National Science Foundation under grant CCF-1553281. This work has been supported by the DOE Joint Genome Institute (http://jgi.doe.gov) by the U.S. Department of Energy, Office of Science, Office of Biological and Environmental Research, through contract DE-AC02-05CH11231 between Lawrence Berkeley National Laboratory and the U.S. Department of Energy.

AUTHOR CONTRIBUTIONS

Conceptualization, B.J.Y., E.R.D., F.J.A., I.B., and X.Q.; methodology, P.N., B.J.Y., and X.Q.; investigation, P.N., M.J.S., B.J.Y., E.R.D., F.J.A., I.B., and X.Q.; formal analysis, P.N., B.J.Y., E.R.D., F.J.A., and X.Q.; software, P.N.; experimental validation, M.J.S. and I.B.; writing – original draft, P.N., M.J.S., I.B., and X.Q.; writing – review & editing, P.N., M.J.S., B.J.Y., E.R.D., F.J.A., I.B., and X.Q.; funding acquisition, B.J.Y., E.R.D., F.J.A., I.B., and X.Q.; resources, F.J.A., I.B., and X.Q.; supervision, I.B. and X.Q.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

Bonneau, R., Reiss, D.J., Shannon, P., Facciotti, M., Hood, L., Baliga, N.S., and Thorsson, V. (2006). The inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets *de novo*. Genome Biol. *7*, R36. https://doi.org/10.1186/gb-2006-7-5-r36.

Bordbar, A., Monk, J.M., King, Z.A., and Palsson, B.O. (2014). Constraint-based models predict metabolic and associated cellular functions. Nat. Rev. Genet. *15*, 107–120.

Carrera, J., Estrela, R., Luo, J., Rai, N., Tsoukalas, A., and Tagkopoulos, I. (2014). An integrative, multi-scale, genome-wide model reveals the phenotypic landscape of *Escherichia coli*. Mol. Syst. Biol. *10*, 735.

Chandrasekaran, S., and Price, N.D. (2010). Probabilistic integrative modeling of genome-scale metabolic and regulatory networks in *Escherichia coli* and *Mycobacterium tuberculosis*. Proc. Natl. Acad. Sci. U S A *107*, 17845–17850.

Chandrasekaran, S., and Price, N.D. (2013). Metabolic constraint-based refinement of transcriptional regulatory networks. PLoS Comput. Biol. *9*, e1003370.

Covert, M.W., and Palsson, B.O. (2003). Constraints-based models: regulation of gene expression reduces the steady-state solution space. J. Theor. Biol. *221*, 309–325.

Covert, M.W., Knight, E.M., Reed, J.L., Herrgard, M.J., and Palsson, B.O. (2004). Integrating high-throughput and computational data elucidates bacterial networks. Nature *429*, 92–96.

Covert, M.W., Xiao, N., Chen, T.J., and Karr, J.R. (2008). Integrating metabolic, transcriptional regulatory and signal transduction models in *Escherichia coli*. Bioinformatics *24*, 2044–2050.

Haury, A.C., Mordelet, F., Vera-Licona, P., and Vert, J.P. (2012). TIGRESS: trustful inference of gene regulation using stability selection. BMC Syst. Biol. *6*, 1–17.

Heirendt, L., Arreckx, S., Pfau, T., Mendoza, S.N., Richelle, A., Heinken, A., Haraldsdóttir, H.S., Wachowiak, J., Keating, S.M., Vlasov, V., and Magnusdóttir, S. (2019). Creation and analysis of biochemical constraint-based models using the COBRA Toolbox v. 3.0. Nat. Protoc. *14*, 639–702.

Højsgaard, S. (2012). Graphical independence networks with the gRain package for R. J. Stat. Softw. *46*, 1–26.

Huynh-Thu, V.A., Irrthum, A., Wehenkel, L., and Geurts, P. (2010). Inferring regulatory networks from expression data using tree-based methods. PLoS One *5*. https://doi.org/10.1371/journal.pone.0012776.

Jensen, P.A., Lutz, K.A., and Papin, J.A. (2011). TIGER: toolbox for integrating genome-scale metabolic models, expression data, and transcriptional regulatory networks. BMC Syst. Biol. *5*, 1–12.

Mahadevan, R., and Schilling, C.H. (2003). The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. Metab. Eng. *5*, 264–276.

Nagarajan, R., Scutari, M., and Lèbre, S. (2013). Bayesian Networks in R with Applications in Systems Biology (Springer-Verlag).

Niu, P., Soto, M.J., Yoon, B.J., Dougherty, E.R., Alexander, F.J., Blaby, I., and Qian, X. (2021). TRIMER: transcription regulation integrated with MEtabolic regulation. iScience *24*, 103218. https://doi.org/10.1016/j.isci.2021.103218.

Shlomi, T., Eisenberg, Y., Sharan, R., and Ruppin, E. (2007). A genome-scale computational study of the interplay between transcriptional regulation and metabolism. Mol. Syst. Biol. *3*, 101.

Wang, Z., Danziger, S.A., Heavner, B.D., Ma, S., Smith, J.J., Li, S., Herricks, T., Simeonidis, E., Baliga, N.S., Aitchison, J.D., and Price, N.D. (2017). Combining inferred regulatory and reconstructed metabolic networks enhances phenotype prediction in yeast. PLoS Comput. Biol. *13*, e1005489.

Yurkovich, J.T., and Palsson, B.O. (2015). Solving puzzles with missing pieces: the power of systems biology. Proc. IEEE *104*, 2–7.