

SOFTWARE

Open Access

sybil – Efficient constraint-based modelling in R

Gabriel Gelius-Dietrich, Abdelmoneim Amer Desouki, Claus Jonathan Fritzsche and Martin J Lercher*

Abstract

Background: Constraint-based analyses of metabolic networks are widely used to simulate the properties of genome-scale metabolic networks. Publicly available implementations tend to be slow, impeding large scale analyses such as the genome-wide computation of pairwise gene knock-outs, or the automated search for model improvements. Furthermore, available implementations cannot easily be extended or adapted by users.

Results: Here, we present sybil, an open source software library for constraint-based analyses in R; R is a free, platform-independent environment for statistical computing and graphics that is widely used in bioinformatics. Among other functions, sybil currently provides efficient methods for flux-balance analysis (FBA), MOMA, and ROOM that are about ten times faster than previous implementations when calculating the effect of whole-genome single gene deletions *in silico* on a complete *E. coli* metabolic model.

Conclusions: Due to the object-oriented architecture of sybil, users can easily build analysis pipelines in R or even implement their own constraint-based algorithms. Based on its highly efficient communication with different mathematical optimisation programs, sybil facilitates the exploration of high-dimensional optimisation problems on small time scales. Sybil and all its dependencies are open source. Sybil and its documentation are available for download from the comprehensive R archive network (CRAN).

Keywords: Constraint-based modelling, Flux-balance analysis, FBA, MOMA, ROOM, GNU R

Background

Constraint-based analyses have become a widely used tool for the study of genome-scale biochemical reaction networks [1]. The most prominent of these methods is flux-balance analysis (FBA). Here, metabolite fluxes through biochemical reactions are constrained by the conservation of mass, by thermodynamics (reaction directionality), by the assumption of a steady state for internal metabolite concentrations, and by the availability of nutrients. These constraints are used as boundary conditions for a linear optimisation problem, in which a biologically motivated objective function — often the yield of biomass production — is maximised. The result is a distribution of metabolic fluxes across the network, comprising a metabolic phenotype (or functional state) of the network [2-4].

Apart from such constraint-based optimisation methods, several other tools that use different philosophies for metabolic modelling are available. One example is

the computation of elementary flux modes to represent the feasible solution space of a metabolic network [5]. Another approach is structural kinetic modelling, i.e., the description of dynamical properties of metabolic networks in combination with experimental data [6].

Several tools for constraint-based optimisation analyses are currently available (reviewed in [7-9]). The most widely used software is the COBRA Toolbox [10] for MATLAB. While these tools provide implementations of FBA and other constraint-based methods, they are relatively slow when applied to large series of simulations (e.g., when calculating the biomass yield of all double-gene knockouts in a unicellular organism). Further, available implementations mostly require licenses for MATLAB, and are not flexible enough to allow users to easily design their own large-scale analyses. For metabolic networks for which elementary modes [11] or extreme pathways [12,13] can be calculated, such higher-level descriptions of the solution space may provide fast alternatives to the constrained-based algorithms implemented in sybil.

The R computer language has become the standard programming environment in many scientific fields that

*Correspondence: lercher@cs.uni-duesseldorf.de
Institute for Computer Science, Heinrich-Heine-University, Universitätsstr 1,
40225 Düsseldorf, Germany

depend on numerical data analysis, in particular in the analysis of biological high-throughput data. However, R currently offers only very limited options for constraint-based analyses. The R package BiGGR [14] provides access to the BiGG database [15] and can perform flux-balance analysis, visualising the results as graphs. The R package abcdeFBA [16] provides flux-balance analysis and phenotypic phase plane analysis. However, both packages are limited in scope and lack flexibility.

With sybil, which shares some functionality with the COBRA Toolbox and the R packages described above, we aim to establish R as a major platform for constraint-based analyses of biological networks. Besides offering powerful analysis tools in a versatile and freely available environment, sybil aims to supersede previous implementations in terms of calculation speed, flexibility and extensibility.

Implementation

Sybil is implemented in the R programming language [17] as an object oriented library (Additional file 1). The design of some of its functions was inspired by the COBRA Toolbox [10]. Once the sybil library is loaded into the R environment, the user can access a range of functions to read and manipulate metabolic network models, to perform different constraint-based calculations, and to visualise the results.

Sybil is programmed for both speed and memory efficiency; in our experience, about 1 GB of RAM should be sufficient for all types of analyses, even when performed on the largest complete single cell type metabolic models currently available.

Sybil provides a set of “high-level” functions to access frequently used complex algorithms with a single function call (e.g., `fluxVar()` for flux variability calculations [18,19], or `geneDeletion()` for prediction of gene deletion effects). Another way to use sybil is to directly use “low-level” functions (e.g., `optimizeProb()` or any of the API-functions from the linked optimiser software). Methods implemented in class `sysBioAlg` provide a particularly comfortable way to execute constraint-based analyses involving optimisation steps (FBA and related algorithms): here, the class takes care of the optimisation software without user interference. Sybil’s architecture provides the user with a highly flexible and adjustable framework. Sybil is equally suited for off-the-shelf constraint-based analyses, for building complex analysis pipelines, and for the development of new constraint-based analysis methods.

The implementation of sybil follows the object oriented programming paradigm. Figure 1 shows the connection between the important classes implemented in sybil. Class `modelorg` contains sybil’s representation of a metabolic network.

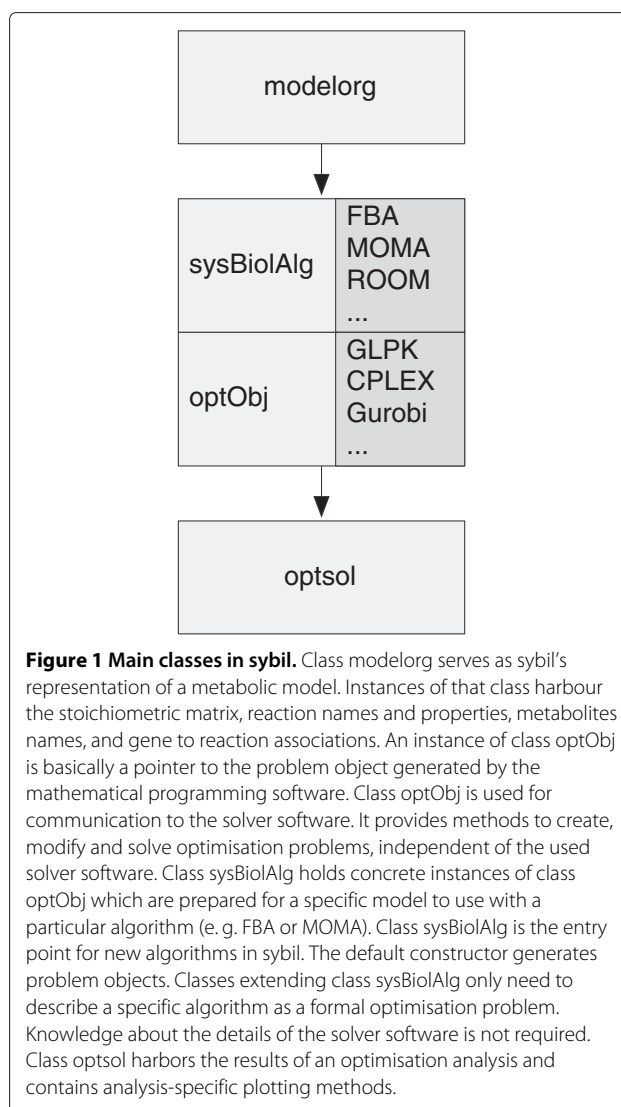


Figure 1 Main classes in sybil. Class `modelorg` serves as sybil’s representation of a metabolic model. Instances of that class harbour the stoichiometric matrix, reaction names and properties, metabolites names, and gene to reaction associations. An instance of class `optObj` is basically a pointer to the problem object generated by the mathematical programming software. Class `optObj` is used for communication to the solver software. It provides methods to create, modify and solve optimisation problems, independent of the used solver software. Class `sysBioAlg` holds concrete instances of class `optObj` which are prepared for a specific model to use with a particular algorithm (e.g. FBA or MOMA). Class `sysBioAlg` is the entry point for new algorithms in sybil. The default constructor generates problem objects. Classes extending class `sysBioAlg` only need to describe a specific algorithm as a formal optimisation problem. Knowledge about the details of the solver software is not required. Class `optsol` harbors the results of an optimisation analysis and contains analysis-specific plotting methods.

A number of functions are available to manipulate metabolic network models, such as `addReact()` to add new reactions to the model, `changeGPR()` to alter the gene-reaction association rules, and `changeUptake()` and `editEnvir()` to change the modelled environments. Instances of class `sysBioAlg` contain a pointer to the problem object, comprised of metabolic model, constraints, and analysis algorithm to be used. For applications that involve repetitive analyses, such as flux variability or genome-wide knockout studies, the problem object used by the optimisation software is prepared only once as an instance of class `sysBioAlg`. Modifications to the problem required in the course of the analysis are then applied at the level of class `sysBioAlg`, so that the problem object must not be re-created for every optimisation. The results returned by the mathematical programming software are stored in instances of class `optsol`.

Results and discussion

Key features

Sybil provides several functions to perform constraint-based analyses of metabolic networks. Genetic perturbations can be simulated through FBA [2,3], minimisation of metabolic adjustment (MOMA) [20], a linear version of the MOMA algorithm similar to [21], or regulatory on/off minimisation (ROOM) [22]. Additionally, sybil can perform flux variability (FVA) [18,19], robustness [23], and phenotypic phase plane (PhPP) [24,25] analyses (see Additional file 2 for a comparison with other constraint-based analysis tools). The implementations are optimised for speed when running a large number of similar optimisations on the same model (e.g. genome-wide gene deletion simulations).

Due to sybil's object oriented implementation, users can easily add new functions. Class `sysBioAlg` can be extended to implement additional algorithms, which are then available to high-level functions in sybil without further user interaction. Like other toolboxes for constraint-based analyses, sybil communicates with external mathematical optimisation software (e.g., GLPK) to generate and solve various types of optimisation problems. This process is handled by class `optObj`, which provides a large set of methods to generate, modify, and solve mathematical programming problems and to access the results; the user does not need any deeper knowledge about the differences of the various solvers that can be used by sybil. However, if necessary, all parameters available within the solver software can be accessed directly in sybil.

In the future, we plan to further extend sybil, e.g. by adding methods that incorporate gene expression data into an FBA approach [26-29]. Two such additions are already implemented in the separate R packages `sybil-DynFBA` [30], which uses dynamic FBA simulations to predict concentration changes of external metabolites as described in [31], and `sybilEFBA` [32] using gene expression data to improve FBA predictions. Another available add-on to sybil is the R package `RSeed` [33], which analyses network topology to identify metabolites that must be acquired from the environment [34]. The R package `sybilSBML` (Additional file 3) adds SBML support to sybil.

Calculation speed

The calculation speed of the optimisations depends on the mathematical optimisation software used. Typically, for large mathematical problems, IBM ILOG CPLEX is slightly faster than the two freely available solvers GLPK and COIN-OR Clp (see below). However, major differences in the running times of different constraint-based analysis tools stem mostly from the overheads produced by the communication between the main program and the solver. This overhead is minimised by sybil

through purpose-built fast interfaces to the C-API of each package.

Most of the implemented algorithms require the generation of an optimisation problem based on the model, the constraints, and the desired algorithm (such as FBA or linear MOMA). During batch calculations, only small changes to the optimisation problem are required, e.g., changes of variable bounds in an *in silico* gene deletion experiment, or alteration of the objective function during flux variability analysis. To speed up iterations over many such small changes, the optimisation problem is formulated only once; all changes are then applied directly to the pre-formed optimisation problem of the mathematical optimisation software.

Figures 2 and 3 compare the running times of different implementations of typical algorithms used in constraint-based modelling; they also illustrate the impact of different mathematical optimisation programs on calculation speed. For all calculations, we used a complete model of *E. coli* metabolism, containing 2382 reactions and 1261 independent genes [35].

Figure 2 shows the performance of different implementations of genome-wide flux variability analysis (FVA) [18,19] using GLPK as the mathematical optimisation program. For FVA, an optimal growth rate was estimated by FBA. Then, for all reactions in the model, we computed the minimal and maximal flux at this growth rate. The software tools `fastFVA` [36], `COBRA Toolbox` [10], and `CellNetAnalyzer` [37] implement FVA for the MATLAB environment; `SBRT` [38] and `OptFlux` [39] are Java-based implementations; `FASIMU` [40] is implemented in bash and awk; `COBRAPy` [41] is a Python implementation; and `abcdeFBA` and sybil provide R implementations. All these software packages perform FVA with a single function call.

As can be seen in Figure 2, `fastFVA` is the fastest implementation of the flux variability algorithm. The main algorithm is fully implemented in C++ and can be accessed from within MATLAB as an extension to the `COBRA Toolbox`. The C++ implementation results in a very fast running time, but makes the program inflexible; only flux variability analysis can be performed, and changes to the solver software parameters require modification of the source code. Sybil, the second fastest implementation, is — compared to other implementations — only slightly slower than `fastFVA`. Sybil's optimisations make use of wrapper functions (in this case through the R package `glpkAPI` [42]), allowing access to the C-API of the mathematical programming software from within R. This combines very short running times with flexible communication with the solver software. `SBRT` uses its own Java interface to GLPK (and IBM ILOG CPLEX), which is in function similar to the wrapper software used by sybil. In `COBRAPy`, a separate Python module provides

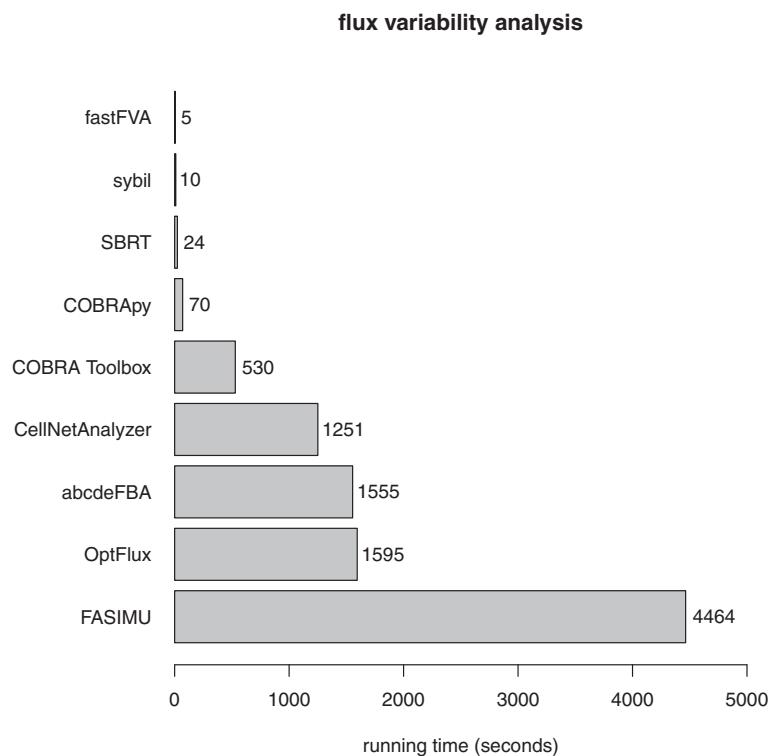


Figure 2 Running times for flux variability analysis. Running time of flux variability analysis in various software packages using the GNU Linear Programming Kit (GLPK). Time was measured in R with the function `Rprof()` and in MATLAB with the Profiler function. In both cases, the running time of a function is reported as “total time”. For OptFlux, the Unix command `top` in delta mode was used. SBRT itself reports the elapsed time for flux variability analysis. For FASIMU, the “real time” reported by the Unix command `time` was used. For COBRApy, the Python module time was used. All simulations were run ten times on the same desktop PC (Intel Xeon, 2.8 GHz, running Mac OS 10.7). The reported running times are arithmetic means of these values.

a connection to GLPK. The MATLAB packages COBRA Toolbox and CellNetAnalyzer make use of `glpk mex` [43], which provides high-level function calls to build and solve mathematical programming problems in one step. This architecture results in longer running times, as the problem needs to be rebuilt for every step in flux variability analysis, even if only minor adjustments to the model are required. The R package `abcdeFBA` uses the R package `Rglpk` [44], which works similar to `glpk mex`. OptFlux and FASIMU use the command line interfaces of GLPK and IBM ILOG CPLEX (FASIMU) or COIN-OR Clp (OptFlux) and generate the necessary input files for every optimisation. FASIMU computes the optimisations one by one, resulting in the longest running time, while OptFlux can run — to some extent — optimisations simultaneously.

Figure 3 shows the performance of genome-wide *in silico* gene deletion experiments with the same complete model of *E. coli* metabolism used for the flux variability analyses. Regardless of the details of the experiment (gene vs. flux deletions; single- vs. double-gene deletions; FBA vs. linear MOMA), `sybil` clearly outperforms other

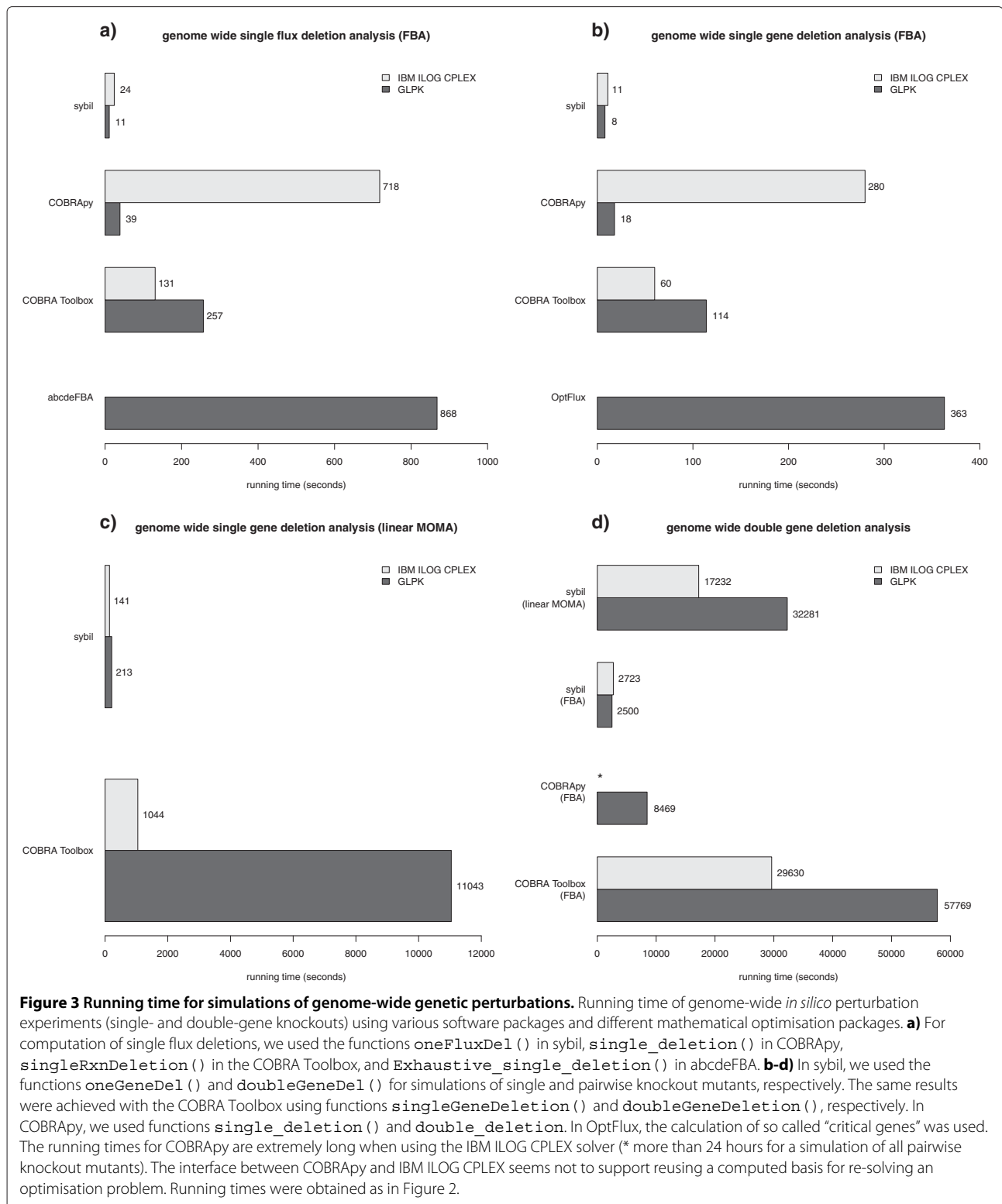
implementations in terms of computation speed; this is achieved through the efficient handling of optimisation problems that repeatedly need to be re-optimised, but do not change very much from one optimisation to the other. `Sybil` was successfully used as the constraint-based core of a machine learning method to reconcile model predictions with genome-scale experimental double-gene knockout data [45]. In this study, we demonstrated the feasibility of automated metabolic model refinement by correcting misannotations in NAD biosynthesis in the metabolic model of yeast (`iMM904`, [46]).

Another fast tool is `F2C2` [47], a MATLAB tool for flux coupling analysis which computes all blocked and coupled reactions of the *E. coli* model in less than five minutes on our test system.

Examples

Reading model files

`Sybil` can read text-based representations of metabolic networks written in the ‘Systems Biology Markup Language’ (SBML) [48], which is an extension of XML. A large range of models in this de-facto standard format is avail-



able from the web pages of the Palsson group at UCSD [49]. Each of these models is the outcome of an elaborate model-building process, which starts from database and literature searches and culminates in an iterative

comparison of computational predictions and lab experiments. Details on how to reconstruct whole genome metabolic network models suitable for constraint based analyses are reviewed in [50,51]. A reconstruction of the

central energy metabolism of *E. coli* [23] is included as an example dataset (Additional file 4). In order to read SBML files, the package `sybilSBML` (Additional file 3) from CRAN, which is itself powered by `LibSBML` [52], is required.

`Sybil` can also read models written in a column-based format, such as exported reaction lists of the BiGG database [15]. Example files for the central energy metabolism of *E. coli* are also included in this format (Additional file 4). These can be read in using the command `readTSVmod()` (assuming Additional file 4 is unpacked in the working directory of R):

```
Ec_core <- readTSVmod(prefix = "Ec_core",  
                    quoteChar = "\\")
```

The variable `Ec_core` now contains an *in silico* representation of the central energy metabolism of *E. coli* which can be used for further analysis. The definition of the column-based format is described in the `sybil` package vignette (Additional file 5).

Constraint-based analysis of metabolic networks

Genetic perturbations of metabolic networks can be simulated using the function `geneDeletion()`. The command

```
ko <- geneDeletion(Ec_core, combinations=1,  
                  algorithm = "fba", solver = "glpkAPI")
```

performs a single gene deletion analysis on the example dataset, using flux-balance analysis to determine reductions in metabolite production, and employing the mathematical optimisation software GLPK for the optimisations. The parameter 'combinations' indicates the number of genes to knockout simultaneously in each optimisation. Setting this parameter to 2 results in the simulation of all possible pairwise gene knockouts, setting it to 3 will compute all triple-gene knockouts. Due to `sybil`'s streamlined communication with the solver software, which only transmits changes to the model rather than the full model for each deletion, this function helps to deal with the combinatorial explosion inherent in systematic multiple-gene knockout experiments. The parameter 'algorithm' indicates the algorithm used to determine the functional state of the metabolic network after gene deletion. It can be set to

- "fba": for flux-balance analysis (this is the default value) as described in [2,3],
- "mtf": for flux-balance analysis and additionally selecting the flux distribution resulting in the smallest absolute total flux,

- "moma": for minimisation of metabolic adjustment as described in [20],
- "lmoma": for a linear version of the MOMA algorithm similar to the version described in [21], or
- "room": for regulatory on/off minimisation as described in [22].

The parameter 'solver' selects the mathematical optimisation software used by the algorithms. It can be set to

- "glpkAPI": for GLPK [53], via R package `glpkAPI` [42],
- "cplexAPI": for IBM ILOG CPLEX [54], via R package `cplexAPI` [55],
- "clpAPI": for COIN-OR Clp [56], via R package `clpAPI` [57],
- "lpSolveAPI": for `lp_solve` [58], via R package `lpSolveAPI` [59], or
- "sybilGUROBI": for Gurobi [60], via R package `sybilGUROBI`.

All R packages are available on CRAN [61], with the exception of `sybilGUROBI`, which is available on request. The `sybil` package vignette (Additional file 5) contains further examples of constraint-based metabolic network analyses, such as flux variability or robustness analyses, as well as graphical representation of results and instructions for the interaction with mathematical optimisation programs.

Conclusions

`Sybil` is designed to address large scale questions in reasonable time frames, making it possible to generate and run *in silico* experiments that result in high-dimensional optimisation problems. New algorithms can be easily implemented using the `sybil` framework and can be distributed as add-on packages to the systems biology community.

Availability and requirements

Project name: `sybil`

Project home page: <http://CRAN.R-project.org/package=sybil>

Operating system(s): Platform independent

Programming language: R

Other requirements: A mathematical optimisation software (one of GLPK, IBM ILOG CPLEX, COIN-OR Clp, or `lpSolve`)

License: GNU GPL

Additional files

Additional file 1: R package sybil. Archive of the sybil version current at submission.

Additional file 2: Features contained in sybil and other commonly used software packages.

Additional file 3: R package sybilSBML. Archive of the sybilSBML version current at submission.

Additional file 4: Example dataset. Archive containing the input files required by sybil for the reconstruction of the central energy metabolism of *E. coli* in column based format and in SBML format. The files packaged into the archive can be read with the sybil command `readTSVmod`.

Additional file 5: sybil package vignette. A user guide for sybil in PDF format. It can be accessed from within a running R session with the command `vignette("sybil")`.

Abbreviations

FBA: Flux-balance analysis; FVA: Flux variability analysis; MOMA: Minimisation of metabolic adjustment; ROOM: Regulatory on/off minimisation; SBML: Systems biology markup language.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

GGD developed the R packages sybil, sybilSBML, glpkAPI, clpAPI and cplexAPI, implemented the algorithms, conceived the handling of the mathematical programming software, and wrote the manuscript. AAD developed sybil add-on packages sybilDynFBA and sybilEFBA and tested and applied sybil and made suggestions for improvements and additions. CJF developed sybil add-on package RSeed and tested and applied sybil and made suggestions for improvements and additions. MJL initiated the project, suggested improvements and additions to sybil, and contributed to the writing of the manuscript. All authors read and approved the final manuscript.

Acknowledgements

We are grateful to Balázs Papp and Balázs Szappanos for helpful discussions and intensive testing. We also thank Csaba Pál, Markus Herrgård, Benjamin Braasch, Marc Andre Daxer, Milan Majtanik, and Rajen Piernikarczyk for helpful discussions.

Received: 19 April 2013 Accepted: 1 November 2013

Published: 13 November 2013

References

1. Kauffman KJ, Prakash P, Edwards JS: **Advances in flux balance analysis.** *Curr Opin Biotechnol* 2003, **14**(5):491–496.
2. Edwards JS, Covert M, Palsson BØ: **Metabolic modelling of microbes: the flux-balance approach.** *Environ Microbiol* 2002, **4**(3):133–140.
3. Orth JD, Thiele I, Palsson BØ: **What is flux balance analysis?** *Nat Biotechnol* 2010, **28**(3):245–248.
4. Schuster S, Pfeiffer T, Fell DA: **Is maximization of molar yield in metabolic networks favoured by evolution?** *J Theor Biol* 2008, **252**(3):497–504.
5. Terzer M, Stelling J: **Large-scale computation of elementary flux modes with bit pattern trees.** *Bioinformatics* 2008, **24**(19):2229–2235.
6. Girbig D, Selbig J, Grimbs S: **A MATLAB toolbox for structural kinetic modeling.** *Bioinformatics* 2012, **28**(19):2546–2547.
7. Raman K, Chandra N: **Flux balance analysis of biological systems: applications and challenges.** *Brief Bioinform* 2009, **10**(4):435–449.
8. Dandekar T, Fieselmann A, Majeed S, Ahmed Z: **Software applications toward quantitative metabolic flux analysis and modeling.** *Brief Bioinform* 2012. doi:10.1093/bib/bbs065.
9. Lakshmanan M, Koh G, Chung BKS, Lee DY: **Software applications for flux balance analysis.** *Brief Bioinform* 2012. doi:10.1093/bib/bbs069.
10. Schellenberger J, Que R, Fleming RMT, Thiele I, Orth JD, Feist AM, Zielinski DC, Bordbar A, Lewis NE, Rahmanian S, Kang J, Hyduke DR, Palsson BØ: **Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox v2.0.** *Nat Protoc* 2011, **6**(9):1290–1307.
11. Schuster S, Fell DA, Dandekar T: **A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks.** *Nat Biotechnol* 2000, **18**(3):326–332.
12. Papin JA, Price ND, Palsson BØ: **Extreme pathway lengths and reaction participation in genome-scale metabolic networks.** *Genome Res* 2002, **12**(12):1889–1900.
13. Wiback SJ, Palsson BØ: **Extreme pathway analysis of human red blood cell metabolism.** *Biophys J* 2002, **83**(2):808–818.
14. Gavai AK: **BiGGR.** [http://CRAN.R-project.org/package=BiGGR]
15. Schellenberger J, Park JO, Conrad TM, Palsson BØ: **BiGG: a Biochemical Genetic and Genomic knowledgebase of large scale metabolic reconstructions.** *BMC Bioinformatics* 2010, **11**:213.
16. Gangadharan A, Rohatgi N: **abcdeFBA.** [http://CRAN.R-project.org/package=abcdeFBA]
17. R Development Core Team: *R: A Language and Environment for Statistical Computing.* Vienna: R Foundation for Statistical Computing; 2012. [http://www.R-project.org]. [ISBN 3-900051-07-0].
18. Mahadevan R, Schilling CH: **The effects of alternate optimal solutions in constraint-based genome-scale metabolic models.** *Metab Eng* 2003, **5**(4):264–276.
19. Reed JL, Palsson BØ: **Genome-scale in silico models of *E. coli* have multiple equivalent phenotypic states: assessment of correlated reaction subsets that comprise network states.** *Genome Res* 2004, **14**(9):1797–1805.
20. Segrè D, Vitkup D, Church GM: **Analysis of optimality in natural and perturbed metabolic networks.** *Proc Natl Acad Sci U S A* 2002, **99**(23):15112–15117.
21. Becker SA, Feist AM, Mo ML, Hannum G, Palsson BØ, Herrgård MJ: **Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox.** *Nat Protoc* 2007, **2**(3):727–738.
22. Shlomi T, Berkman O, Ruppin E: **Regulatory on/off minimization of metabolic flux changes after genetic perturbations.** *Proc Natl Acad Sci U S A* 2005, **102**(21):7695–7700.
23. Palsson BØ: *Systems Biology: Properties of Reconstructed Networks.* Cambridge: Cambridge University Press; 2006.
24. Edwards JS, Ramakrishna R, Palsson BØ: **Characterizing the metabolic phenotype: a phenotype phase plane analysis.** *Biotechnol Bioeng* 2002, **77**:27–36.
25. Price ND, Reed JL, Palsson BØ: **Genome-scale models of microbial cells: evaluating the consequences of constraints.** *Nat Rev Microbiol* 2004, **2**(11):886–897.
26. Akesson M, Förster J, Nielsen J: **Integration of gene expression data into genome-scale metabolic models.** *Metab Eng* 2004, **6**(4):285–293.
27. Colijn C, Brandes A, Zucker J, Lun DS, Weiner B, Farhat MR, Cheng TY, Moody DB, Murray M, Galagan JE: **Interpreting expression data with metabolic flux models: predicting *Mycobacterium tuberculosis* mycolic acid production.** *PLoS Comput Biol* 2009, **5**(8):e1000489.
28. van Berlo RJP, de Ridder D, Daran JM, Daran-Lapujade PAS, Teusink B, Reinders MJT: **Predicting metabolic fluxes using gene expression differences as constraints.** *IEEE/ACM Trans Comput Biol Bioinform* 2011, **8**:206–216.
29. Covert MW, Schilling CH, Palsson BØ: **Regulation of gene expression in flux balance models of metabolism.** *J Theor Biol* 2001, **213**:73–88.
30. Amer Desouki A: **sybilDynFBA.** [http://CRAN.R-project.org/package=sybilDynFBA]
31. Varma A, Palsson BØ: **Stoichiometric flux balance models quantitatively predict growth and metabolic by-product secretion in wild-type *Escherichia coli* W3110.** *Appl Environ Microbiol* 1994, **60**(10):3724–3731.
32. Amer Desouki A: **sybilEFBA.** [http://CRAN.R-project.org/package=sybilEFBA]
33. Fritzscheier CJ: **RSeed.** [http://CRAN.R-project.org/package=RSeed]
34. Borenstein E, Kupiec M, Feldman MW, Ruppin E: **Large-scale reconstruction and phylogenetic analysis of metabolic environments.** *Proc Natl Acad Sci U S A* 2008, **105**(38):14482–14487.
35. Feist AM, Henry CS, Reed JL, Krummenacker M, Joyce AR, Karp PD, Broadbelt LJ, Hatzimanikatis V, Palsson BØ: **A genome-scale metabolic**

- reconstruction for *Escherichia coli* K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information. *Mol Syst Biol* 2007, **3**:121.
36. Gudmundsson S, Thiele I: **Computationally efficient flux variability analysis.** *BMC Bioinformatics* 2010, **11**:489.
 37. Klamt S, Saez-Rodriguez J, Gilles ED: **Structural and functional analysis of cellular networks with CellNetAnalyzer.** *BMC Syst Biol* 2007, **1**:2.
 38. Wright J, Wagner A: **The systems biology research tool: evolvable open-source software.** *BMC Syst Biol* 2008, **2**:55.
 39. Rocha I, Maia P, Evangelista P, Vilaça P, Soares S, Pinto JP, Nielsen J, Patil KR, Ferreira EC, Rocha M: **OptFlux: an open-source software platform for in silico metabolic engineering.** *BMC Syst Biol* 2010, **4**:45.
 40. Hoppe A, Hoffmann S, Gerasch A, Gille C, Holzhütter HG: **FASIMU: flexible software for flux-balance computation series in large metabolic networks.** *BMC Bioinformatics* 2011, **12**:28.
 41. Ebrahim A, Lerman JA, Palsson BØ, Hyduke DR: **COBRApy: constraints-based reconstruction and analysis for python.** *BMC Syst Biol* 2013, **7**:74.
 42. Gelius-Dietrich G: **glpkAPI.** [<http://CRAN.R-project.org/package=glpkAPI>]
 43. Giorgetti N: **glpkmex.** [<http://glpkmex.sourceforge.net/>]
 44. Hornik K, Theussl S: **Rglpk.** [<http://CRAN.R-project.org/package=Rglpk>]
 45. Szappanos B, Kovács K, Szamecz B, Honti F, Costanzo M, Baryshnikova A, Gelius-Dietrich G, Lercher MJ, Jelasity M, Myers CL, Andrews BJ, Boone C, Oliver SG, Pál C, Papp B: **An integrated approach to characterize genetic interaction networks in yeast metabolism.** *Nat Genet* 2011, **43**(7):656–662.
 46. Mo ML, Palsson BØ, Herrgård MJ: **Connecting extracellular metabolomic measurements to intracellular flux states in yeast.** *BMC Syst Biol* 2009, **3**:37.
 47. Larhlimi A, David L, Selbig J, Bockmayr A: **F2C2: a fast tool for the computation of flux coupling in genome-scale metabolic networks.** *BMC Bioinformatics* 2012, **13**:57.
 48. Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, Kitano H, Arkin AP, Bornstein BJ, Bray D, Cornish-Bowden A, Cuellar AA, Dronov S, Gilles ED, Ginkel M, Gor V, Goryanin II, Hedley WJ, Hodgman TC, Hofmeyr JH, Hunter PJ, Juty NS, Kasberger JL, Kremling A, Kummer U, Le Novère N, Loew LM, Lucio D, Mendes P, Minch E, Mjolsness ED, et al.: **The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models.** *Bioinformatics* 2003, **19**(4):524–531.
 49. **In Silico Organisms | Systems Biology Research Group** [<http://gcrp.ucsd.edu/Downloads>]
 50. Feist AM, Herrgård MJ, Thiele I, Reed JL, Palsson BØ: **Reconstruction of biochemical networks in microorganisms.** *Nat Rev Microbiol* 2009, **7**(2):129–43.
 51. Oberhardt MA, Palsson BØ, Papin JA: **Applications of genome-scale metabolic reconstructions.** *Mol Syst Biol* 2009, **5**:320.
 52. Bornstein BJ, Keating SM, Jouraku A, Hucka M: **LibSBML: an API library for SBML.** *Bioinformatics* 2008, **24**(6):880–881.
 53. Makhorin A: **GNU Linear Programming Kit (GLPK).** [<http://www.gnu.org/software/glpk/>]
 54. **IBM ILOG CPLEX** [<https://www.ibm.com/developerworks/university/academicinitiative/>]
 55. Gelius-Dietrich G: **cplexAPI.** [<http://CRAN.R-project.org/package=cplexAPI>]
 56. **COIN OR Clp** [<https://projects.coin-or.org/Clp>]
 57. Gelius-Dietrich G: **clpAPI.** [<http://CRAN.R-project.org/package=clpAPI>]
 58. **lp_solve** [<http://lpsolve.sourceforge.net/5.5/index.htm>]
 59. Konis K: **lpSolveAPI.** [<http://CRAN.R-project.org/package=lpSolveAPI>]
 60. **Gurobi** [<http://www.gurobi.com>]
 61. **Comprehensive R Archive Network (CRAN)** [<http://cran.r-project.org>]

doi:10.1186/1752-0509-7-125

Cite this article as: Gelius-Dietrich et al.: sybil – Efficient constraint-based modelling in R. *BMC Systems Biology* 2013 **7**:125.

Submit your next manuscript to BioMed Central and take full advantage of:

- Convenient online submission
- Thorough peer review
- No space constraints or color figure charges
- Immediate publication on acceptance
- Inclusion in PubMed, CAS, Scopus and Google Scholar
- Research which is freely available for redistribution

Submit your manuscript at
www.biomedcentral.com/submit

