EFMEvolver: Computing elementary flux modes in genome-scale metabolic networks

Christoph Kaleta*, Luís Filipe de Figueiredo*, Jörn Behre, and Stefan Schuster†

Department of Bioinformatics, Friedrich Schiller University Jena, Ernst-Abbe-Platz 2, D-07743 Jena, Germany

Abstract: Elementary flux mode analysis (EFM analysis) is an important method in the study of biochemical pathways. However, the computation of EFMs is limited to small and medium size metabolic networks due to a combinatorial explosion in their number in larger networks. Additionally, the existing tools to compute EFMs require to enumerate all EFMs before selecting those of interest. The method presented here extends EFM analysis to genome-scale models. Instead of computing the entire set of EFMs an optimization problem is used to determine a single EFM. Coupled with a genetic algorithm (GA) this allows to explore the solution space and determine specific EFMs of interest. Applied to a network in which the set of EFMs is known our method was able to find all EFMs in two cases and in another case almost the entire set before aborted. Furthermore, we determined the parts of three metabolic networks that can be used to produce particular amino acids and found that these parts correspond to significant portions of the entire networks.
Availability: Source code and an executable are available upon request.

1 Introduction

In the post-genomic era, the analysis of metabolic networks is essential for molecular biology. These networks are complex and the subdivision of a network into pathways makes the analysis more comprehensive. However, the focus only on specific classically known pathways can conceal the view on the actual metabolic capabilities of an organism [KdFS09]. Thus, the construction of genome-scale metabolic networks that model the entire metabolism of organisms has come to importance [FP08].

A method that has been used to comprehensively studying pathways in metabolic networks is elementary flux mode analysis [SDF99]. Elementary flux modes (EFMs) are a systematic definition of the biological concept of a pathway. They correspond to minimal sets of reactions that can perform at steady state [SDF99]. EFM analysis has already been used to study biochemical relevant metabolic pathways [CS04, dFSKF09], to study metabolic network properties such as fragility and robustness [SKB+02, BWvK+08], and to optimize microorganisms with respect to the production of a certain metabolite [TUS08]. However, EFM analysis has been limited to small and medium scale networks because the number of EFMs grows exponentially with the size of the network [KS02]. For instance, Yeung *et*

*Both authors contributed equally
†Corresponding author (`stefan.schu@uni-jena.de`)

al. [YTP07] estimated that the number of extreme pathways [SLP00], a subset of EFMs, is at the order of 10^{29} for a genome-scale model of human.

Due to this problem, alternative approaches for the identification of pathways based on graph theory have been proposed [$\mathrm{RAS}^{+}05$, CCWvH06, BK08]. These methods abstract from the metabolic network by converting it into a graph and consider only connected paths. While they operate efficiently in genome-scale metabolic networks, they bear the problem that a detected pathway does not automatically imply that a net-conversion of the source metabolite into a specific target metabolite is possible [PB08, dFSKF09].

Here we want to present a method that allows the enumeration of EFMs in genome-scale metabolic models. Starting from an initial pathway, the space of EFMs is explored using a genetic algorithm (GA). GAs have already been used in the analysis of metabolic networks to find combinations of gene knockouts that improve the production of a given metabolite [PRFN05]. We used benchmark models for EFM analysis to validate our new method and applied it to a study of amino-acid synthesis in genome-scale metabolic models.

2 Methods

The aim of our algorithm is, given a metabolic network and an input medium, to find all EFMs producing a certain metabolite. The employed strategy is based on the observation that gene knockouts can force an organism to use pathways alternative to those found under standard conditions. Thus, we are detecting EFMs by evolving a population in which each individual corresponds to a set of knockouts. However, instead of considering the knockouts of genes we here focus on the “knockout” of reactions. By searching for a specific EFM avoiding reactions that are knocked out and iterating over different sets of knockouts we are able to determine different EFMs.

2.1 Detecting a single EFM

A metabolic network comprising m metabolites and n reactions is defined by the mxn stoichiometric matrix $\mathbf{N}$. Each metabolite can be defined to be either internal or external. External metabolites differ to internal metabolites in that their concentration is assumed to be buffered by the system. Examples for such external metabolites are energy currency metabolites like ATP, NADH and FADH. Since their concentration is assumed to be constant they are not required to be balanced by an EFM.

To be an EFM, a flux $\mathbf{v} \in \mathbb{R}^n$ through a reaction network has to fulfill the following conditions: (1) steady-state condition, i.e., all internal metabolites are balanced; (2) irreversible reactions have positive fluxes; (3) non-decomposability of the enzyme set, i.e., the non-zero indices of one EFM cannot be a subset of the non-zero indices of another EFM. In our approach reversible reactions are decomposed into two irreversible reactions with opposite directions. Therefore, all fluxes have to be positive.

Given a set K of reactions to be knocked out and an index μ corresponding to a target reaction which produces a certain metabolite of interest, the optimization problem to compute an EFM can be formulated as a linear program by minimizing $\sum_{r=1}^{n} v_r$ subject to

$$\mathbf{N}\mathbf{v} = 0 \quad (1)$$

$$\mathbf{v} \geq 0 \quad (2)$$

$$v_\mu \geq 1 \quad (3)$$

$$\forall i \in K: \; v_i = 0 \quad (4)$$

Using eqs. 1 and 2 we only allow for a strictly positive flux $\mathbf{v}$ that obeys the steady-state condition. Eq. 3 forces the solution to have a positive flux through a given reaction which can be the outflow of the product of interest, i.e., if a solution exists, $\mathbf{v}$ produces the metabolite of interest. Eq. 4 guarantees that we only find a flux that does not use the reactions in K that are knocked out. By minimizing the overall flux and solving the linear program using the simplex algorithm [Sch98] we achieve that $\mathbf{v}$ corresponds to an EFM. This property of $\mathbf{v}$ will be shown in the following.

The solution space of the steady-state and the irreversibility condition (eqs. 1 and 2) in the space of possible fluxes $\mathbb{R}^n$ corresponds to a convex polyhedral cone $\mathcal{P}$ [GK04]. Since, we split reversible reactions, the extreme rays or spanning vectors of $\mathcal{P}$ correspond to the EFMs of the system. Furthermore, a knockout of a reaction only leads to the disappearance of some EFMs [SDF99]. Thus, for every K chosen, the cone is still spanned by EFMs and eq. 4 does not impact the property of the spanning vectors of $\mathcal{P}$ of being EFMs. Furthermore, eq. 3 cuts $\mathcal{P}$ with a hyperplane at $v_\mu = 1$ (Figure 1C). Since $\mathcal{P}$ is unbounded the edges of the solution space of eqs. 1 - 3 correspond to the intersection points between the EFMs defined by eqs. 1 as well as 2 and the hyperplane defined by $v_\mu = 1$. These points can each be written as the corresponding EFM multiplied with a scaling-factor. From linear programming it is known that the simplex algorithm used to solve such problems always returns a solution that can be found at the edges of the solution space [Sch98]. Thus, using the simplex algorithm and minimizing the objective function subject to eqs. 1 - 4 will always return an EFM.

In principle, the described linear program can find all EFMs by testing every possible set of knocked out reactions K. However, this is computationally inefficient and thus we will next outline an algorithm that allows to explore the space of EFMs more efficiently.

2.2 Genetic Algorithm

The aim of the GA is to test different sets of reactions to be knocked out in order to find all EFMs. Each such set of reactions corresponds to an individual. Each individual is represented by a binary genome $\mathbf{G}$ of length n, i.e., the number of reactions in the system. $G_i = 1$ indicates that reaction i can be used by that organism and $G_i = 0$ that this reaction is knocked out. From each genome an EFM can be derived by mapping $\mathbf{G}$ to the

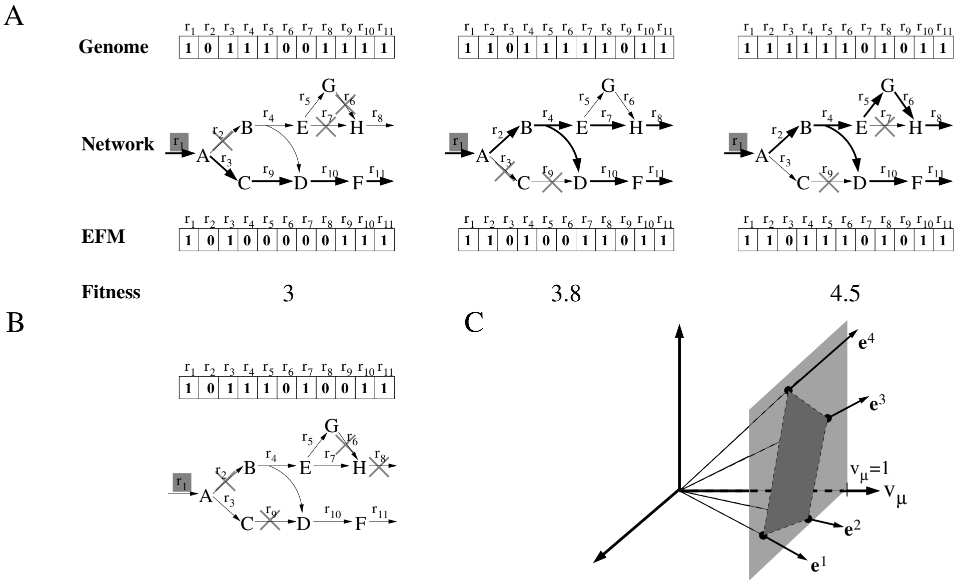

Figure 1: Scheme of the computation of EFMs. **A** Viable individuals. The target reaction $\mu = r_1$ is shaded in gray. In the upper row the genome of each individual is given. The second row indicates the reactions knocked out in the model and the third row the EFM obtained from the linear program. Even though the EFM of the third individual is also a valid EFM satisfying eqs. 1 - 4 for the second individual it is not minimal since the sum of fluxes is higher. The fourth row gives the fitness of each individual for a population containing the three depicted genomes. **B** Individual for which no EFM can be found. **C** Three-dimensional solution space of eqs. 1 - 3 for 3 reactions (not shown). The solution space is defined by the intersection of the solution space of eqs. 1 and 2, spanned by the EFMs $\mathbf{e}^1$ to $\mathbf{e}^4$, and the half-space defined by eq. 2. Optimal solutions of the linear program can always be found in the edges of the solution space (black circles).

set of knocked out reactions K and solving the linear program described in the previous section. Thus, we can obtain an EFM associated to an individual (Figure 1A and 1B). Solving the linear program described in the last section we can only find a single EFM. In consequence, by specifying different sets of reactions that should not be used by an EFM, that is, by knocking them out, we can sample EFMs.

Central for each GA is the definition of a fitness function that returns a numerical value indicating the quality of an individual. In contrast to other approaches the aim of the GA described here is not to find an individual that is optimal in some sense, but to detect all possible EFMs in a metabolic network. Thus, we attribute higher fitness to individuals whose associated EFMs use reactions which are not frequent in the EFMs of the population. Given a population $\mathbf{G}^1$, ..., $\mathbf{G}^s$ of individuals and the associated EFMs $\mathbf{e}^1$, ..., $\mathbf{e}^s$ the

fitness $f(\mathbf{G}^k)$ of a particular individual $\mathbf{G}^k$ is defined by

$$f(\mathbf{G}^k) = \sum_{i=1}^{n} \frac{sign(e_i^k)}{\sum_{j=1}^{s} sign(e_i^j)} \quad (5)$$

with $sign(x)$ returning '1' if x is non-zero, i.e., if a reaction is used, and '0' otherwise.

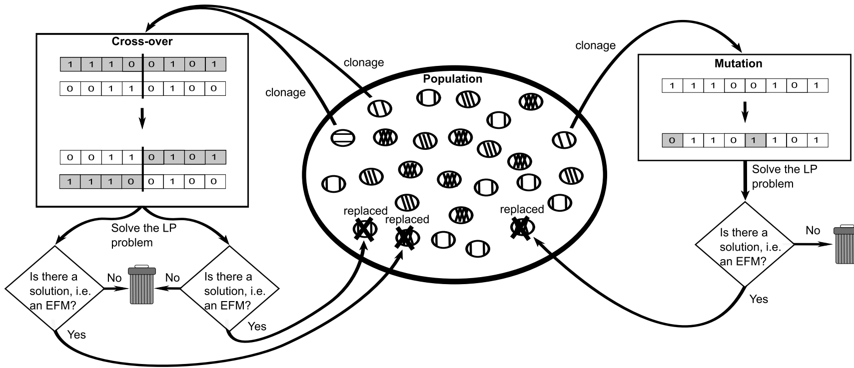

Figure 2: Setup of the GA. Individuals from the population are cloned and subsequently mutated or recombined. Afterward the viability of the individuals is tested by determining an EFM contained in them that uses the target reaction μ. If no such EFM is found, the individual is discarded. Otherwise it is reinserted by replacing a randomly chosen individual in the population.

For the GA we use the setup depicted in Figure 2. We assume a constant population size of s individuals and use two genetic operators: mutation and recombination. Before selecting individuals from the population it is decided whether a mutation or recombination should be performed. With probability $1 - p_{rec}$ one individual is mutated and with probability p_{rec} two individuals are recombined. To apply these operators, individuals are cloned from the population. By cloning we mean that an individual is selected and its genome copied creating a new individual. Thus, the original individual persists in the population. Given the individual fitness values $f(\mathbf{G}^1)$, ..., $f(\mathbf{G}^s)$ the probability of individual k to be cloned is proportional to its fitness:

$$P(\text{“Individual } k \text{ is cloned”}) = \frac{f(\mathbf{G}^k)}{\sum_{i=1}^{s} f(\mathbf{G}^i)} \quad (6)$$

During a mutation event, after cloning a single individual, each position in its genome is mutated with probability p_{mut}. Subsequently, it is tested whether the new individual is “viable” by determining the EFM associated to it. If such an EFM is found, the individual is re-inserted into the population by replacing a randomly chosen individual. Furthermore, the EFM that has been found is compared to all previously found EFMs and is added

to this set if it has not already been detected. If two individuals are recombined, they are first cloned and then the genomes are interchanged starting from a random position. Subsequently it is tested for both if they are viable, and, if this is the case, they are re-inserted replacing two randomly chosen individuals of the population. Thus, EFMs are detected as a side product of checking the viability of new individuals.

An important advantage of GAs is that they can be easily parallelized by the use of separate threads that mutate, recombine, and test individuals. Thus, the multi-processor architecture of modern desktop PCs is fully exploited.

3 Results

We applied our method to compute EFMs producing lysine, threonine, and arginine in two metabolic networks of *Escherichia coli* and one metabolic network of *Corynebacterium glutamicum*. Especially for the industrial production of lysine *C. glutamicum* is of importance [WBE06]. The first network of *E. coli* has been presented in [BWvK$^+$08]. It comprises 220 reactions and models amino acid metabolism. This network has the advantage that we can compute EFMs using Metatool [vKS06]. The second network represents a genome-scale model of *E. coli* metabolism and comprises 3558 reactions [FHR$^+$07]. The model of *C. glutamicum* contains 641 reactions and has been presented in [KN09]. In order to avoid side-pathways used for the balancing of co-factors and to provide an input medium we set the metabolites ammonium, AMP, ATP, CO_2, coenzyme A, glucose, NAD^+, NADH, $NADP^+$, NADPH, oxygen, protons, and inorganic ions to external status. As parameters for the computation we used a population size of $s = 100$ individuals, a mutation rate of $p_{mut} = 0.01$ per reaction and a probability of $p_{rec} = 0.3$ for recombination events. Computations were performed on an Intel® Core™2 Quad Q9300 machine with 4096 MB RAM running Linux Kernel 2.6.25 and Java Hotspot VM version 1.6.0. *Clp* version 1.0.6 from the COIN-OR project [LH03] has been used to solve the linear programs. An overview on the results is given in Table 1 and Figure 3.

As a first benchmark we tested to what extend our method can recover EFMs in a system in which they are already known. The model of [BWvK$^+$08] contains 3436 EFMs producing lysine, 444 EFMs producing threonine and 27450 EFMs producing arginine. We found all EFMs producing threonine and lysine after 491 s and 4821 s, respectively. For arginine we recovered 95.6% of all EFMs after a running time of 7200 s. In comparison, Metatool 5.1 took only 61 s to find all 65840 EFMs. However, a direct run-time comparison even to the currently fastest algorithm for the enumeration of EFMs presented in [TS08] does not bear much meaning since these methods in general only return the entire set of EFMs. This is not practicable in genome-scale networks since the number of EFMs exceeds by far current limitations in memory and processing power [YTP07]. An interesting behavior of the GA can be observed from these experiments. First, the time-course shows a kind of saturation when having found most of the EFMs. Furthermore, we observe phases in which only few new EFMs are found and sudden jumps in which the number increases rapidly as in the case of threonine in the model of amino acid metabolism at $t = 320$ s. While this particular behavior is also observable in the case of lysine in the model of *C. glutamicum*, a saturation

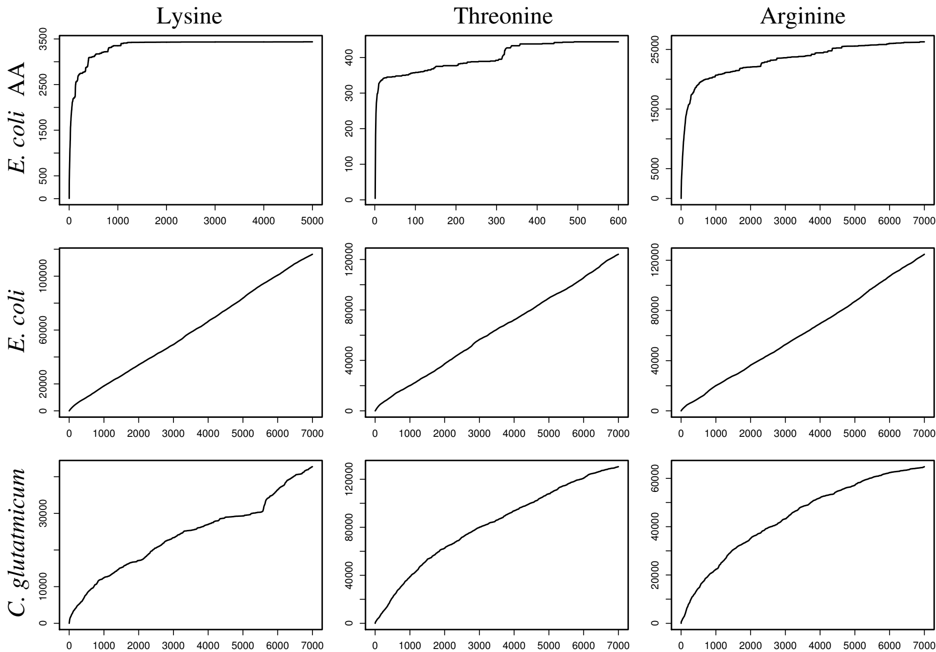

Figure 3: Time-course of the determination of EFMs for the three test-models: *E. coli* AA, [BWvK+08]; *E. coli*, [FHR+07]; *C. glutamicum*, [KN09]. The X-axis gives the running time in seconds and the Y-axis the number of EFMs found.

can be observed for the two other amino acids. In conjunction with the limited size of this model these results indicate that our method has already discovered a significant portion of all EFMs producing the three amino acids. In contrast, in the genome-scale system of *E. coli* we observe an almost linear increase in the number of EFMs without any saturation indicating that the number of EFMs existing in this model is much larger than the number already sampled.

Furthermore we tested the time required for the computation of 2000 EFMs in all models. We found the influence of network size on the running time much smaller than expected. Thus, it took on average 26.3 s to find 2000 EFMs in the model of *C. glutamicum* and 43 s in the genome-scale model of *E. coli* although both models differ more than five-fold in the number of reactions. This behaviour might be attributed to the simplex algorithm used to solve the linear programing problem described in Section 2.1. Since we are iteratively solving very similar problems and the simplex algorithm can start from a previous solution after changing some constraints, new solutions can be found very fast without need to consider the entire problem, but only a specific sub-part for which constraints were changed.

Another interesting aspect of the detected EFMs arises from the part of the network that can be used for the production of particular amino acids. For this analysis we combined

Model	# Rea.	AA	# EFMs	# Min.	CS	2000 EFMs
E. coli	220	Lysine	3436	16	94	95 s
AA metabolism		Threonine*	444	11	67	839 s
[BWvK+08]		Arginine	26276	18	95	8 s
E. coli	3558	Lysine	118598	29	1826	49 s
Genome-scale		Threonine	126491	26	2084	38 s
[FHR+07]		Arginine	127988	37	1895	42 s
C. glutamicum	641	Lysine	43115	23	240	28 s
Genome-scale		Threonine	131346	24	245	22 s
[KN09]		Arginine	65236	35	246	29 s

Table 1: Overview on computed EFMs. For each of the three test-models (number of reactions in the second column) the GA has been used to determine EFMs for the production of lysine, threonine and arginine (third column). The fourth column gives the number of EFMs detected after a running time of 7200 s. The fifth and sixth column indicate the minimal length of a detected EFM for the production of the given amino acid and the total number of different reactions used by all EFMs. The last column indicates the time required for the computation of 2000 EFMs averaged over 10 runs. In the case marked with *, the system only contained 444 EFMs.

all the computed EFMs for each test-case and determined the number of reactions used (Table 1). Furthermore, we determined the minimal number of reactions used by an EFM for the production of a given amino acid (Table 1). Combining all EFMs, the part of the metabolic network that can be used for the production of each amino acid varies in between 31% to 59% of the total network size. In consequence, there seems to be a great versatility in potential pathways. However, this versatility can be mostly attributed to the side-products of amino acid biosynthesis. For instance, in the production of lysine succinyl-CoA is converted to succinate. There are two ways of balancing succinyl-CoA and succinate. Either succinyl-CoA is additionally produced from the input medium and succinate is disposed through some other pathway, or succinate is reconverted into succinyl-CoA. Hence, we see a combinatorial explosion since the basic route producing lysine can be combined, on the one hand, with every pathway producing succinyl-CoA and consuming succinate. On the other hand this route can be combined with every possible pathway converting succinate into succinyl-CoA. This is also apparent from an analysis of the 64699 EFMs producing amino acids in the model of [BWvK+08]. Here we found that 35% of the EFMs do not only produce a single, but several amino acids. These additional amino acids can serve as sinks for side-metabolites.

4 Discussion

In this work we have outlined a new approach based on a genetic algorithm (GA) that allows to determine EFMs using a specific reaction in genome-scale metabolic networks. Previous methods that are based on searching paths in a graph representation of a metabolic network only guarantee to find connected routes while EFMs correspond to routes of actual metabolic conversions [dFSKF09]. Computing EFMs in a network in which they also

can be enumerated using deterministic algorithms we demonstrated that even large sets of EFMs can be recovered almost entirely. Comparing the time-course of the number of EFMs enumerated between a small and two large networks we concluded that we had already found a significant portion of all EFMs in a genome-scale model of *C. glutamicum* but only a small portion in a much larger model of *E. coli*. Analyzing the parts of the metabolic network which can be used by EFMs we found that they corresponded to 31% to 59% of the entire network even though individual pathways are usually much shorter. We attributed this result to the large variability of pathways that can be used to balance side-metabolites of amino acid biosynthetic pathways.

There exist several alternative approaches that allow a similar analysis of pathways in genome-scale networks. They either decompose a large network into smaller subnetworks or consider the entire network. The former approaches bear the problem that they only consider a small network on the local scale and thus they can contain artificial pathways that do not appear on the scale of the entire system [KdFS09]. Among the latter approaches especially constrained based methods are of importance. Methods from this field that allow to perform a similar analysis are flux balance analysis (FBA, [VP94]), flux variability analysis (FVA, [MS03]), and stochastic sampling of the solution space of eqs. 1 - 3 with additional upper bounds on reaction fluxes [WFGP04]. However, FBA only returns a specific pathway optimizing a certain objective function [VP94] and flux variability analysis only determines the set of reactions that can take part in alternative optimal pathways, without allowing to identify these pathways [MS03]. Stochastic sampling in contrast is very similar to our approach, but returns solutions that lie within the solution space of eqs. 1 - 3. Thus, rather than EFMs fluxes that correspond to combinations of EFMs are returned.

Our method represents an important step towards the analysis of EFMs, and thus of pathways, in genome-scale metabolic networks. While we used a fitness function that selects for diversity one can think of other functions that can be used. Thus, it is of interest to analyze suboptimal EFMs for the production of some metabolite which are in a specific range of yield per mole of an input metabolite or fulfill additional criteria like the production of a certain side-metabolite. Furthermore, since EFMs correspond to the concept of minimal transition invariants (MTIs) in petri-nets [SPM$^+$00, KH08], our approach can also be useful to find MTIs in large petri-nets.

5 Acknowledgments

Financial support from the German Ministry for Research and Education (BMBF) to C.K. and J.B. (grant FKZ 0315285E and FKZ 0313078E), and from the Fundação Calouste Gulbenkian, Fundação para a Ciência e a Tecnologia (FCT) and Siemens SA Portugal (PhD grant SFRH/BD/32961/2006) to L.F.F. is gratefully acknowledged.

References

[BK08] T. Blum and O. Kohlbacher. Using atom mapping rules for an improved detection of relevant routes in weighted metabolic networks. *J Comput Biol*, 15(6):565–576, 2008.

[BWvK+08] J. Behre, T. Wilhelm, A. von Kamp, E. Ruppin, and S. Schuster. Structural robustness of metabolic networks with respect to multiple knockouts. *J Theor Biol*, 252(3):433–441, Jun 2008.

[CCWvH06] D. Croes, F. Couche, S. J. Wodak, and J. van Helden. Inferring meaningful pathways in weighted metabolic networks. *J Mol Biol*, 356(1):222–236, Feb 2006.

[CS04] R. Carlson and F. Srienc. Fundamental *Escherichia coli* biochemical pathways for biomass and energy production: Identification of reactions. *Biotechnol Bioeng*, 85(1):1–19, Jan 2004.

[dFSKF09] L. F. de Figueiredo, S. Schuster, C. Kaleta, and D. A. Fell. Can sugars be produced from fatty acids? A test case for pathway analysis tools. *Bioinformatics*, 25(1):152–158, Jan 2009.

[FHR+07] A. M. Feist, C. S. Henry, J. L. Reed, M. Krummenacker, A. R. Joyce, P. D. Karp, L. J. Broadbelt, V. Hatzimanikatis, and B. Ø Palsson. A genome-scale metabolic reconstruction for *Escherichia coli* K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information. *Mol Syst Biol*, 3:121, 2007.

[FP08] A. M. Feist and B. Ø. Palsson. The growing scope of applications of genome-scale metabolic reconstructions using *Escherichia coli*. *Nat Biotechnol*, 26(6):659–667, Jun 2008.

[GK04] J. Gagneur and S. Klamt. Computation of elementary modes: A unifying framework and the new binary approach. *BMC Bioinformatics*, 5:175, 2004.

[KdFS09] C. Kaleta, L. F. de Figueiredo, and S. Schuster. Can the whole be less than the sum of its parts? Pathway analysis in genome-scale metabolic networks using elementary flux patterns. *Genome Res Res*, 2009. Accepted.

[KH08] I. Koch and M. Heiner. *Biological Network Analysis*, chapter Petri Nets, pages 139 – 180. Wiley Book Series in Bioinformatics. Wiley & Sons, 2008.

[KN09] K. R. Kjeldsen and J. Nielsen. In silico genome-scale reconstruction and validation of the *Corynebacterium glutamicum* metabolic network. *Biotechnol Bioeng*, 102(2):583–597, Feb 2009.

[KS02] S. Klamt and J. Stelling. Combinatorial complexity of pathway analysis in metabolic networks. *Mol Biol Rep*, 29(1-2):233–236, 2002.

[LH03] R. Lougee-Heimer. The Common Optimization INterface for Operations Research: Promoting open-source software in the operations research community. *IBM J Res Dev*, 47(1):57–66, 2003.

[MS03] R. Mahadevan and C. H. Schilling. The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. *Metab Eng*, 5(4):264–276, Oct 2003.

[PB08] F. J. Planes and J. E. Beasley. A critical examination of stoichiometric and path-finding approaches to metabolic pathways. *Brief Bioinform*, 9(5):422–436, Sep 2008.

[PRFN05] K. R. Patil, I. Rocha, J. Förster, and J. Nielsen. Evolutionary programming as a platform for in silico metabolic engineering. *BMC Bioinformatics*, 6:308, 2005.

[RAS+05] S. A. Rahman, P. Advani, R. Schunk, R. Schrader, and D. Schomburg. Metabolic pathway analysis web service (Pathway Hunter Tool at CUBIC). *Bioinformatics*, 21(7):1189–1193, Apr 2005.

[Sch98] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, June 1998.

[SDF99] S. Schuster, T. Dandekar, and D. A. Fell. Detection of elementary flux modes in biochemical networks: A promising tool for pathway analysis and metabolic engineering. *Trends Biotechnol*, 17(2):53–60, Feb 1999.

[SKB+02] J. Stelling, S. Klamt, K. Bettenbrock, S. Schuster, and E. D. Gilles. Metabolic network structure determines key aspects of functionality and regulation. *Nature*, 420(6912):190–193, Nov 2002.

[SLP00] C. H. Schilling, D. Letscher, and B. Ø. Palsson. Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective. *J Theor Biol*, 203(3):229–248, Apr 2000.

[SPM+00] S. Schuster, T. Pfeiffer, F. Moldenhauer, I. Koch, and T. Dandekar. Structural analysis of metabolic networks: Elementary flux modes, analogy to Petri nets, and application to *Mycoplasma Pneumoniae*. In *German Conference on Bioinformatics*, pages 115–120, 2000.

[TS08] Marco Terzer and Jörg Stelling. Large-scale computation of elementary flux modes with bit pattern trees. *Bioinformatics*, 24(19):2229–2235, Oct 2008.

[TUS08] C. T. Trinh, P. Unrean, and F. Srienc. Minimal *Escherichia coli* cell for the most efficient production of ethanol from hexoses and pentoses. *Appl Environ Microbiol*, 74(12):3634–3643, Jun 2008.

[vKS06] A. von Kamp and S. Schuster. Metatool 5.0: Fast and flexible elementary modes analysis. *Bioinformatics*, 22(15):1930–1931, Aug 2006.

[VP94] A. Varma and B. Ø. Palsson. Stoichiometric flux balance models quantitatively predict growth and metabolic by-product secretion in wild-type *Escherichia coli* W3110. *Appl Environ Microbiol*, 60(10):3724–3731, Oct 1994.

[WBE06] V. F. Wendisch, M. Bott, and B. J. Eikmanns. Metabolic engineering of *Escherichia coli* and *Corynebacterium glutamicum* for biotechnological production of organic acids and amino acids. *Curr Opin Microbiol*, 9(3):268–274, Jun 2006.

[WFGP04] S. J. Wiback, I. Famili, H. J. Greenberg, and B. . Palsson. Monte Carlo sampling can be used to determine the size and shape of the steady-state flux space. *J Theor Biol*, 228(4):437–447, Jun 2004.

[YTP07] M. Yeung, I. Thiele, and B. Ø. Palsson. Estimation of the number of extreme pathways for metabolic networks. *BMC Bioinformatics*, 8(1):363, 2007.