

Software

Open Access

## SNA – a toolbox for the stoichiometric analysis of metabolic networks

Robert Urbanczik\*

Address: Institute of Pharmacology, University of Bern, Friedbühlstr. 49, CH-3010 Bern, Switzerland

Email: Robert Urbanczik\* - [urbanczik@physik.uni-wuerzburg.de](mailto:urbanczik@physik.uni-wuerzburg.de)

\* Corresponding author

Published: 13 March 2006

Received: 24 November 2005

*BMC Bioinformatics* 2006, **7**:129 doi:10.1186/1471-2105-7-129

Accepted: 13 March 2006

This article is available from: <http://www.biomedcentral.com/1471-2105/7/129>

© 2006 Urbanczik; licensee BioMed Central Ltd.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/2.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

### Abstract

**Background:** Despite recent algorithmic and conceptual progress, the stoichiometric network analysis of large metabolic models remains a computationally challenging problem.

**Results:** SNA is a interactive, high performance toolbox for analysing the possible steady state behaviour of metabolic networks by computing the generating and elementary vectors of their flux and conversions cones. It also supports analysing the steady states by linear programming. The toolbox is implemented mainly in Mathematica and returns numerically exact results. It is available under an open source license from: [http://bioinformatics.org/project/?group\\_id=546](http://bioinformatics.org/project/?group_id=546).

**Conclusion:** Thanks to its performance and modular design, SNA is demonstrably useful in analysing genome scale metabolic networks. Further, the integration into Mathematica provides a very flexible environment for the subsequent analysis and interpretation of the results.

### Background

Advances in genomics have enabled the large scale reconstruction of biochemical reaction networks where the main information provided concerns the stoichiometry and reversibility of the involved reactions. While such models disregard the kinetics of the reactions, biologically meaningful predictions can nevertheless be obtained by analysing the stoichiometrically viable steady states of the network [1]. The aim of SNA is to provide a comprehensive interactive environment for such stoichiometric network analysis.

In mathematical terms the stoichiometrically viable steady states form a convex polyhedral cone which is called the flux cone. Recently [2] it has been pointed out that instead of analysing the full flux cone, one can consider a simpler object, the conversion cone. This amounts to giving a black box description of the metabolism which

only takes into account the consumption and production of external compounds. So, the conversion cone describes the overall reactions, (in the sense of [3]) which can be effected in steady state between the external metabolites of the network, disregarding the internal mechanism.

Linear Programming, as in Flux Balance Analysis [4], can be used to search for specific points in these cones. But executing a linear program will just return a single point in the cone, providing only limited information about the whole range of possible steady states the biochemical reaction network can assume. More complete descriptions are obtained by computing a minimal generating set for each cone, i.e. a minimal subset such that each vector in the cone can be represented as a linear combination, with non-negative scalar coefficients, of the subset vectors. If all reactions in the network are irreversible, the flux and the conversion cone are pointed and computing their mini-

mal generating sets amounts to listing the edges of the two cones. In the presence of reversible reactions, however, the description provided by a minimal generating set is less satisfactory. On one hand the cones then may no longer be pointed and many quite different minimal generating sets then exist. On the other hand, if there are reversible reactions, cancellation can occur when combining the vectors of a minimal generating set. So, from the fact that, e.g., each vector in the minimal generating set of the flux cone has a non-zero flow through a specific reversible reaction, one cannot conclude that this reaction must run in any steady state. Hence, in presence of reversible reactions, one may wish to precompute all possible cancellations. This leads to the problem of enumerating all the elementary vectors [2,5] of the flux and of the conversion cone. In the case of the flux cone, an elementary vector represents a minimal stoichiometrically viable pathway through the network and is often called an elementary mode [6].

By computing the set of elementary flux vectors a complete and very explicit description of the possible steady state behaviour is obtained and the answers to many questions about the metabolism, such as the maximal yield of some compound and the reactions essential for its synthesis are immediately found by inspecting this set. However, for large networks, enumerating all elementary fluxes may not be desirable, or even feasible, since their number can be very large. It is then often useful to first consider the conversion cone which is much simpler than the flux cone when many elementary fluxes are equivalent with regard to the consumed and produced compounds, differing only in the details of the internal synthesis. While this provides a less complete description of the metabolism than the flux cone, many interesting questions can nevertheless be answered by computing the elementary conversions. Cases in point are (a) the determination of minimal media [2] and (b) checking whether the metabolic model is consistent with an experimentally observed overall reaction and gauging the efficiency of the latter [7].

### Implementation

I shall only consider the architecture of the toolbox here since the main mathematical and computational concepts implemented in SNA have been described elsewhere. In particular the Nullspace algorithm used by SNA for calculating elementary fluxes, which runs significantly faster than previous procedures [6], is described in [8]. Further, the concept of a conversion cone is introduced in [2], together with the procedure for computing elementary conversions.

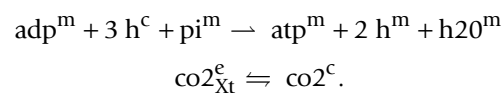
The user interface for SNA is Mathematica and the functionality described in the preceding section is provided by a Mathematica package called SNAsym. The matrix

level computational geometry routines underpinning the operation of SNAsym are delegated to the lower level package SNAmat. SNAmat may be of independent interest to people wishing to avoid using SNA proper and only interested in a computational engine for calculating elementary modes from a stoichiometry matrix.

In contrast to SNAsym, which is pure Mathematica code, the lower level package is a hybrid consisting of Mathematica code communicating with a binary program compiled from Matlab and C sources. The division of labour is that the Mathematica part of SNAmat does the arithmetic, whereas, for speed, the purely logical operations are done by the binary. An important motivation for this arises from the fact that in mathematical terms only rational operations are required in the network analysis tasks implemented by SNA, i.e. no irrational functions such as root taking are needed. Hence, if the stoichiometric factors in the networks are represented by fractions the entire analysis can only lead to numbers which are rational as well and thus have an exact numerical representation. SNA exploits this, by doing all of the arithmetic in Mathematica and using the built-in functionality of this software, to always provide numerically exact results. Due to the division of labour between Mathematica and binary code, SNA is nevertheless fast enough to handle large scale problems. For instance, it takes only 7 minutes to calculate the  $5 \times 10^5$  elementary fluxes for the model of the central carbon metabolism of *E. coli* [9] on a state of the art PC.

### Results

Biochemical reactions are represented in SNA by Mathematica expressions such as



Each metabolite has a body which is a string denoting the chemical species (e.g. adp) and a superscript designating the compartment. Above the strings 'm', 'c', and 'e' are used for mitochondrial, cytosolic and, respectively, extracellular metabolites. In addition each metabolite has a rôle, the default rôle being internal while other rôles are specified by a subscript. The subscript *Xt* denotes an external metabolite which can be used both as an input and an output to the network, whereas metabolites with subscript *Xtin* or *Xtout* are restricted to being inputs or, respectively, outputs. Note that rôle and compartment are treated as quite distinct concepts. The reason is that in a study focusing on just an organelle, such as mitochondria, one might wish to treat all cytosolic metabolites as external.

We first load SNA

```
# << "~/SNA/mathcode/SNAsym.m";
```

and then the network (assigning it to the variable PM).

```
# PM = << "PyrMet.m";
```

The following displays the reactions in the network together with their names.

```
# trpairs@PM // TableForm
```

R[BOBdehyd]	NADH + ACAC <sub>xtout</sub> ⇌ NAD + BOB <sub>xtout</sub>
R[AcoHyd]	CIT <sub>xtout</sub> ⇌ ICI <sub>xtout</sub>
R[FUMarase]	FUM <sub>xtout</sub> ⇌ MAL <sub>xtout</sub>
R[MALdehyd]	NAD + MAL <sub>xtout</sub> ⇌ NADH + OAA <sub>xtout</sub>
R[ACACtran]	2 AcCoA → 2 CoA + ACAC <sub>xtout</sub>
R[ATPase]	ATP → ADP
R[OxPhos2]	2 ADP + FADH → 2 ATP + FAD
R[NucKin]	ADP + GTP → ATP + GDP
R[OxPhos1]	3 ADP + NADH → 3 ATP + NAD
R[SuCoAsyn]	GDP + SuCoA → CoA + GTP + SUC <sub>xtout</sub>
R[PYRcarbo]	ATP + PYR <sub>xtin</sub> → ADP + OAA <sub>xtout</sub>
R[PYRdehyd]	CoA + NAD + PYR <sub>xtin</sub> → AcCoA + NADH
R[AKGdehyd]	CoA + NAD + AKG <sub>xtout</sub> → NADH + SuCoA
R[ICIdehyd]	NAD + ICI <sub>xtout</sub> → NADH + AKG <sub>xtout</sub>
R[CITsyn]	AcCoA + OAA <sub>xtout</sub> → CoA + CIT <sub>xtout</sub>
R[SUCdehyd]	FAD + SUC <sub>xtout</sub> → FADH + FUM <sub>xtout</sub>

We next enumerate the elementary fluxes.

```
# {felvs, nrev} = symfluxelvs[PM];
```

```
# nrev
```

Zero fluxes are reversible, the number of elementary fluxes is:

```
# felvs // Length
```

```
# 33
```

And the first one is:

```
# felvs // First
```

```
# 2 R[CITsyn] - R[MALdehyd] +
  R[OxPhos1] + 3 R[PYRcarbo] +
  2 R[PYRdehyd] + 5 Rx[0 → PYRxtin] +
  2 Rx[CITxtout → 0] + Rx[MALxtout → 0]
```

Note the sparse vector notation using the reaction names, and that SNA has automatically added the exchange reactions for the external metabolites.

Next, we obtain an overview of the overall reactions by calculating the elementary vectors of the conversion cone.

```
# conversionelvs[PM] // TableForm
```

PYR <sub>xtin</sub> → 0
2 PYR <sub>xtin</sub> → ACAC <sub>xtout</sub>
2 PYR <sub>xtin</sub> → AKG <sub>xtout</sub>
2 PYR <sub>xtin</sub> → BOB <sub>xtout</sub>
2 PYR <sub>xtin</sub> → CIT <sub>xtout</sub>
2 PYR <sub>xtin</sub> → ICI <sub>xtout</sub>
2 PYR <sub>xtin</sub> → SUC <sub>xtout</sub>
4 PYR <sub>xtin</sub> → CIT <sub>xtout</sub> + 2 OAA <sub>xtout</sub>
4 PYR <sub>xtin</sub> → ICI <sub>xtout</sub> + 2 OAA <sub>xtout</sub>
5 PYR <sub>xtin</sub> → 2 CIT <sub>xtout</sub> + FUM <sub>xtout</sub>
5 PYR <sub>xtin</sub> → FUM <sub>xtout</sub> + 2 ICI <sub>xtout</sub>
5 PYR <sub>xtin</sub> → 2 CIT <sub>xtout</sub> + MAL <sub>xtout</sub>
5 PYR <sub>xtin</sub> → 2 ICI <sub>xtout</sub> + MAL <sub>xtout</sub>
5 PYR <sub>xtin</sub> → BOB <sub>xtout</sub> + 3 OAA <sub>xtout</sub>
7 PYR <sub>xtin</sub> → 2 ACAC <sub>xtout</sub> + 3 FUM <sub>xtout</sub>
7 PYR <sub>xtin</sub> → 2 ACAC <sub>xtout</sub> + 3 MAL <sub>xtout</sub>
7 PYR <sub>xtin</sub> → AKG <sub>xtout</sub> + 5 OAA <sub>xtout</sub>
8 PYR <sub>xtin</sub> → ACAC <sub>xtout</sub> + 6 OAA <sub>xtout</sub>
11 PYR <sub>xtin</sub> → 4 BOB <sub>xtout</sub> + 3 FUM <sub>xtout</sub>
11 PYR <sub>xtin</sub> → 4 BOB <sub>xtout</sub> + 3 MAL <sub>xtout</sub>
11 PYR <sub>xtin</sub> → 9 OAA <sub>xtout</sub> + SUC <sub>xtout</sub>
13 PYR <sub>xtin</sub> → 4 AKG <sub>xtout</sub> + 5 FUM <sub>xtout</sub>
13 PYR <sub>xtin</sub> → 4 AKG <sub>xtout</sub> + 5 MAL <sub>xtout</sub>
16 PYR <sub>xtin</sub> → 15 OAA <sub>xtout</sub>
17 PYR <sub>xtin</sub> → 9 FUM <sub>xtout</sub> + 4 SUC <sub>xtout</sub>
17 PYR <sub>xtin</sub> → 9 MAL <sub>xtout</sub> + 4 SUC <sub>xtout</sub>
19 PYR <sub>xtin</sub> → 15 FUM <sub>xtout</sub>
19 PYR <sub>xtin</sub> → 15 MAL <sub>xtout</sub>

For flux balance analysis, to speed up the linear programming, an optimized representation of the network is calculated first.

```
# PMfba = FBAPrep[PM];
```

We next define a constraint, limiting Pyruvate uptake to lie between 0 and 1.

```
# constr = {
  { Rx[0 → "PYR"xtin], {0, 1}
};
```

The following maximizes Malate production subject to the constraint:

```
# {opt, flux} =
  FBA[PMfba,
    Rx["MAL"xtout → 0],
    constr];
```

```
# opt
15
19
```

At most 15 units of Malate can be produced per 19 units of consumed Pyruvate, as one would expect from the conversion cone analysis (cf. the last elementary conversion).

Finally, a flux realizing this production rate is:

```
# flux
4 R[AcoHyd] + 4 R[AKGdehyd] +
19 19
4 R[CITsyn] + 4 R[FUMarase] +
19 19
4 R[ICIdehyd] - 11 R[MALdehyd] +
19 19
4 R[NucKin] + R[OxPhos1] +
19 19
4 R[OxPhos2] + 15 R[PYRcarbo] +
19 19
4 R[PYRdehyd] + 4 R[SUCdehyd] +
19 19
4 R[SuCoAsyn] + Rx[0 → PYRxtin] +
19
15 Rx[MALxtout → 0]
```

Figure 1

A Mathematica session using SNA to analyse a model of the Pyruvate metabolism in rat liver mitochondria. The model is adapted from [7] where the full metabolite names are given, as well as the explanation for the unusual conversion  $PYR_{xtin} \rightarrow 0$ .

**Functionality**

Based on a list of such reactions and an appropriate naming scheme for the reactions, SNA is used to construct the abstract data type *mnet* (metabolic net). The toolbox provides a large set of functions for the basic manipulation of *mnet*'s: e.g. joining them, extracting subnetworks and changing the rôle of the metabolites in the network. Less basic functions include, determining which reactions are

feasible, i.e. admit a non-zero steady state flow, or simplifying the network in a way that preserves e.g. the conversion cone.

But, of course, the main functionality of SNA is related to the tasks mentioned in the first section. Besides supporting flux balance analysis, the toolbox provides functions analysing the flux or the conversion cone by calculating a

minimal generating set or enumerating all the elementary vectors. A session, demonstrating some of the functions of SNA, is shown in Fig. 1.

### Documentation

The main documentation for SNA consists of 9 tutorial Mathematica notebooks. After a basic introduction to the toolbox, the tutorials quickly move on to show how SNA can be used to analyse quite large metabolic networks, considering as examples models of the human cardiac mitochondria [10], the central carbon mechanism of *E. coli* [9] and, finally, a genome-scale model of *S. cerevisiae* [11]. In addition, two of the tutorials demonstrate how metabolic networks can be imported into SNA from external formats. The first shows how to construct an *mnet* given a table of numbers representing a stoichiometry matrix. The second shows how to parse textual representations of metabolic networks such as the ones used by the Palsson group. e.g. [10,11]. Further, SBML models can be read into Mathematica using the MathSBML software developed by B.E. Shapiro [12].

### Conclusion

Currently, the most widely used program for calculating elementary flux vectors probably is Metatool [13]. The latest version of Metatool (5.0) uses the same basic approach [8] as SNA, and the two are also quite similar in terms of required computing time. However, in contrast to Metatool, SNA does not rely on machine precision and always produces numerically exact results. This can be of advantage when analysing large metabolic networks where the stoichiometric factors can span many orders of magnitude, e.g. 7 orders of magnitude for the above mentioned *S. cerevisiae*. Further, compared to Metatool, SNA has as much expanded functionality and, in particular, provides functions for calculating elementary conversions. Finally, the design of SNA is very modular. While this may make the learning curve steeper, it provides the flexibility required to tackle the computational challenge still posed by the analysis of large metabolic networks.

### Availability and requirements

SNA is distributed under an open source license and can be downloaded from [http://bioinformatics.org/project/?group\\_id=546](http://bioinformatics.org/project/?group_id=546). The toolbox runs under PC-Linux and additionally requires Mathematica (5.0 or higher). Further, if one wants to recompile the binary program mentioned in the implementation section, Matlab 6.5 is needed.

### Acknowledgements

It is a pleasure to acknowledge helpful discussion with Jörg Stucki and Clemens Wagner.

### References

1. Price N, Reed J, Palsson B: **Genome-scale Models of Microbial Cells: Evaluating the consequences of constraints.** *Nat Rev Microbiol* 2004, **2**:886-897.
2. Urbanczik R, Wagner C: **Functional stoichiometric analysis of metabolic networks.** *Bioinformatics* 2005, **21**:4176-4180.
3. Happel J, Sellers P: **The characterization of complex systems of chemical reactions.** *Chem Eng Comm* 1989, **83**:221-240.
4. Ibarra R, Edwards J, Palsson B: **Escherichia coli K-12 Undergoes Adaptive Evolution to Achieve in silico Predicted Optimal Growth.** *Nature* 2002, **420**:186-189.
5. Rockafellar R: *Convex Analysis* Princeton: Princeton University Press; 1970.
6. Schuster S, Fell D, Dandekar T: **A General Definition of Metabolic Pathways Useful for Systematic Organisation and Analysis of Complex Metabolic Pathways.** *Nature Biotech* 2000, **18**:326-332.
7. Stucki J, Urbanczik R: **Pyruvate metabolism in rat liver mitochondria: what is optimized at steady state?** *FEBS Journal* 2005, in print
8. Urbanczik R, Wagner C: **An improved algorithm for stoichiometric network analysis: theory and applications.** *Bioinformatics* 2005, **21**:1203-1210.
9. Stelling J, Klamt S, Bettenbrock K, Schuster S, Gilles E: **Metabolic Network Structure determines Key Aspects of Functionality and Regulation.** *Nature* 2002, **420**:190-193.
10. Vo T, Greenberg H, Palsson B: **Reconstruction and functional characterization of the human mitochondrial metabolic network based on proteomic and biochemical data.** *J Biol Chem* 2004, **279**:39532-39540.
11. Duarte N, Herrgard M, Palsson B: **Reconstruction and Validation of Saccharomyces cerevisiae iND750, a Fully Compartmentalized Genome-scale Metabolic Model.** *Genome Research* 2004, **14**:1298-1309.
12. Shapiro B: **MathSBML.** [<http://sbml.org/software/mathsbml/>].
13. Pfeiffer T, et al.: **METATOOL: For Studying Metabolic Networks.** *Bioinformatics* 1999, **15**:251-257.

Publish with **BioMed Central** and every scientist can read your work free of charge

"BioMed Central will be the most significant development for disseminating the results of biomedical research in our lifetime."

Sir Paul Nurse, Cancer Research UK

Your research papers will be:

- available free of charge to the entire biomedical community
- peer reviewed and published immediately upon acceptance
- cited in PubMed and archived on PubMed Central
- yours — you keep the copyright

Submit your manuscript here:  
[http://www.biomedcentral.com/info/publishing\\_adv.asp](http://www.biomedcentral.com/info/publishing_adv.asp)

