

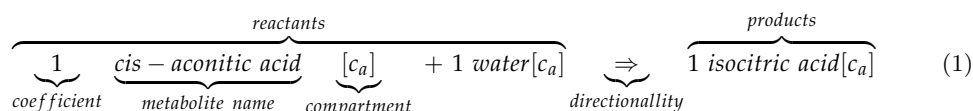
minval: An R package for MINimal VALidation of stoichiometric reactions

by Daniel Osorio, Janneth Gonzalez and Andres Pinzon-Velasco

Abstract A genome-scale metabolic reconstruction is a compilation of all stoichiometric reactions that can describe the entire cellular metabolism of an organism, and they have become an indispensable tool for our understanding of biological phenomena, covering fields that range from systems biology to bioengineering. Interrogation of metabolic reconstructions are generally carried through Flux Balance Analysis, an optimization method in which the biological sense of the optimal solution is highly sensitive to thermodynamic unbalance, caused by the presence of stoichiometric reactions whose compounds are not produced or consumed in any other reaction (orphan metabolites) and by mass unbalance. The **minval** package was designed as a tool to identify orphan metabolites and evaluate the mass and charge balance of stoichiometric reactions. The package also includes functions to characterize and write models in TSV and SBML formats, extract all reactants, products, metabolite names and compartments from a metabolic reconstruction.

Introduction

A chemical reaction is a process where a set of chemical compounds called *reactants* are transformed into others called *products* (?). The accepted way to represent a chemical reaction is called a *stoichiometric reaction*, where reactants are placed on the left and the products on the right separated by an arrow which indicates the direction of the reaction, as shown in equation ?? (?). In biochemistry, a set of chemical reactions that transform a substrate into a product, after several chemical transformations is called a metabolic pathway (?). The compilation of all stoichiometric reactions included in all metabolic pathways that can describe the entire cellular metabolism encoded in the genome of a particular organism is known as a *genome-scale metabolic reconstruction* (?) and has become an indispensable tool for studying metabolism of biological entities at the systems level (?).



Reconstruction of genome-scale metabolic models starts with a compilation of all known stoichiometric reactions for a given organism, according to the presence of enzyme-coding genes in its genome. The stoichiometric reactions catalyzed by these enzymes are usually downloaded from specialized databases such as KEGG (?), BioCyc (?), Reactome (?), BRENDA (?) or SMPDB (?). However, the downloaded stoichiometric reactions are not always mass-charge balanced and don't represent complete pathways as to construct a high-quality metabolic reconstruction (?). Therefore the identification and curation of these type of reactions is a time-consuming process which the researcher have to complete manually using available literature or experimental data (?).

Genome-scale metabolic reconstructions are usually interrogated through Flux Balance Analysis (FBA), an optimization method that allows us to understand the metabolic status of the cell, to improve the production capability of a desired product or make a rapid evaluation of cellular physiology at genomic-scale (?). Nevertheless, FBA method is high sensitive to thermodynamic unbalance, so in order to increase the validity of a biological extrapolation (i.e. an optimal solution) from a FBA analysis it is mandatory to avoid this type of unbalancing in mass conservation through all model reactions (?). Another drawback when determining the validity of a metabolic reconstruction is the presence of reactions with compounds that are not produced or consumed in any other reaction (dead ends), generally known as orphan metabolites (?). The presence of this type of metabolites can be problematic since they lead to an artificial cellular accumulation of metabolism products which generates a bias in the biological conclusions. Tracking these metabolites is also a time-consuming process, which most of the time has to be performed manually or partially automatized by in-house scripting. Given that typical genome-scale metabolic reconstructions account for hundreds or thousands of biochemical reactions, the manual curation of these models is a task that can lead to both, the introduction of new errors and to overlook some others.

Two of the most popular implementations of FBA analysis are **COBRA** (?) and **RAVEN** (?) which operate as tools under the commercial MATLAB® environment. On the R environment side **sybil** and **abcdeFBA** are the most common ones. **COBRA** and **RAVEN** include some functions

for mass and charge balance (`checkMassChargeBalance` and `getElementalBalance` respectively). These functions identify mass unbalanced reactions, based in the chemical formula or the IUPAC International Chemical Identifier (InChI) supplied manually by the user for each metabolite included in the genome-scale metabolic reconstruction.

With the aim of minimizing the manual introduction of thousands of chemical formulas in a genome-scale reconstruction as well as to avoid the sometimes limiting use of licensed software, we have developed the `minval` package. The `minval` package includes twelve functions designed to characterize, check and depurate metabolic reconstructions before its interrogation through Flux Balance Analysis (FBA).

Installation and functions

The `minval` package includes twelve functions and is available for download and installation from CRAN, the Comprehensive R Archive Network. To install and load it, just type:

```
> install.packages("minval")
> library(minval)
```

The `minval` package requires an R version 2.10 or higher. Development releases of the package are available on the GitHub repository <http://github.com/gibbslab/minval>.

Inputs and syntax

The functions included in `minval` package take as input a set of stoichiometric reactions where the metabolites should be separated by a plus symbol (+) between two blank spaces and may have just one stoichiometric coefficient before the name. The reactants should be separated from products by an arrow using the following symbol => for irreversible reactions and <=> for reversible reactions. The data can be loaded from traditional human-readable spreadsheets through other CRAN-available packages such as `gdata`, `readxl` or `xlsx`. To show the potential use of the `minval` package a human-readable model composed by a set of 19 stoichiometric reactions that represent an unbalanced model of the glycolysis process was included. To load it just type:

```
> glycolysisFile <- system.file("extdata", "glycolysisModel.csv", package = "minval")
> glycolysisModel <- read.csv(file = glycolysisFile,
+                             sep = '\t',
+                             stringsAsFactors = FALSE)
```

Glycolysis is the metabolic pathway that converts glucose ($C_6H_{12}O_6$), into two molecules of pyruvate ($CH_3COCOO^- + H^+$) through a sequence of ten enzyme-catalyzed reactions. Glycolysis occurs in most organisms in the cytosol of the cell and can be summarized as follows: 1 `alpha-D-Glucose[c]` + 2 `NAD+[c]` + 2 `ADP[c]` + 2 `Orthophosphate[c]` => 2 `Pyruvate[c]` + 2 `NADH[c]` + 2 `H+[c]` + 2 `ATP[c]` + 2 `H2O[c]`

Syntax Validation

The first step for a metabolic reconstruction validation is the check of their stoichiometric reactions. The `validateSyntax` function validate the syntax (Equation ??) of all reactions in a metabolic reconstruction for several FBA implementations (i.e. **COBRA** and **RAVEN**) and returns a boolean value 'TRUE' if the syntax is correct. Syntax validation is a critical step due valid stoichiometric reactions are required to write models in TSV or SBML formats.

```
> validateSyntax(reactionList = glycolysisModel$REACTION)

[1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[16] TRUE TRUE TRUE TRUE
```

Metabolic models

Metabolic models include additional to the stoichiometric reactions also another information that allows model and interrogates them through FBA, the generally associated information is:

Label	Description	Default Value
ID	A list of single character strings containing the reaction abbreviations, Entries in the field abbreviation are used as reaction ids, so they must be unique.	Mandatory
REACTION	A set of stoichiometric reactions with the previously described characteristics.	Mandatory
GPR	A set of genes joined by boolean operators as AND or OR, rules may be nested by parenthesis. GPR rules represent the relationship between genes to synthesize the required enzyme or enzymes to catalyze the stoichiometric reaction.	Optional (the column can be empty)
LOWER.BOUND	A list of numeric values containing the lower bounds of the reaction rates. If not set, zero is used for an irreversible reaction and -1000 for a reversible reaction.	-1000 or 0
UPPER.BOUND	A list of numeric values containing the upper bounds of the reaction rates. If not set, 1000 is used by default.	1000
OBJECTIVE	A list of numeric values containing objective values (0 or 1) for each reaction	0 or 1

The standard format to share and store biological processes such as metabolic models is the Systems Biology Markup Language (**SBML**) format. The [minval](#) package includes the `writeSBMLmod` function which is able to write models in SBML format as follows:

```
> writeSBMLmod(modelData = glycolysisModel,
+               modelID = "Glycolysis",
+               outputFile = "glycolysis.xml")
```

Metabolic models in SBML format can be readed through the `readSBMLmod` function of the [sybilSBML](#) R package:

```
> glycoModel <- sybilSBML::readSBMLmod("glycolysis.xml")
> glycoModel
```

```
model name:           Glycolysis
number of compartments 2
                     c
                     b
number of reactions:   19
number of metabolites: 18
number of unique genes: 22
objective function:    +1 R00200
```

After load the metabolic model, it can be interrogated through FBA using the `optimizeProb` function of the [sybil](#) R package. In this case, the reaction 'R00200' was set as the objective function. The 'R00200' reaction describes the production of pyruvate from phosphoenolpyruvate an alpha-D-Glucose derivate.

```
> sybil::optimizeProb(glycoModel)

solver:           glpkAPI
method:           simplex
algorithm:        fba
number of variables: 19
number of constraints: 18
return value of solver: solution process was successful
solution status:   solution is optimal
value of objective function (fba): 6.000000
value of objective function (model): 6.000000
```

The interrogated glycolysis model estimates a production of six molecules of pyruvate by each alpha-D-Glucose molecule, probably due a mass unbalance in their stoichiometric reactions. FBA methods are sensitive to thermodynamic (mass-charge) unbalance, so in order to achieve a valid biological extrapolation is mandatory to avoid this type of unbalancing in all model reactions.

Mass - Charge Balance Validation

The second step for a metabolic reconstruction validation is to check the stoichiometric reactions mass-charge balance. In a balanced stoichiometric reaction according to the *Lomonosov-Lavoisier*

law, the mass comprising the reactants should be the same mass present in the products. This process requires the use of a reference with chemical formulas, molecular weights and/or net charges for each metabolite included in the metabolic model.

Reference values for each metabolite can be manually provided or downloaded through the `downloadChEBI` function included into the `minval` package from the Chemical Entities of Biological Interest (ChEBI) database, a freely available dictionary of molecular entities focused on 'small' chemical compounds involved in biochemical reactions. To download the latest version of the ChEBI database just type:

```
> ChEBI <- downloadChEBI(release = "latest",
+                         woAssociations = TRUE)
```

The `checkBalance` function included into the `minval` package can test mass-charge balance using a user-given reference of formulas, masses or charges. The `checkBalance` function returns a boolean value 'TRUE' if stoichiometric reaction is balanced. For this example an user provided reference is used.

```
> # Loading reference
> chemicalData <- read.csv2(file = system.file("extdata", "chemData.csv",
+                                             package = "minval"))
> # Mass-Balance evaluation
> checkBalance(reactionList = glycolysisModel$REACTION,
+             referenceData = chemicalData,
+             ids = "NAME",
+             mFormula = "FORMULA")

[1] TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[13] TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

As is shown above, the third stoichiometric reaction is unbalanced. It can be corrected replacing manually the unbalanced reaction as follows:

```
> glycolysisModel$REACTION[3] <- "D-Glyceraldehyde 3-phosphate[c] + Orthophosphate[c] +
+ NAD+[c] <=> 3-Phospho-D-glyceroyl phosphate[c] + NADH[c] + H+[c]"
```

And mass-balance can be tested again:

```
> checkBalance(reactionList = glycolysisModel$REACTION,
+             referenceData = chemicalData,
+             ids = "NAME",
+             mFormula = "FORMULA")

[1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[16] TRUE TRUE TRUE TRUE
```

When all stoichiometric reactions are mass-balanced, then the model can be exported and loaded to be interrogated again:

```
> writeSBMLmod(modelData = glycolysisModel,
+             modelID = "GlycolysisBalanced",
+             outputFile = "glycolysisBalanced.xml")
> sybil::optimizeProb(sybilSBML::readSBMLmod("glycolysisBalanced.xml"))
```

```
solver:                glpkAPI
method:                 simplex
algorithm:              fba
number of variables:    19
number of constraints:  18
return value of solver: solution process was successful
solution status:        solution is optimal
value of objective function (fba): 2.000000
value of objective function (model): 2.000000
```

```
> stoichiometricMatrix(reactionList = glycolysisModel$REACTION)
```

	reactions									
metabolites	R01	R02	R03	R04	R05	R06	R07	R08	R09	R10
2-Phospho-D-glycerate[c]	-1	0	0	0	0	-1	0	0	0	0

Phosphoenolpyruvate[c]	1	0	0	0	0	0	0	0	0	-1
H2O[c]	1	0	0	0	0	0	0	0	0	0
D-Glyceraldehyde 3-phosphate[c]	0	-1	-1	1	0	0	0	0	0	0
Glycerone phosphate[c]	0	1	0	1	0	0	0	0	0	0
Orthophosphate[c]	0	0	-1	0	0	0	0	0	0	0
NAD+[c]	0	0	-1	0	0	0	0	0	0	0
3-Phospho-D-glyceroyl phosphate[c]	0	0	1	0	1	0	0	0	0	0
NADH[c]	0	0	1	0	0	0	0	0	0	0
H+[c]	0	0	1	0	0	0	0	0	0	0
beta-D-Fructose 1,6-bisphosphate[c]	0	0	0	-1	0	0	0	0	1	0
ATP[c]	0	0	0	0	-1	0	-1	0	-1	1
3-Phospho-D-glycerate[c]	0	0	0	0	-1	1	0	0	0	0
ADP[c]	0	0	0	0	1	0	1	0	1	-1
alpha-D-Glucose[c]	0	0	0	0	0	0	-1	0	0	0
alpha-D-Glucose 6-phosphate[c]	0	0	0	0	0	0	1	-1	0	0
beta-D-Fructose 6-phosphate[c]	0	0	0	0	0	0	0	1	-1	0
Pyruvate[c]	0	0	0	0	0	0	0	0	0	1
reactions										
metabolites	R11	R12	R13	R14	R15	R16	R17	R18	R19	
2-Phospho-D-glycerate[c]	0	0	0	0	0	0	0	0	0	
Phosphoenolpyruvate[c]	0	0	0	0	0	0	0	0	0	
H2O[c]	-1	0	0	0	0	0	0	0	0	
D-Glyceraldehyde 3-phosphate[c]	0	0	0	0	0	0	0	0	0	
Glycerone phosphate[c]	0	0	0	0	0	0	0	0	0	
Orthophosphate[c]	0	0	0	-1	0	0	0	0	0	
NAD+[c]	0	-1	0	0	0	0	0	0	0	
3-Phospho-D-glyceroyl phosphate[c]	0	0	0	0	0	0	0	0	0	
NADH[c]	0	0	-1	0	0	0	0	0	0	
H+[c]	0	0	0	0	-1	0	0	0	0	
beta-D-Fructose 1,6-bisphosphate[c]	0	0	0	0	0	0	0	0	0	
ATP[c]	0	0	0	0	0	-1	0	0	0	
3-Phospho-D-glycerate[c]	0	0	0	0	0	0	0	0	0	
ADP[c]	0	0	0	0	0	0	0	0	-1	
alpha-D-Glucose[c]	0	0	0	0	0	0	-1	0	0	
alpha-D-Glucose 6-phosphate[c]	0	0	0	0	0	0	0	0	0	
beta-D-Fructose 6-phosphate[c]	0	0	0	0	0	0	0	0	0	
Pyruvate[c]	0	0	0	0	0	0	0	-1	0	

Reactants and Products

As described before, stoichiometric reactions represent the transformation of reactants into products in a chemical reaction. The **reactants** and **products** functions extract and return all reactants or products respectively in a stoichiometric reaction as a vector. If reaction is irreversible ('=>') then reactants and products are separated and returned afterward as follows:

```
> reactants(reactionList = "ADP[c] + Phosphoenolpyruvate[c] => ATP[c] + Pyruvate[c]")
```

```
[1] "ADP[c]" "Phosphoenolpyruvate[c]"
```

```
> products(reactionList = "ADP[c] + Phosphoenolpyruvate[c] => ATP[c] + Pyruvate[c]")
```

```
[1] "ATP[c]" "Pyruvate[c]"
```

In reversible cases ('<=>') all reactants at some point can act as products and *vice versa*, for that reason both functions return all reaction metabolites:

```
> reactants(reactionList = "H2O[c] + Urea-1-Carboxylate[c] <=> 2 CO2[c] + 2 NH3[c]")
```

```
[1] "H2O[c]" "Urea-1-Carboxylate[c]" "CO2[c]"
```

```
[4] "NH3[c]"
```

```
> products(reactionList = "H2O[c] + Urea-1-Carboxylate[c] <=> 2 CO2[c] + 2 NH3[c]")
```

```
[1] "H2O[c]" "Urea-1-Carboxylate[c]" "CO2[c]"
```

```
[4] "NH3[c]"
```

Metabolites

The `metabolites` function automatically identifies and lists all metabolites (with or without compartments) for a specific or a set of stoichiometric reactions. This list is usually required for programs that perform FBA analysis as an independent input spreadsheet. In this example we show how to extract all metabolites (reactants and products) with and without compartment included in the RECON 2.04 metabolic reconstruction.

```
> metabolites(reactionList = glycolysisModel$REACTION)

[1] "2-Phospho-D-glycerate[c]"      "Phosphoenolpyruvate[c]"
[3] "H2O[c]"                       "D-Glyceraldehyde 3-phosphate[c]"
[5] "Glycerone phosphate[c]"        "Orthophosphate[c]"
[7] "NAD+[c]"                      "3-Phospho-D-glyceroyl phosphate[c]"
[9] "NADH[c]"                      "H+[c]"
[11] "beta-D-Fructose 1,6-bisphosphate[c]" "ATP[c]"
[13] "3-Phospho-D-glycerate[c]"      "ADP[c]"
[15] "alpha-D-Glucose[c]"            "alpha-D-Glucose 6-phosphate[c]"
[17] "beta-D-Fructose 6-phosphate[c]" "Pyruvate[c]"
```

Orphan Metabolites

Those compounds that are not produced or consumed in any other reaction, generally called orphan metabolites, are one of the main causes of mass unbalance in metabolic reconstructions. The `orphanReactants` function, identifies compounds that are not produced internally by any other reaction and should be added to the reconstruction, for instance, as an exchange reaction following the protocol proposed by ?. In following example we show how to extract all orphan compounds included in the Human metabolic reconstruction.

```
> head(orphanReactants(reactionList = glycolysisModel$REACTION), n = 100)
```

```
[1] NA
```

following the protocol proposed by ?. In this example we show the option added to `orphan.*` functions, that permits to report the orphan metabolites as a list grouped by compartment:

```
> head(orphanProducts(reactionList = glycolysisModel$REACTION, byCompartment = TRUE), n = 100)
```

```
[[1]]
character(0)
```

Compartments

As well as in cells, in which not all reactions occur in all compartments, stoichiometric reactions in a metabolic reconstruction can be labeled to be restricted for a single compartment during FBA, by the assignment of a compartment label after each metabolite name. Some FBA implementations require the reporting of all compartments included in the metabolic reconstruction as an independent section of the human-readable input file. In this example we show how to extract all compartments for all reactions included in the glutamate/glutamine cycle.

```
> compartments(glugln)
```

```
[1] "c_n" "r_n" "m_n" "c_a" "r_a" "m_a"
```

Association with ChEBI

The Chemical Entities of Biological Interest (**ChEBI**) database is a freely available dictionary of molecular entities focused on 'small' chemical compounds involved in biochemical reactions (?). Amongst other characteristics, the release 136 of the ChEBI database contains a set of standardized metabolite names, synonyms and molecular formula for at least 52521 chemical compounds. The use of standardized metabolite names facilitate the sharing process and inter-conversion to another metabolite names or identifiers (??). The `minval` package contains five functions to validate and extract values from a local copy of the ChEBI database release 136. The `is.chebi` function takes a compound name as input, compares it against all the compounds names in ChEBI and returns a logical value TRUE if a match is found. In this next four examples we show the potential use of the functions using as input the acetyl-CoA compound.

```
> is.chebi("acetyl-CoA")
```

```
[1] TRUE
```

The `chebi.id` function takes a compound name as input, compares it against all the compounds names in ChEBI and returns the compound identifier if a match is found.

```
> chebi.id("acetyl-CoA")
```

```
[1] "15351"
```

The `chebi.formula` function takes a compound name as input, compares it against all the compounds names in ChEBI and returns the molecular formula if a match is found.

```
> chebi.formula("acetyl-CoA")
```

```
[1] "C23H38N7O17P3S"
```

The `chebi.candidates` function takes a compound name as input, compares it against all the compounds synonyms in ChEBI and returns possible compound names if a match is found.

```
> candidates<-chebi.candidates("acetyl-CoA")
```

```
> head(candidates)
```

```
[1] "acetoacetyl-CoA"          "acetyl-CoA"
[3] "(1-hydroxycyclohexyl)acetyl-CoA" "cinnamoyl-CoA"
[5] "2-methylacetoacetyl-CoA"    "phenylacetyl-CoA"
```

The `to.ChEBI` function translates the compounds names of a stoichiometric reaction into their corresponding identifier or molecular formula in the ChEBI database. In this example we show how to use the `to.ChEBI` function for the Ubiquinol and FAD production reaction in astrocytes mitochondrias.

```
> toChEBI("FADH2[m_a] + ubiquinone-0[m_a] => FAD[m_a] + Ubiquinol[m_a]")
```

```
[1] "1 17877 + 1 27906 => 1 16238 + 1 17976"
```

```
> toChEBI("FADH2[m_a] + ubiquinone-0[m_a] => FAD[m_a] + Ubiquinol[m_a]", formula = TRUE)
```

```
[1] "1 C27H35N9O15P2 + 1 C9H10O4 => 1 C27H33N9O15P2 + 1 C9H12O4(C5H8)n"
```

Mass Balance Validation

Thermodynamic unbalance of genome-scale metabolic reconstructions can also be promoted by stoichiometric mistakes. In a well balanced stoichiometric reaction according to the Lomonósov-Lavoisier law, the mass comprising the reactants should be the same mass present in the products. The `unbalanced` function converts the metabolites identifiers to molecular formulas, multiplies the atom numbers by their respective stoichiometric coefficient, and establishes if the atomic composition of reactants and products are the same, it returns a logical value TRUE if mass is unbalanced. In this example we show the mass balance evaluation for the first twenty reactions of the glutamate/glutamine cycle.

```
> unbalanced(glugln[1:20])
```

```
[1] FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[13] FALSE TRUE FALSE FALSE TRUE TRUE TRUE TRUE
```

The `unbalanced` function also include an option to show the molecular formula of mass unbalanced formulas through the option `show.formulas`.

```
> unbalanced(glugln[1:20], show.formulas = TRUE)
```

```
[,1]
[1,] "alpha-D-Glucose 6-phosphate[r_n] + water[r_n] => alpha-D-Glucose[r_n] + phos ..."
[2,] "beta-D-fructofuranose 1,6-bisphosphate[c_n] + water[c_n] => beta-D-fructofur ..."
[3,] "D-Glyceraldehyde 3-phosphate[c_n] + phosphate(3-)[c_n] + NAD(+)[c_n] <=> 3-p ..."
[4,] "ATP[c_n] + 3-phosphoglyceric acid[c_n] <=> ADP[c_n] + 3-phosphonato-D-glycer ..."
[5,] "3-phosphonato-D-glyceroyl phosphate(4-)[c_n] => 2,3-bisphospho-D-glyceric ac ..."
[6,] "2,3-bisphospho-D-glyceric acid[c_n] + water[c_n] => 3-phosphoglyceric acid[c ..."
```

```
[,2]
[1,] "1 C6H13O9P + 1 H2O => 1 C6H12O6 + 1 O4P"
[2,] "1 C6H14O12P2 + 1 H2O => 1 C6H13O9P + 1 O4P"
[3,] "1 C3H7O6P + 1 O4P + 1 C21H28N7O14P2 <=> 3 C3H4O10P2 + 1 C21H29N7O14P2 + 1 H"
[4,] "1 C10H16N5O13P3 + 3 C3H7O7P <=> 1 C10H15N5O10P2 + 3 C3H4O10P2"
[5,] "3 C3H4O10P2 => 2 C3H8O10P2"
[6,] "2 C3H8O10P2 + 1 H2O => 3 C3H7O7P + 1 O4P"
```

Summary

We introduced the **minval** package to evaluate mass balancing correctness of metabolic reconstructions and to extract all reactants, products, orphan metabolites, metabolite names and compartments for a set of stoichiometric reactions, which together represent the minimal validation that should be performed in a genome-scale metabolic reconstruction. We also show in a step by step fashion, how this minimal evaluation process of mass balance can be performed using the 128 non-exchange reactions included in the glutamate/glutamine cycle included in the “glugln” dataset. We also showed some examples of metabolite names - ChEBI database association procedures.

Acknowledgements

DO and JG were supported by the Pontificia Universidad Javeriana (Grant ID 5619, 6371, 6375)

Bibliography

- R. Agren, L. Liu, S. Shoaie, W. Vongsangnak, I. Nookaew, and J. Nielsen. The RAVEN Toolbox and Its Use for Generating a Genome-scale Metabolic Model for *Penicillium chrysogenum*. *PLoS Computational Biology*, 9(3), 2013. doi: 10.1371/journal.pcbi.1002980. [p2]
- S. A. Becker, A. M. Feist, M. L. Mo, G. Hannum, B. Ø. Palsson, and M. J. Herrgard. Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox. *Nature protocols*, 2(3):727–38, 2007. doi: 10.1038/nprot.2007.99. [p2]
- T. Bernard, A. Bridge, A. Morgat, S. Moretti, I. Xenarios, and M. Pagni. Reconciliation of metabolites and biochemical reactions for metabolic networks. *Briefings in Bioinformatics*, 15(1): 123–135, 2014. doi: 10.1093/bib/bbs058. [p6]
- R. Caspi, T. Altman, R. Billington, K. Dreher, H. Foerster, C. A. Fulcher, T. A. Holland, I. M. Keseler, A. Kothari, A. Kubo, M. Krummenacker, M. Latendresse, L. A. Mueller, Q. Ong, S. Paley, P. Subhraveti, D. S. Weaver, D. Weerasinghe, P. Zhang, and P. D. Karp. The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases. *Nucleic Acids Research*, 42(D1):D459–D471, 2014. doi: 10.1093/nar/gkt1103. [p1]
- A. Chang, I. Schomburg, S. Placzek, L. Jeske, M. Ulbrich, M. Xiao, C. W. Sensen, and D. Schomburg. BRENDA in 2015: exciting developments in its 25th year of existence. *Nucleic Acids Research*, 43 (D1):D439–D446, 2015. doi: 10.1093/nar/gku1068. [p1]
- W. L. Chen, D. Z. Chen, and K. T. Taylor. Automatic reaction mapping and reaction center detection. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 3(6):560–593, 2013. doi: 10.1002/wcms.1140. [p1]
- D. Croft, A. F. Mundo, R. Haw, M. Milacic, J. Weiser, G. Wu, M. Caudy, P. Garapati, M. Gillespie, M. R. Kamdar, B. Jassal, S. Jupe, L. Matthews, B. May, S. Palatnik, K. Rothfels, V. Shamovsky, H. Song, M. Williams, E. Birney, H. Hermjakob, L. Stein, and P. D’Eustachio. The Reactome pathway knowledgebase. *Nucleic Acids Research*, 42(D1):D472–D477, 2014. doi: 10.1093/nar/gkt1102. [p1]
- K. Degtyarenko, P. de Matos, M. Ennis, J. Hastings, M. Zbinden, A. McNaught, R. Alcantara, M. Darsow, M. Guedj, and M. Ashburner. ChEBI: a database and ontology for chemical entities of biological interest. *Nucleic Acids Research*, 36:D344–D350, 2007. doi: 10.1093/nar/gkm791. [p6]

- A. Gevorgyan, M. G. Poolman, and D. a. Fell. Detection of stoichiometric inconsistencies in biomolecular models. *Bioinformatics*, 24(19):2245–2251, 2008. doi: 10.1093/bioinformatics/btn425. [p1]
- J. B. Hendrickson. Comprehensive System for Classification and Nomenclature of Organic Reactions. *Journal of Chemical Information and Computer Sciences*, 37(97):850–852, 1997. doi: 10.1021/ci970040v. [p1]
- T. Jewison, Y. Su, F. M. Disfany, Y. Liang, C. Knox, A. Maciejewski, J. Poelzer, J. Huynh, Y. Zhou, D. Arndt, Y. Djoumbou, Y. Liu, L. Deng, A. C. Guo, B. Han, A. Pon, M. Wilson, S. Rafatnia, P. Liu, and D. S. Wishart. SMPDB 2.0: Big Improvements to the Small Molecule Pathway Database. *Nucleic Acids Research*, 42(D1):D478–D484, 2014. doi: 10.1093/nar/gkt1067. [p1]
- M. Kanehisa. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, 28(1): 27–30, 2000. doi: 10.1093/nar/28.1.27. [p1]
- H. U. Kim, T. Y. Kim, and S. Y. Lee. Metabolic flux analysis and metabolic engineering of microorganisms. *Mol. BioSyst.*, 4(2):113–120, 2008. doi: 10.1039/B712395G. [p1]
- M. Lakshmanan, G. Koh, B. K. S. Chung, and D.-Y. Lee. Software applications for flux balance analysis. *Briefings in Bioinformatics*, 15(1):108–122, 2014. doi: 10.1093/bib/bbs069. [p1]
- A. Lambert, J. Dubois, and R. Bourqui. Pathway preserving representation of metabolic networks. *Computer Graphics Forum*, 30(3):1021–1030, 2011. doi: 10.1111/j.1467-8659.2011.01951.x. [p1]
- J. M. Park, T. Y. Kim, and S. Y. Lee. Constraints-based genome-scale metabolic simulation for systems metabolic engineering. *Biotechnology Advances*, 27(6):979–988, 2009. doi: 10.1016/j.biotechadv.2009.05.019. [p1]
- A. Ravikrishnan and K. Raman. Critical assessment of genome-scale metabolic networks: the need for a unified standard. *Briefings in Bioinformatics*, 16(6):1057–1068, 2015. doi: 10.1093/bib/bbv003. [p6]
- E. Reznik, P. Mehta, and D. Segrè. Flux Imbalance Analysis and the Sensitivity of Cellular Growth to Changes in Metabolite Pools. *PLoS Computational Biology*, 9(8):e1003195, 2013. doi: 10.1371/journal.pcbi.1003195. [p1]
- I. Thiele and B. Ø. Palsson. A protocol for generating a high-quality genome-scale metabolic reconstruction. *Nature Protocols*, 5(1):93–121, 2010. doi: 10.1038/nprot.2009.203. [p1, 6]

Daniel Osorio

Grupo de Investigación en Bioinformática y Biología de Sistemas
Instituto de Genética, Universidad Nacional de Colombia
Bogotá
Colombia
dcosoriorh@unal.edu.co

Janneth Gonzalez

Grupo de Investigación en Bioquímica Experimental y Computacional
Facultad de Ciencias, Pontificia Universidad Javeriana
Bogotá
Colombia
janneth.gonzalez@javeriana.edu.co

Andres Pinzon-Velasco

Grupo de Investigación en Bioinformática y Biología de Sistemas
Instituto de Genética, Universidad Nacional de Colombia
Bogotá
Colombia
ampinzonv@unal.edu.co