

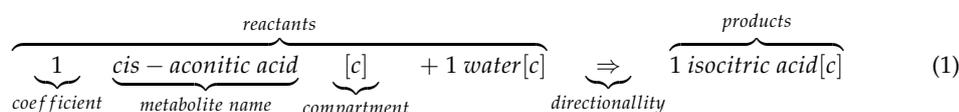
minval: An R package for MINimal VALidation of Stoichiometric Reactions

by Daniel Osorio, Janneth González and Andrés Pinzón

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* (Chen et al., 2013). 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 1 (Hendrickson, 1997). In biochemistry, a set of chemical reactions that transform a substrate into a product, after several chemical transformations is called a metabolic pathway (Lambert et al., 2011). 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* (Park et al., 2009) and has become an indispensable tool for studying metabolism of biological entities at the systems level (Thiele and Palsson, 2010).



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 (Kanehisa, 2000), BioCyc (Caspi et al., 2014), Reactome (Croft et al., 2014), BRENDA (Chang et al., 2015) or SMPDB (Jewison et al., 2014). 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 (Thiele and Palsson, 2010; Gevorgyan et al., 2008). 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 (Lakshmanan et al., 2014).

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 (Kim et al., 2008; Park et al., 2009). 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 (Reznik et al., 2013). 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 (Park et al., 2009; Thiele and Palsson, 2010). 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** (Becker et al., 2007) and **RAVEN** (Agren et al., 2013) which operate as tools under the commercial MATLAB® environment. On the R environment side **sybil** (Gelius-Dietrich et al., 2013) and **abcdeFBA** (Gangadharan and Rohatgi, 2012) 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 deurate metabolic reconstructions before its interrogation through Flux Balance Analysis (FBA).

To show the potential use of the functions included into the **minval** package, a human-readable model composed by a set of 19 stoichiometric reactions that represent the glycolysis process was included. Glycolysis is the metabolic pathway that converts a molecule of glucose (C₆H₁₂O₆), into two molecules of pyruvate (CH₃COCOO⁻ + 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]

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 in the GitHub repository <http://github.com/gibbslab/minval>.

Inputs and syntaxis

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 load the included glycolysis model just type:

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

Syntax validation

The first step for a metabolic reconstruction validation is to check the syntax of their stoichiometric reactions. The validateSyntax function validate the syntax (Equation 1) 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
[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:

```
> colnames(glycolysisModel)
[1] "ID" "DESCRIPTION" "REACTION" "GPR" "LOWER.BOUND"
[6] "UPPER.BOUND" "OBJECTIVE"
```

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
DESCRIPTION	A reaction description	Optional (the column can be empty)
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

SBML files

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 was used.

```
> # Loading reference
> chemicalData <- read.csv2(file = system.file("extdata", "chemData.csv",
+                                             package = "minval"))
> head(chemicalData, n= 5)
```

	NAME	FORMULA	MASS	CHARGE
1	H2O	H2O	18.0106	0
2	H+	H	1.0078	1
3	ATP	C10H16N5O13P3	506.9957	0
4	NAD+	C21H28N7O14P2	664.1169	1
5	3-Phospho-D-glyceroyl phosphate	C3H8O10P2	265.9593	0

```
> # 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 TRUE
[13] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

As is shown above, the third stoichiometric reaction is mass-unbalanced. It can be corrected replacing manually the unbalanced reaction by a balanced one 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, in this case using the molecular mass of each metabolite as reference:

```
> checkBalance(reactionList = glycolysisModel$REACTION,
+             referenceData = chemicalData,
+             ids = "NAME",
+             mWeight = "MASS")
[1] 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

```

As shown above, the correct mass-charge balance allows predicting in an accurate way the net yield of pyruvate from an alpha-D-glucose molecule through the glycolytic pathway using FBA analysis.

Characterize model

A metabolic reconstruction generally includes three types of reactions *compartmentalized*, *transport* and *exchange* reactions. The *compartmentalized reactions* are those in where all involved metabolites (all reactants and products) are assigned to the same compartment. e.g. $h[m] + nadph[m] + o2[m] + 25hvitd2[m] \Rightarrow h2o[m] + nadp[m] + 1a25dhvitd2[m]$. The *transport reactions* are those in where the involved metabolites are assigned to two or more compartments. e.g. $2 hco3[e] + na1[e] \Leftrightarrow 2 hco3[c] + na1[c]$, and finally, the *exchange reactions* are those used to import or release metabolites to the boundary. e.g. $acetone[e] \Leftrightarrow$. Characterize the stoichiometric reactions of a metabolic model is a required and time-consuming work. The **minval** package includes the `characterizeReactions` function to characterize the stoichiometric reactions and metabolites by type and compartment. This function counts the number of reactions, computes the relative frequency of each reaction type (transport, exchange and compartmentalized), computes the relative frequency of reactions by compartment, counts the number of unique metabolites and computes the relative frequency of metabolites by compartment. The `characterizeReactions` function returns all these information as a labeled list. To show its potential use, the RECON 2.04 Human Metabolic Reconstruction (Thiele et al., 2013) was included in a human-readable format. To load and characterize it just type:

```

> # Loading the Human Metabolic Reconstruction RECON 2.04
> RECON <- read.csv(system.file("extdata", "rRECON2.csv",
+                             package = "minval"))
> # Characterizing the stoichiometric reactions
> charRECON <- characterizeReactions(reactionList = RECON$REACTION)
> charRECON
$Reactions
[1] 7441

$rType

Compartmentalized reaction      Exchange reaction
                    55.825830                9.420777
Transport reaction
                    34.753393

$Reaction

      c      e      g      l      m      n      r      x
24.593469 1.760516 3.628545 2.983470 9.958339 1.666443 6.208843 5.026206

$Metabolites
[1] 5063

$Metabolites

      c      e      g      l      m      n      r      x
37.092633 12.680229 6.261110 5.964843 14.892356 3.258937 11.258147 8.591744

```

Computed values can be easily plotted as follows:

```

> # Combining two plots into one overall graph
> par(mfrow=c(1,2))
> # Plotting reactions by Type

```

```

> pie(x = charRECON$rType,
+     main = "Reactions by Type")
> # Plotting reactions by Compartment
> pie(x = charRECON$cReaction,
+     main = "Reactions by Compartment",
+     labels = compartmentNames)

```

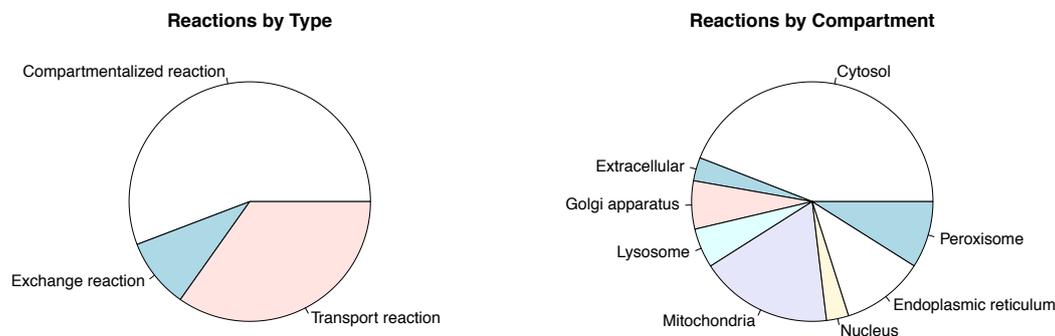


Figure 1: Distribution by type (left) and by compartments (right) of the reactions included into the RECON 2.04 Human Metabolic Reconstruction (Thiele et al., 2013).

Stoichiometric matrix

A metabolic reconstruction is often represented in a more compact form called the stoichiometry matrix (S). If a metabolic reconstruction has n reactions and m participating metabolites, then the stoichiometry matrix will have correspondingly m rows and n columns. Values in the stoichiometric matrix represent the metabolite coefficients in each reaction. To generate the stoichiometric matrix of a metabolic reconstruction just type:

```

> stoichiometricMatrix(reactionList = glycolysisModel$REACTION)

```

metabolites	reactions									
	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

metabolites	reactions								
	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) included in a metabolic reconstruction with and without compartments.

```
> metabolites(reactionList = glycolysisModel$REACTION)
[1] "2-Phospho-D-glycerate[c]"           "Phosphoenolpyruvate[c]"
[3] "H2O[c]"                             "D-Glyceraldehyde 3-phosphate[c]"
[5] "Glycerone phosphate[c]"             "Orthophosphate[c]"
[7] "NAD+[c]"                            "3-Phospho-D-glyceroyl phosphate[c]"
[9] "NADH[c]"                            "H+[c]"
[11] "beta-D-Fructose 1,6-bisphosphate[c]" "ATP[c]"
[13] "3-Phospho-D-glycerate[c]"           "ADP[c]"
[15] "alpha-D-Glucose[c]"                 "alpha-D-Glucose 6-phosphate[c]"
[17] "beta-D-Fructose 6-phosphate[c]"     "Pyruvate[c]"
> metabolites(reactionList = glycolysisModel$REACTION, woCompartment = TRUE)
[1] "2-Phospho-D-glycerate"           "Phosphoenolpyruvate"
[3] "H2O"                             "D-Glyceraldehyde 3-phosphate"
[5] "Glycerone phosphate"             "Orthophosphate"
[7] "NAD+"                            "3-Phospho-D-glyceroyl phosphate"
[9] "NADH"                            "H+"
[11] "beta-D-Fructose 1,6-bisphosphate" "ATP"
[13] "3-Phospho-D-glycerate"           "ADP"
[15] "alpha-D-Glucose"                 "alpha-D-Glucose 6-phosphate"
[17] "beta-D-Fructose 6-phosphate"     "Pyruvate"
```

Orphan metabolites

Those compounds that are not produced or consumed in any other reaction are generally called orphan metabolites, they represent one of the main causes of mass unbalances in metabolic reconstructions

generating dead-ends without flux. The `orphanMetabolites` function extracts all orphan compounds included into a metabolic reconstruction.

```
> orphanMetabolites(reactionList = glycolysisModel$REACTION[noExchange])
[1] "alpha-D-Glucose[c]" "Pyruvate[c]"      "H+[c]"
[4] "H2O[c]"              "NAD+[c]"      "NADH[c]"
[7] "Orthophosphate[c]"
```

Due not all orphans are not consumed and not produced, 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 input exchange reaction following the protocol proposed by [Thiele and Palsson \(2010\)](#).

```
> orphanReactants(reactionList = glycolysisModel$REACTION[noExchange])
[1] "alpha-D-Glucose[c]" "H+[c]"      "H2O[c]"
[4] "NAD+[c]"           "NADH[c]"    "Orthophosphate[c]"
```

By another side, the `orphanProducts` function, identifies compounds that are not consumed internally by any other reaction and should be added to the reconstruction, for instance, as an output exchange (sink) reaction.

```
> orphanProducts(reactionList = glycolysisModel$REACTION[noExchange])
[1] "Pyruvate[c]"      "H+[c]"      "H2O[c]"
[4] "NAD+[c]"          "NADH[c]"    "Orthophosphate[c]"
```

Compartments

As well as in eukaryotic cells, in which not all reactions occur in all compartments, stoichiometric reactions in a metabolic reconstruction can be labeled to be restricted to 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 RECON 2.04 Human Metabolic Reconstruction ([Thiele et al., 2013](#)).

```
> compartments(reactionList = RECON$REACTION)
[1] "c" "l" "m" "r" "e" "x" "n" "g"
```

TSV files

Additional to the SBML format, the TSV format is the default input of metabolic models for the `sybil` R package. The TSV format is composed of three text files, following a character-separated (tab by default) value format where each line contains one entry (stoichiometric reaction and associated info). The `writeTSVmod` function can write a metabolic model in a TSV format as follows:

```
> writeTSVmod(modelData = glycolysisModel,
+             modelID = "Glycolysis",
+             outputFile = "glycolysis")
```

Metabolic models in TSV format can be readed through the `readTSVmod` function included in the `sybil` package:

```
> sybil::readTSVmod(prefix = "glycolysis",quoteChar = "\\")
model name:          Glycolysis
number of compartments 1
                    [c]
number of reactions: 19
number of metabolites: 18
number of unique genes: 22
objective function:  +1 R00200
```

Summary

We introduced the `minval` package to check the syntax validity, evaluate the mass-charge balance and extract all orphan metabolites of a set of stoichiometric reactions. Together, this steps represent the minimal validation that should be performed in a genome-scale metabolic reconstruction. Functions to characterize and export metabolic models in SBML and TSV formats as well as to extract all

reactants, products, metabolite names and compartments for a set of stoichiometric reactions were also introduced. Moreover, we also show in a step by step fashion, how this minimal evaluation process of mass balance can avoid an overestimation of the net yield of pyruvate from an alpha-D-glucose molecule when using an unbalanced model of the glycolysis pathway.

Acknowledgements

DO and JG were supported by the Pontificia Universidad Javeriana (Grant ID 5619, 6235, 6371, 6375). We thank to the anonymous reviewers for their helpful comments and suggestions to improve the package.

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. URL <https://doi.org/10.1371/journal.pcbi.1002980>. [p115]
- 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. URL <https://doi.org/10.1038/nprot.2007.99>. [p115]
- R. Caspi, T. Altman, R. Billington, K. Dreher, H. Foerster, C. A. Fulcher, T. A. Holland, I. M. Kessler, 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. URL <https://doi.org/10.1093/nar/gkr1014>. [p114]
- 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. URL <https://doi.org/10.1093/nar/gku1068>. [p114]
- 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. URL <https://doi.org/10.1002/wcms.1140>. [p114]
- 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. URL <https://doi.org/10.1093/nar/gkv1351>. [p114]
- A. Gangadharan and N. Rohatgi. *ABCDE FBA: A-Biologist-Can-Do-Everything of Flux Balance Analysis with this package*, 2012. URL <https://CRAN.R-project.org/package=abcdeFBA>. R package version 0.4. [p115]
- G. Gelius-Dietrich, C. J. Fritze-meier, A. A. Desouki, and M. J. Lercher. Sybil – efficient constraint-based modelling in r. *BMC Systems Biology*, 7(1):125, 2013. ISSN 1752-0509. URL <https://doi.org/10.1186/1752-0509-7-125>. [p115]
- A. Gevorgyan, M. G. Poolman, and D. a. Fell. Detection of Stoichiometric Inconsistencies in Biomolecular Models. *Bioinformatics*, 24(19):2245–2251, 2008. URL <https://doi.org/10.1093/bioinformatics/btn425>. [p114]
- J. B. Hendrickson. Comprehensive System for Classification and Nomenclature of Organic Reactions. *Journal of Chemical Information and Computer Sciences*, 37(97):850–852, 1997. URL <https://doi.org/10.1021/ci970040v>. [p114]
- 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. URL <https://doi.org/10.1093/nar/gkt1067>. [p114]
- M. Kanehisa. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, 28(1):27–30, 2000. URL <https://doi.org/10.1093/nar/28.1.27>. [p114]

- 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. URL <https://doi.org/10.1039/b712395g>. [p114]
- 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. URL <https://doi.org/10.1093/bib/bbs069>. [p114]
- A. Lambert, J. Dubois, and R. Bourqui. Pathway Preserving Representation of Metabolic Networks. *Computer Graphics Forum*, 30(3):1021–1030, 2011. URL <https://doi.org/10.1111/j.1467-8659.2011.01951.x>. [p114]
- 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. URL <https://doi.org/10.1016/j.biotechadv.2009.05.019>. [p114]
- 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. URL <https://doi.org/10.1371/journal.pcbi.1003195>. [p114]
- I. Thiele and B. Ø. Palsson. A protocol for Generating a High-Quality Genome-Scale Metabolic Reconstruction. *Nature Protocols*, 5(1):93–121, 2010. URL <https://doi.org/10.1038/nprot.2009.203>. [p114, 121]
- I. Thiele, N. Swainston, R. M. Fleming, A. Hoppe, S. Sahoo, M. K. Aurich, H. Haraldsdottir, M. L. Mo, O. Rolfsson, M. D. Stobbe, and others. A community-driven global reconstruction of human metabolism. *Nature biotechnology*, 31(5):419–425, 2013. URL <https://doi.org/10.1038/nbt.2488>. [p118, 119, 121]

Daniel Osorio

Grupo de Investigación en Bioinformática y Biología de Sistemas

Instituto de Genética, Universidad Nacional de Colombia

Bogotá

Colombia

ORCID: 0000-0003-4424-8422

dcosorioh@unal.edu.co

Janneth González

Grupo de Investigación en Bioquímica Experimental y Computacional

Facultad de Ciencias, Pontificia Universidad Javeriana

Bogotá

Colombia

ORCID: 0000-0003-2009-3374

janneth.gonzalez@javeriana.edu.co

Andrés Pinzón

Grupo de Investigación en Bioinformática y Biología de Sistemas

Instituto de Genética, Universidad Nacional de Colombia

Bogotá

Colombia

ampinzonv@unal.edu.co