

G2F as a Novel Tool to Find and Fill Gaps in Metabolic Networks

by Daniel Osorio, Andrés Pinzón Velasco, Nicolás Mendoza-Mejía, Felipe Rojas-Rodríguez and Janneth González

Abstract During the building of a genome-scale metabolic model, there are several dead-end metabolites and substrates which cannot be imported, produced, nor used by any reaction incorporated in the network. The presence of these dead-end metabolites can block out the net flux of the objective function when it is evaluated through Flux Balance Analysis (FBA), and when it is not blocked, bias in the biological conclusions increase. In this aspect, the refinement to restore the connectivity of the network can be carried out manually or using computational algorithms. The g2f package was designed as a tool to find the gaps from dead-end metabolites and fill them from the stoichiometric reactions of a reference, filtering candidate reactions using a weighting function. Additionally, this algorithm allows to download all the sets of gene-associated stoichiometric reactions for a specific organism from the KEGG database. Our package is compatible with both 4.0.0 and 3.6.0 R versions.

Key words

genome-scale metabolic model, gap filling, g2f algorithm, open-source.

Introduction

Genome-scale metabolic models (GEMs) are multi-compartment metabolic reconstructions that specify the set of chemical reactions catalyzed by an organism (usually hundreds to thousands) covering the metabolic biochemical molecular function of a complete genome (Szappanos et al., 2011). The main goal of these reconstructions is to relate the genome of a given organism with its physiology, incorporating every metabolic transformation that this organism can perform (Agren et al., 2013; Chen et al., 2012). The GEMs are converted into computational models for the simulation of a species-specific metabolism in order to gain insight into the complex interactions that give rise to the metabolic capabilities (Alper et al., 2005; Fong et al., 2005; Cook and Nielsen, 2017). The predictive accuracy of a model depends on the comprehensiveness and biochemical fidelity of the reconstruction (Thiele et al., 2014).

The GEM construction process can be divided in two fundamental stages: (1) The generation of a draft of the reconstructed network. Here, the reactions associated with the enzymes that participate in the metabolism of a particular organism are downloaded from specialized databases such as KEGG, MetaCyc, or ModelSEED (Pham et al., 2019; van Steijn et al., 2019). (2) A refinement of the network is done manually or through the use of computational algorithms (Pham et al., 2019; van Steijn et al., 2019). Similar steps are performed during the construction of a tissue-specific metabolic reconstruction, defined as the subset of reactions included in a genome-scale metabolic reconstruction that are highly associated with the metabolism of a specific tissue (Palsson, 2009; Schultz and Qutub, 2016; van Steijn et al., 2019). These are constructed from the measured gene expression or proteomic data allowing researchers to characterize and predict the metabolic behavior of a tissue under any physiological conditions (Ataman et al., 2017). It is important to highlight that a drawback of this approach arises from the fact that only the reactions associated with specific enzymes or genes can be mapped from the measured data. Therefore, the spontaneous reactions and non-facilitated transport reactions are missing in the first stages (Schultz and Qutub, 2016).

If all relevant exchange reactions are available, a high-quality model is expected to be able to carry flux in all its reactions (Agren et al., 2013); thus, a refinement stage in the reconstruction is required to restore the connectivity of the network. In this aspect, the gaps in the draft reconstruction are identified, and candidate reactions to fill the gaps are found using literature and metabolic databases (Satish Kumar et al., 2007; Thiele and Palsson, 2010). The network gaps can be associated with dead-end metabolites, which cannot be imported nor produced by any of the reactions in the network, or metabolites that are not used as substrates or released by any of the reactions. The presence of this type of metabolites can be problematic when the metabolic network is transformed into a metabolic steady-state model; mainly because flux through the network is blocked due to the incomplete connectivity with the rest of the network. Therefore, it is not possible to accurately optimize the metabolic flux distribution under an objective function, increasing the bias in the biological conclusions obtained from the reconstruction (Satish Kumar et al., 2007).

A manual refinement can be performed as an iterative process to assemble a higher confidence compendium of organism-specific metabolic reactions on a draft metabolic reconstruction (Bateman, 2010; Heavner and Price, 2015; Howe et al., 2008). Since the network reconstructions typically involve thousands of metabolic reactions, the model refinement can be a very complex task, which not only requires plenty of time and intensive use of available literature, databases and experimental data (Heavner and Price, 2015; Lakshmanan et al., 2014), but also can lead to the introduction of new errors and to overlook old ones (Agren et al., 2013; Machado et al., 2018). These metabolic network gap refinement can also be performed using several algorithms developed for open-source environments such as Python and GAMS or in closed-source environment such as MATLAB (Wang and Marci, 2018). Commonly implemented algorithms are mainly based on optimization procedures to fill the gaps that allow the production of a specific metabolite or give flux for a single biological objective function. Other algorithms modify the directionality of reactions or add new reactions to the model without associated evidence (Table 1)

Table 1: Description and comparison of the methods used for gap find and filling. The available algorithms are presented under the different environments.

Algorithm	Implementation		(Open source)	
	Package	Environment	Package	Environment
"SMILEY"	COBRApy	Python	Yes	Yes
"gapFind" and "gapFill"	-	GAMS	-	Yes
"growMatch"	COBRApy	Python	Yes	Yes
"fastgapfill"	openCOBRA	MATLAB	Yes	No

Table caption is put below the table

Table 1 we listed the four most used algorithms for gap filling across three environments. SMILEY, developed by Reed et al. (2006), identifies the minimum number of reactions required to allow the model a specific metabolite production through an optimization function. Reactions to fill the gaps are identified from a universal database of stoichiometric reactions, and the process is carried out one metabolite per time (user-defined). Alternatively, "gapFind" and "gapFill" in GAMS were developed by Satish Kumar et al. (2007) and identifies the metabolites ('gapFind') in the metabolic network reconstruction, which cannot be produced under any uptake conditions in both single and multicompartiment. Subsequently, 'gapFill' identify the reactions from a customized multi-organism database that restores the connectivity of these metabolites to the original network using optimization-based procedures. In the process, the procedure makes several intra-model modifications such as: (1) modify the directionality of the reactions in the model, (2) add fake external transport mechanisms, and (3) add fake intracellular transport reactions in multi-compartment models. "growMatch" was developed by Kumar and Maranas (2009), and it identifies the minimum number of reactions required to allow the model flux to a selected objective function through an optimization algorithm. Reactions to fill the gaps are identified from a universal database of stoichiometric reactions. The process is carried out one objective function per time (user-defined). Finally, developed by Thiele et al. (2014), the 'fastGapFill' algorithm identifies the blocked reactions through an optimization procedure. It searches candidate reactions to fill the gaps in a universal database of stoichiometric reactions through the 'fastCore' algorithm. This second algorithm computes a compact flux consistent model and uses it to filter and determine the reactions to be added. In the filling process, fake transport reactions between compartments are added.

In this aspect and with the aim of offering an open-source tool that improves the refinement of drafts network reconstructions and the depuration of metabolic models under the R environment, we introduce the g2f R package. This tool includes five functions to identify and fill gaps, calculate the addition cost of a reaction, and depurate metabolic networks of blocked reactions (no activated under any scenario). The implemented *gapFill* algorithm in g2f identifies the dead-end metabolites and traces them in a universal database of stoichiometric reactions used as a reference to select candidate reactions to be added. Selected reactions are then filtered by the function *additionCost* considering metabolites present in the original reconstruction to minimize the number of new metabolites to be added. The function calculates the cost of adding a reaction by dividing the amount of non-included metabolites in the reference metabolic network over the total number of metabolites involved in the reaction. The later is done to minimize the number of false-positive metabolites that could increase the number of new gaps in the model. Also, *blockedReactions* search for blocked reactions, so *gapFill* can fill blocked paths in the network. Finally, *getReactionsList* extracts the reactions from the model in the form of a list of strings, so it can be easily compared with the list or reactions obtained from *getReference*, which downloads specific stoichiometric matrices from KEGG in order to reconstruct specific organism models.

Table 2: Workflow of g2f packet

Workflow
Input: A Sybil metabolic model.
1. with <i>getReference</i> : Reference reactions list is retrieved from KEGG database.
2. with <i>blockedReactions</i> : Check if there is any dead-end metabolite, the results serve as a guide to the user.
3. with <i>getReactionsList</i> : List of reactions is extracted from input metabolic model.
4. with <i>additionCost</i> : The addition cost for the reference reactions list can be calculated to do a manual check.
5. with <i>gapFill</i> : Find dead-end metabolites and fill the gaps with reactions from the reference list which are below the addition cost treshold defined.
Loop user defined times (default = 5)
5.1. Search dead-end reactants and products.
5.2. Calculates the addition cost of the reference reactions.
5.3. Filters reference reactions with a cost above the threshold.
5.4. Selects the filtered reactions that have any orphan reactant or product.
5.5. Fills the gaps in the model with the selected reactions.
Output: List of the added reactions with their addition costs

Installation and Functions

The **g2f** package is available for download and installation from the Comprehensive R Archive Networks (CRAN, [Hornik \(2012\)](#)). This package is compatible with R 3.6.0 and 4.0.0 versions. To get the latest stable version of **g2f** install it directly from GitHub:

```
# Install 'devtools' R Package
R> install.packages('devtools')

# Install 'g2f' package
R> setRepositories(ind=1:2)
R> devtools::install_github('gibbslab/g2f')
R> library('g2f')
```

g2f includes 5 functions in order to identify gaps (metabolites not produced or not consumed in any reaction) and fill the gaps from a reference metabolic reconstruction. Briefly, the gap-filling reconstruction is based on the stoichiometric reaction matrix either from a specific model or by the complete set of gene-associated stoichiometric reactions for a specific organism from the KEGG database using a weighting function. Table 3 summarizes the functions contained in the **g2f** R package.

Table 3: Descriptions of **g2f** available functions.

Function	Description
blockedReactions	Identifies blocked reactions in a metabolic network.
additionCost	Calculates the cost of addition of a stoichiometric reaction.
getReactionsList	Extract the reaction list from a model.
getReference	Download all stoichiometric reactions from the KEGG database.
gapFill	Find and fill gaps in a metabolic network.

Downloading reference data from KEGG database

The KEGG database is a resource widely used as reference in genomics, metagenomics, metabolomics and other studies. Moreover, KEGG has been used for modeling and simulation in systems biology, specifically in GEMs ([Kanehisa, 2006](#); [Kanehisa et al., 2016](#); [Martín-Jiménez et al., 2017](#)). Currently, the database includes complete genomes, biological pathways, and the associated stoichiometric reactions for 542 eukaryotes, 5979 bacteria, and 334 archaea. The **g2f** *getReference* function downloads from KEGG all the gene-associated KeggOrthology (KO) stoichiometric reactions, and their correspondent

E.C. numbers for a customized organism, through the use of KEGG organism ID. Based on the KOs associated to the reactions, their respective gene-protein-reaction is constructed as follows: all genes associated to a given KO are linked by an AND operator. After that, when a reaction has more than one associated KO, previously linked genes are now joined by an OR operator. As an example, to download all the stoichiometric reactions (1492) associated to *Escherichia coli* just type:

```
R> e.coli <- getReference(organism = "eco")
```

Identify blocked reactions

To identify the blocked reactions included in a metabolic model, the `blockedReactions` function sets each one of the reactions included in the model (one at the time) as the objective function and optimizes the system through Flux Balance Analysis (FBA). Reactions that are not participating in any possible solution during all evaluations are returned as a blocked reaction.

As an example, we identify the blocked reactions in the E. coli core metabolic model included in the `sybil` package (Gelius-Dietrich et al., 2013).

```
R> data("Ec_core")
R> blockedReactions(Ec_core)
```

```
|=====| 100%
[1] "EX_fru(e)" "EX_fum(e)" "EX_mal_L(e)" "FUMt2_2" "MALt2_2"
```

Calculating the additional cost

Adding new reactions in order to fill gaps can be an easy path to increase the number of dead-end metabolites (Hosseini and Marashi, 2017). Therefore, as a strategy to reduce the possible addition of new dead-end metabolites into the system, the `additionCost` function calculates the cost of adding new metabolites based on metabolites that constitute the new reaction and those that compose the stoichiometric reactions already present in the metabolic reconstruction (Equation 1). Values of the function represent a weight ranging between 0 and 1.

$$additionCost = \frac{n(\text{metabolites}(\text{newReaction})) \notin (\text{metabolites}(\text{reactionList}))}{n(\text{metabolites}(\text{newReaction}))} \quad (1)$$

As an example, we select a sample of reactions from the downloaded reference for E. coli and calculate the addition cost for the remaining reactions (6 first values are shown).

```
R> reactionList <- sample(e.coli$reaction,10)
R> head(
+   additionCost(reaction = e.coli$reaction,
+   reference = reactionList)
+ )
[1] 1.0000000 1.0000000 1.0000000 0.8000000 0.8333333 1.0000000
```

To understand the results of the `additionCost` we present two examples for the glutamine synthetase reaction in the glutamate metabolism of E. coli core model.

```
[c]: ATP + Glu-L + Nh4 --> ADP + Gln-L + h + pi
```

The reaction takes as input Adenosine triphosphate (ATP), L-Glutamate (Glu-L) and Ammonium (Nh4) and produces Adenosine diphosphate (ADP), L-Glutamine (Gln-L), H+ (h) and inorganic Phosphate (pi) in the cytoplasm. We are going to assume that this reaction is going to be added to the model and that the number of metabolites to be added change between two conditions. In the first case, the reaction would be evaluated by `additionCost` but one of the seven metabolites is not present on the list of reactions of the complete model. On the second situation, four of the seven metabolites are not present in the metabolite list of the model. By dividing the number of metabolites to be added by the total number of metabolites in the reaction, `additionCost` produces 0.14 and 0.57 as resulting values for the two conditions, respectively. In this sense, if we pick a threshold of 0.2 for the `gapfill`, the first case would allow the reaction to be added but not the second condition. By using a threshold of 0.2 is possible to set a medium point for the reaction addition. Where higher values are more permissive and lower values are more restrictive.

"Gap find and fill" performing input and syntax

To identify network gaps in a metabolic model and fill them from a reference network, the "gapFill" function performs several steps: (1) The dead-end metabolites are identified from the stoichiometric matrix, (2) the candidate reactions to be added by comparing the metabolites against the metabolite list of the model, (3) the addition cost of each candidate reaction is calculated, (4) the candidate reactions with an addition cost lower or equal to the user-defined limit are added to the reaction list. Finally (5), the process returns to step 1 until no more original gaps can be filled under the user-defined limit. The function returns a set of candidate stoichiometric reactions to fill the original gaps included in the metabolic network.

As an example, we show how to fill dead-end metabolites included in the previously selected sample using all downloaded stoichiometric reactions from the KEGG database for E. coli as the reference.

```
R> reactionsAdded <- gapFill(reactionList = reactionList,
+   reference = e.coli$reaction,
+   limit = 1/4
+ )
48% gaps filled in the last iteration
26% gaps filled in the last iteration
13% gaps filled in the last iteration
13% gaps filled in the last iteration
4% gaps filled in the last iteration

R> head(reactionsAdded)
addCost                                     react
1    0.00      L-Glutamine + D-Fructose 6-phosphate <=> L-Glutamate + D-Glucosamine
        6-phosphate
2    0.25                                ATP + Pyruvate <=> ADP + Phosphoenolpyruvate
3    0.00                                ATP + AMP <=> 2 ADP
4    0.25                                ATP + dTDP <=> ADP + dTTP
5    0.00  ATP + 5-Fluorouridine diphosphate <=> ADP + 5-Fluorouridine triphosphate
6    0.25                                ATP + UDP <=> ADP + UTP
```

The output is a data frame with the reactions that were found to fill the gaps in the model, with the corresponding "additionCost" calculated for each one.

Compatibility

In order to provide compatibility, **g2f** implements **'getReactionsList'** a function that helps to extract the reactions of a **Sybil** model as a list of strings, each string **being** a reaction, which is the input format of **'gapFill'** accepts.

In the examples before, we used a reduced version for the reference organism of E.coli from KEGG. Now we will use a converted model to SBML using KEGG2SBML (Moutselos et al., 2009) from (Akiya Jouraku and Kitano, 2008), which will be converted into **Sybil** with the help of the **'sybilSBML'** package, and then the reactions list will be extracted to use them with the **'gapFill'** function. Note that we have done this because the name of the reaction metabolites in the model should be the same as the ones used in KEGG, and the E.coli core metabolic model included in the **'sybil'** package does not meet this requirement.

```
# Install and import sybilSBML package
R> install.packages('sybilSBML')
R> library('sybilSBML')

# Read the SBML and convert it to sybil
R> mod <- readSBMLmod("eco/eco00730.xml", bndCond = FALSE)

# Extract the model's reactions
R> react <- getReactionsList(mod)

# Fill the gaps
R> reactionsAdded <- gapFill(reactionList = react$react,
+   reference = e.coli$reaction,
```



```
limit = 1/4
)

20% gaps filled in the last iteration
0% gaps filled in the last iteration
0% gaps filled in the last iteration
0% gaps filled in the last iteration
0% gaps filled in the last iteration
addCost                                react
1      0                                ATP + ADP <=> ADP + ATP
2      0 ATP + H2O <=> ADP + Orthophosphate
```

g2f performance

We tested the performance of **g2f** against the most used platforms for gap-filling in metabolic networks using a computer with i7 8750h 2.2GHz processor and 12Gb DDR4 Ram. We compared the performance of R package **g2f**, Python **cobrapy** **gapfill** function and Matlab **COBRA** **fastgapfilling** function (Table 4). The benchmark was performed for each gap-filling algorithm by deleting 10 random reactions across the E. coli core model (Orth et al., 2010).

Table 4: Performance of g2f compared with other gap-filling algorithms. The limit is associated with the threshold for the limit of gap-filling. TicToc was the methodological approach used to measure the performance time. Solution is the capacity of the model to run an FBA after the gap fill function was run. A single iteration of the gap-filling algorithm Cobrapy-"gapfill" was unable to generate a suitable FBA.

Platform	Limit	TicToc (sec)	Solution
R: g2f – "gapfill"	0.1	2.83	Feasible
	0.15	2.76	
	0.2	2.73	
	0.25	6.91	
Python: Cobrapy – "gapfill"	-	1.369	Unfeasible
Matlab: COBRA – "fastgapfill" [Cplex solver]	0.1	7.858	Feasible
	0.15	8.836	
	0.2	9.001	
	0.25	5.695	

Considering the computational performance and flux recovery across the network (FBA solution), **g2f** arises as a suitable method for Genome-scale metabolic network reconstructions gap-filling using curated models as reference.

Application

A wide variety of open-source, paid software and webtools have been developed to fill the gaps in automated or manual metabolic reconstructions (Karp et al., 2018; Machado et al., 2018; Prigent et al., 2017). Performing a gap-filling accurately is a challenging task considering the possibility of overestimating reaction addition or by excluding metabolites from the filling by inqurate thresholds (Pan and Reed, 2018). **g2f** offers an R based open-source alternative capable of integrating with systems biology packages such as "Sybil" (Gelius-Dietrich et al., 2013) or "MinVal" (Osorio et al., 2017) as well as big projects such as Recon3D (Brunk et al., 2018) or the Human Metabolic Atlas (Pornputtapong et al., 2015). Finally, considering that the majority of metabolic models are derived from annotated genomes where not all the enzymes are known, **g2f** offers the possibility to optimize the topology of public available metabolic models or automated metabolic reconstructions.

Conclusions

We developed **g2f**, a novel R package to find dead-end metabolites in a genome-scale metabolic reconstruction and fill the reaction gaps with metabolites available in a stoichiometric matrix from a reference model. Additionally, **g2f** filters the candidate reactions using a weighting function and a user-defined limit. We depicted the functions included in the package using the E. coli reference

model downloaded from the KEGG database^x and the core metabolic model included in the **(sybil)** package. Finally, the performance of **g2f** was compared with other gap^{bold}-filling algorithms (**Cobrapy**–**"gapfill"** and Matlab: **COBRA**–**"fastgapfill"**), showing an adequate feasibility and performance speed.

Acknowledgements

This work was supported by the Pontificia Universidad Javeriana, Bogotá, Colombia^x and Minciencias IDs 7740, 8845^x and 20304 to JG. We thank ~~to~~ the anonymous reviewers and testers for their helpful comments and suggestions to improve the CRAN package.

Availability and implementation

g2f requires R version 2.10 or higher. Source code and documentation are available in the **g2f** accession of the CRAN database. Development releases of the package are available on the **GitHub repository**.

Summary } It is a little awkward to have summary here (after acknowledgment, etc.)

Dead^x-end metabolites are a major drawback in genome^x-scale metabolic reconstruction and analysis. Since there is a lack of available tools to solve this situation in the R environment, hereby³ we introduce the **g2f** package to find and fill dead-end metabolites in a given reconstruction based on a reference template. Our method allows users to filter candidate reactions using a weighting function and a user-defined limit. We show step by step the functionality of each procedure included in the package using a reference model downloaded from the KEGG database for *Escherichia coli*^x and the core metabolic model included in the **(sybil)** 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):e1002980, Mar. 2013. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1002980. URL <https://dx.plos.org/10.1371/journal.pcbi.1002980>. [p]
- A. F. Akiya Jouraku, Nobuyuki Ohta and H. Kitano. systems-biology, 2008. URL <http://www.systems-biology.org/001/001.html>. [p]
- H. Alper, Y.-S. Jin, J. Moxley, and G. Stephanopoulos. Identifying gene targets for the metabolic engineering of lycopene biosynthesis in *Escherichia coli*. *Metabolic Engineering*, 7(3):155–164, May 2005. ISSN 10967176. doi: 10.1016/j.ymben.2004.12.003. URL <https://linkinghub.elsevier.com/retrieve/pii/S1096717604000849>. [p]
- M. Ataman, D. F. Hernandez Gardiol, G. Fengos, and V. Hatzimanikatis. redGEM: Systematic reduction and analysis of genome-scale metabolic reconstructions for development of consistent core metabolic models. *PLoS Computational Biology*, 13(7), 2017. ISSN 1553-734X. doi: 10.1371/journal.pcbi.1005444. URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5519011/>. [p]
- A. Bateman. Curators of the world unite: the International Society of Biocuration. *Bioinformatics*, 26(8): 991–991, Apr. 2010. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/btq101. URL <https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btq101>. [p]
- E. Brunk, S. Sahoo, D. C. Zielinski, A. Altunkaya, A. Dräger, N. Mih, F. Gatto, A. Nilsson, G. A. Preciat Gonzalez, M. K. Aurich, A. Prlić, A. Sastry, A. D. Danielsdottir, A. Heinken, A. Noronha, P. W. Rose, S. K. Burley, R. M. T. Fleming, J. Nielsen, I. Thiele, and B. O. Palsson. Recon3d enables a three-dimensional view of gene variation in human metabolism. *Nature Biotechnology*, 36(3): 272–281, Mar. 2018. ISSN 1087-0156, 1546-1696. doi: 10.1038/nbt.4072. URL <http://www.nature.com/articles/nbt.4072>. [p]
- N. Chen, I. J. d. Val, S. Kyriakopoulos, K. M. Polizzi, and C. Kontoravdi. Metabolic network reconstruction: advances in in silico interpretation of analytical information. *Current Opinion in Biotechnology*, 23(1):77–82, Feb. 2012. ISSN 09581669. doi: 10.1016/j.copbio.2011.10.015. URL <https://linkinghub.elsevier.com/retrieve/pii/S0958166911007129>. [p]

- D. J. Cook and J. Nielsen. Genome-scale metabolic models applied to human health and disease. *WIREs Systems Biology and Medicine*, 9(6):e1393, 2017. ISSN 1939-005X. doi: 10.1002/wsbm.1393. URL <http://onlinelibrary.wiley.com/doi/abs/10.1002/wsbm.1393>. [p]
- S. S. Fong, A. P. Burgard, C. D. Herring, E. M. Knight, F. R. Blattner, C. D. Maranas, and B. O. Palsson. In silico design and adaptive evolution of *Escherichia coli* for production of lactic acid. *Biotechnology and Bioengineering*, 91(5):643–648, Sept. 2005. ISSN 00063592, 10970290. doi: 10.1002/bit.20542. URL <http://doi.wiley.com/10.1002/bit.20542>. [p]
- G. Gelius-Dietrich, A. Desouki, C. Fritzemeier, and M. J. Lercher. sybil – Efficient constraint-based modelling in R. *BMC Systems Biology*, 7(1):125, 2013. ISSN 1752-0509. doi: 10.1186/1752-0509-7-125. URL <http://bmcsystbiol.biomedcentral.com/articles/10.1186/1752-0509-7-125>. [p]
- B. D. Heavner and N. D. Price. Transparency in metabolic network reconstruction enables scalable biological discovery. *Current Opinion in Biotechnology*, 34:105–109, Aug. 2015. ISSN 09581669. doi: 10.1016/j.copbio.2014.12.010. URL <https://linkinghub.elsevier.com/retrieve/pii/S0958166914002250>. [p]
- K. Hornik. The Comprehensive R Archive Network: The Comprehensive R Archive Network. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(4):394–398, July 2012. ISSN 19395108. doi: 10.1002/wics.1212. URL <http://doi.wiley.com/10.1002/wics.1212>. [p]
- Z. Hosseini and S.-A. Marashi. Discovering missing reactions of metabolic networks by using gene co-expression data. *Scientific Reports*, 7(1):41774, Mar. 2017. ISSN 2045-2322. doi: 10.1038/srep41774. URL <http://www.nature.com/articles/srep41774>. [p]
- D. Howe, M. Costanzo, P. Fey, T. Gojobori, L. Hannick, W. Hide, D. P. Hill, R. Kania, M. Schaeffer, S. St Pierre, S. Twigger, O. White, and S. Yon Rhee. The future of biocuration. *Nature*, 455(7209):47–50, 2008. ISSN 1476-4687. doi: 10.1038/455047a. URL <https://www.nature.com/articles/455047a>. [p]
- M. Kanehisa. From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Research*, 34(90001):D354–D357, Jan 2006. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gkj102. URL <https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkj102>. [p]
- M. Kanehisa, Y. Sato, M. Kawashima, and M. Furumichi. KEGG as a reference resource for gene and protein annotation. *Nucleic Acids Research*, 44(D1):D457–D462, Jan. 2016. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gkv1070. URL <https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkv1070>. [p]
- P. D. Karp, D. Weaver, and M. Latendresse. How accurate is automated gap filling of metabolic models? *BMC Systems Biology*, 12(1):73, Dec. 2018. ISSN 1752-0509. doi: 10.1186/s12918-018-0593-7. URL <https://bmcsystbiol.biomedcentral.com/articles/10.1186/s12918-018-0593-7>. [p]
- V. S. Kumar and C. D. Maranas. GrowMatch: An Automated Method for Reconciling In Silico/In Vivo Growth Predictions. *PLoS Computational Biology*, 5(3):e1000308, Mar. 2009. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1000308. URL <https://dx.plos.org/10.1371/journal.pcbi.1000308>. [p]
- 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, Jan. 2014. ISSN 1467-5463, 1477-4054. doi: 10.1093/bib/bbs069. URL <https://academic.oup.com/bib/article-lookup/doi/10.1093/bib/bbs069>. [p]
- D. Machado, S. Andrejev, M. Tramontano, and K. R. Patil. Fast automated reconstruction of genome-scale metabolic models for microbial species and communities. *Nucleic Acids Research*, 46(15):7542–7553, Sept. 2018. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gky537. URL <https://academic.oup.com/nar/article/46/15/7542/5042022>. [p]
- C. A. Martín-Jiménez, D. Salazar-Barreto, G. E. Barreto, and J. González. Genome-Scale Reconstruction of the Human Astrocyte Metabolic Network. *Frontiers in Aging Neuroscience*, 9, Feb. 2017. ISSN 1663-4365. doi: 10.3389/fnagi.2017.00023. URL <http://journal.frontiersin.org/article/10.3389/fnagi.2017.00023/full>. [p]
- K. Moutselos, I. Kanaris, A. Chatziioannou, I. Maglogiannis, and F. N. Kolisis. KEGGconverter: a tool for the in-silico modelling of metabolic networks of the KEGG pathways database. *BMC Bioinformatics*, 10(1):324, 2009. ISSN 1471-2105. doi: 10.1186/1471-2105-10-324. URL <https://doi.org/10.1186/1471-2105-10-324>. [p]

- J. D. Orth, B. Palsson, and R. M. T. Fleming. Reconstruction and Use of Microbial Metabolic Networks: the Core Escherichia coli Metabolic Model as an Educational Guide. *EcoSal Plus*, 4(1), Sep 2010. ISSN 2324-6200. doi: 10.1128/ecosalplus.10.2.1. URL <http://www.asmscience.org/content/journal/ecosalplus/10.1128/ecosalplus.10.2.1>. [p]
- D. Osorio, J. González, and A. Pinzón. minval: An R package for MINimal VALidation of Stoichiometric Reactions. *The R Journal*, 9(1):114, 2017. ISSN 2073-4859. doi: 10.32614/RJ-2017-031. URL <https://journal.r-project.org/archive/2017/RJ-2017-031/index.html>. [p]
- B. Palsson. Metabolic systems biology. *FEBS Letters*, 583(24):3900–3904, Dec. 2009. ISSN 00145793. doi: 10.1016/j.febslet.2009.09.031. URL <http://doi.wiley.com/10.1016/j.febslet.2009.09.031>. [p]
- S. Pan and J. L. Reed. Advances in gap-filling genome-scale metabolic models and model-driven experiments lead to novel metabolic discoveries. *Current Opinion in Biotechnology*, 51:103–108, June 2018. ISSN 09581669. doi: 10.1016/j.copbio.2017.12.012. URL <https://linkinghub.elsevier.com/retrieve/pii/S0958166917302045>. [p]
- N. Pham, R. van Heck, J. van Dam, P. Schaap, E. Saccenti, and M. Suarez-Diez. Consistency, Inconsistency, and Ambiguity of Metabolite Names in Biochemical Databases Used for Genome-Scale Metabolic Modelling. *Metabolites*, 9(2):28, Feb. 2019. ISSN 2218-1989. doi: 10.3390/metabo9020028. URL <http://www.mdpi.com/2218-1989/9/2/28>. [p]
- N. Pornputtapong, I. Nookaew, and J. Nielsen. Human metabolic atlas: an online resource for human metabolism. *Database*, 2015, Jan. 2015. ISSN 1758-0463. doi: 10.1093/database/bav068. URL <https://academic.oup.com/database/article/doi/10.1093/database/bav068/2433201>. [p]
- S. Prigent, C. Frioux, S. M. Dittami, S. Thiele, A. Larhlimi, G. Collet, F. Gutknecht, J. Got, D. Eveillard, J. Bourdon, F. Plewniak, T. Tonon, and A. Siegel. Meneco, a Topology-Based Gap-Filling Tool Applicable to Degraded Genome-Wide Metabolic Networks. *PLOS Computational Biology*, 13(1):e1005276, Jan. 2017. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1005276. URL <https://dx.plos.org/10.1371/journal.pcbi.1005276>. [p]
- J. L. Reed, T. R. Patel, K. H. Chen, A. R. Joyce, M. K. Applebee, C. D. Herring, O. T. Bui, E. M. Knight, S. S. Fong, and B. O. Palsson. Systems approach to refining genome annotation. *Proceedings of the National Academy of Sciences*, 103(46):17480–17484, Nov. 2006. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.0603364103. URL <http://www.pnas.org/cgi/doi/10.1073/pnas.0603364103>. [p]
- V. Satish Kumar, M. S. Dasika, and C. D. Maranas. Optimization based automated curation of metabolic reconstructions. *BMC Bioinformatics*, 8(1):212, 2007. ISSN 14712105. doi: 10.1186/1471-2105-8-212. URL <http://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-8-212>. [p]
- A. Schultz and A. A. Qutub. Reconstruction of Tissue-Specific Metabolic Networks Using CORDA. *PLOS Computational Biology*, 12(3):e1004808, Mar. 2016. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1004808. URL <http://dx.plos.org/10.1371/journal.pcbi.1004808>. [p]
- B. Szappanos, K. Kovács, B. Szamecz, F. Honti, M. Costanzo, A. Baryshnikova, G. Gelius-Dietrich, M. J. Lercher, M. Jelasity, C. L. Myers, B. J. Andrews, C. Boone, S. G. Oliver, C. Pál, and B. Papp. An integrated approach to characterize genetic interaction networks in yeast metabolism. *Nature Genetics*, 43(7):656–662, July 2011. ISSN 1061-4036, 1546-1718. doi: 10.1038/ng.846. URL <http://www.nature.com/articles/ng.846>. [p]
- I. Thiele and B. Palsson. A protocol for generating a high-quality genome-scale metabolic reconstruction. *Nature Protocols*, 5(1):93–121, Jan. 2010. ISSN 1754-2189, 1750-2799. doi: 10.1038/nprot.2009.203. URL <http://www.nature.com/articles/nprot.2009.203>. [p]
- I. Thiele, N. Vlassis, and R. M. T. Fleming. fastGapFill: efficient gap filling in metabolic networks. *Bioinformatics*, 30(17):2529–2531, Sept. 2014. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/btu321. URL <https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btu321>. [p]
- L. van Steijn, F. J. Verbeek, H. P. Spaink, and R. M. Merks. Predicting metabolism from gene expression in an improved whole-genome metabolic network model of danio rerio. *Zebrafish*, 16(4):348–362, 2019. ISSN 1545-8547. doi: 10.1089/zeb.2018.1712. URL <https://www.liebertpub.com/doi/10.1089/zeb.2018.1712>. Publisher: Mary Ann Liebert, Inc., publishers. [p]
- H. Wang and S. Marci. RAVEN 2.0: A versatile toolbox for metabolic network reconstruction and a case study on Streptomyces coelicolor. *PLOS Computational Biology*, page 17, 2018. [p]

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

dcosorih@unal.edu.co

Andrés Pinzón Velasco

Grupo de Investigación en Bioinformática y Biología de Sistemas. Instituto de Genética, Universidad Nacional de Colombia.

Bogotá

Colombia

ampinzonv@unal.edu.co

Nicolás Mendoza-Mejía

Grupo de Investigación en Bioquímica Experimental y Computacional. Departamento de Nutrición y Bioquímica, Facultad de Ciencias, Pontificia Universidad Javeriana.

Bogotá

Colombia

mendozamejian@gmail.com

Felipe Rojas-Rodriguez

Division of Molecular Pathology, The Netherlands Cancer Institute - Antoni van Leeuwenhoek Hospital.

Amsterdam 1066 CX

The Netherlands.

f.rojas@nki.nl

Janneth González

Grupo de Investigación en Bioquímica Experimental y Computacional. Departamento de Nutrición y Bioquímica, Facultad de Ciencias, Pontificia Universidad Javeriana.

Bogotá

Colombia

ORCID: 0000-0003-2009-3374

janneth.gonzalez@javeriana.edu.co