BioISO: An Objective-Oriented Application for Assisting the Curation of Genome-Scale Metabolic Models

Fernando Cruz, João Capela, Eugénio C. Ferreira, Miguel Rocha, and Oscar Dias

***Abstract*—As the reconstruction of Genome-Scale Metabolic Models (GEMs) becomes standard practice in systems biology, the number of organisms having at least one metabolic model is peaking at an unprecedented scale. The automation of laborious tasks, such as gap-finding and gap-filling, allowed the development of GEMs for poorly described organisms. However, the quality of these models can be compromised by the automation of several steps, which may lead to erroneous phenotype simulations. Biological networks constraint-based In Silico Optimisation (*BioISO*) is a computational tool aimed at accelerating the reconstruction of GEMs. This tool facilitates manual curation steps by reducing the large search spaces often met when debugging *in silico* biological models. *BioISO* uses a recursive relation-like algorithm and Flux Balance Analysis (FBA) to evaluate and guide debugging of *in silico* phenotype simulations. The potential of *BioISO* to guide the debugging of model reconstructions was showcased and compared with the results of two other state-of-the-art gap-filling tools (*Meneco* and *fastGapFill*). In this assessment, *BioISO* is better suited to reducing the search space for errors and gaps in metabolic networks by identifying smaller ratios of dead-end metabolites. Furthermore, *BioISO* was used as *Meneco's* gap-finding algorithm to reduce the number of proposed solutions for filling the gaps.**

***Index Terms*—*BioISO*, gap-finding algorithm, genome-scale metabolic models, open-source software, python.**

I Introduction

GENOME-SCALE Metabolic Models (GEMs) are becoming standard practice in systems biology. GEMs can be used to simulate the organism's phenotype under different environmental and genetic conditions [1], [2], [3]. Flux Balance Analysis (FBA) [4], or related constraint-based methods, are used for solving linear programming problems outlined by constraints imposed over the stoichiometric model.

Nevertheless, reconstructing GEMs is still challenging [1], as model validation and manual curation can be laborious tasks [3]. Most bottlenecks derive from accumulated errors, which require complex and unique solutions. For instance, when a metabolic network is converted into a stoichiometric model, FBA often mispredicts the organism's experimental growth rate due to errors like missing or blocked reactions and dead-end metabolites (gaps), among others.

The reconstruction of GEMs can follow two diverse paradigms: bottom-up [1] and top-down [5].

The fast and automated top-down paradigm does not resort to gap-filling procedures. This approach consists of reconstructing a universal GEM that has been curated previously for most common errors [5]. This universal simulation-ready model is then converted to an organism-specific model by carving reactions and metabolites for which evidence is missing. Thus, the top-down paradigm can be extremely useful to create microbial community models by merging the automated single-species models into community-scale networks [5]. Nevertheless, it is still unclear whether single-species models' phenotype simulations are unreasonably biased by the universal GEM.

Unlike the top-down paradigm, the widely-used bottom-up paradigm consists of four main steps: draft reconstruction based on genome functional annotation; refinement and curation of the draft reconstruction; conversion to stoichiometric model; model validation [1]. The last steps of a bottom-up reconstruction usually include several time-consuming and repetitive tasks to fix errors that emerged during the draft reconstruction, thus solving the discrepancy between the predicted phenotype and experimental results. While mistakes can be solved using manual curation, there are several gap-find and gap-fill tools to accelerate the debugging process. Gap-find algorithms aim to find either missing or blocked reactions and dead-end metabolites in a draft reconstruction, whereas gap-fill ones are responsible for finding potential solutions to the errors mentioned above.

Most state-of-the-art tools for debugging draft reconstructions comprehend both automated gap-finding and gap-filling procedures [6], [7], [8], [9]. Nevertheless, there are other tools developed for only one of these procedures [10], [11], [12].

Manuscript received 20 December 2021; revised 3 August 2023; accepted 23 November 2023. Date of publication 3 January 2024; date of current version 3 April 2024. The work of Fernando Cruz was supported by the FCT under Grant SFRH/BD/139198/2018. The work of João Capela was supported by the FCT under Grant 2021.08789.BD. The work of Oscar Dias was supported by FCT for the Assistant Research contract obtained under CEEC Individual 2018. This work was supported in part by the Portuguese Foundation for Science and Technology (FCT) under Grant UIDB/04469/2020, in part by Biodata.pt. Infraestrutura Portuguesa de Dados Biológicos under Grant 22231/01/SAICT/2016, in part by the Lisboa Portugal Regional Operational Programme under Grant Lisboa2020, and in part by the Portugal 2020 Partnership Agreement, through the European Regional Development Fund (ERDF) [DOI: 03425/2018/CP1581/CT0020]. *(Corresponding author: Oscar Dias.)*

The authors are with the Centre of Biological Engineering, University of Minho, 4710-057 Braga, Portugal. (e-mail: fernando.cruz@ceb.uminho.pt; joao.capela@ceb.uminho.pt; ecferreira@deb.uminho.pt; mrocha@di.uminho.pt; odias@deb.uminho.pt).

BioISO was implemented as Python package, and it is also available at https://bioiso.bio.di.uminho.pt as a web-service and in *merlin* as a plugin.

This article has supplementary downloadable material available at https://doi.org/10.1109/TCBB.2023.3339972, provided by the authors.

Digital Object Identifier 10.1109/TCBB.2023.3339972

Besides, gap-find and gap-fill tools can also be separated according to the gap-finding and gap-filling methodology, respectively.

Regarding gap-finding algorithms, tools such as *biomassPrecursorCheck* [10], and *Meneco* [6] are based on guided-search algorithms to identify gaps or errors directly associated with a given objective/reaction. Both tools check the metabolic network topological features to find gaps, asserting the existence of a given metabolite's predecessors and successors. The *COBRA Toolbox*'s *BiomassPrecursorCheck* tool searches for predecessors immediately upstream of the biomass reaction of a given model, whereas *Meneco* performs the gap-search according to a set of seed and target metabolites provided as input. However, the search depth of the latter may encompass the whole metabolic network.

On the other hand, *gapFind/gapFill* [7], *fastGapFill* [8], and *Gauge* [9] are based on exhaustive searches. Thus, these methodologies identify gaps all over the metabolic network, regardless of a given objective. *GapFind/gapFill* and *fastGapfill* highlight gaps using a stoichiometry-like approach. These methods search the stoichiometric matrix for no-production and no-consumption metabolites. Alternatively, *Gauge* combines Flux Coupling Analysis and gene expression data to propose gaps in a draft GEM.

Regarding gap-fill, all available tools require a dataset of metabolic reactions, usually retrieved from a biochemical database (e.g., KEGG [13], BiGG [14], or MetaCyc [15]), to resolve metabolic gaps [6], [7], [8], [9], [11], [12],. Besides a database of metabolic reactions, both *Gauge* [9] and *Mirage* [12] require gene expression data. *Smiley* [11] relies on additional growth phenotype data to identify minimal environmental conditions for which the model mispredicted growth and non-growth phenotypes. *Meneco*, *gapFind/gapFill*, *fastGapFill*, *Gauge,* and *Smiley* consider the minimal reaction set of the whole dataset to resolve every single gap. Alternatively, *Mirage* considers a pan-metabolic network that assures flux through all metabolites, followed by a pruning step to reduce the large set of solutions. Thus, the solution set is often the result of two very different gap-filling approaches, namely the parsimonious and pruning approaches.

Most state-of-the-art tools for debugging draft reconstructions rely on proprietary software, such as MATLAB (Mathworks) or GAMS. From the above-mentioned tools, *Meneco* is the only freely available to the community, as it is available as a Python package. It is worth noticing, though, that all tools require coding skills. More importantly, most tools require and return excessively verbose outputs, such as large arrays of missing metabolites and even greater sets of potential solutions. The analysis of these results can be challenging for wet-lab scientists without coding skills or data analysis expertise.

Furthermore, gap-filling tools usually warn that gaps might result from missing mappings between the metabolites' abbreviations and the reference database identifiers. Besides the mapping's limitations, several tools require different format-files for the metabolic data, such as SBML (e.g., Meneco), KEGG reaction database *lst* format file (e.g., *fastGapFill*), customised text files (e.g., *gapFind/gapFill*), or data structures (e.g., *Gauge*). On the other hand, other tools lack information on how a different source of solutions can be used (e.g., *Mirage* and *Smiley*).

Several gap-find and gap-fill state-of-the-art tools have been described with further detail being given in the Appendix A and B, available online.

The introduction of artefacts in metabolic networks can hinder GEM's applications, such as metabolic engineering and drug targeting tasks. These issues may be extremely relevant for organisms that have evolved due to a combination of extensive loss-of-function events and acquisition of key genes, via horizontal gene transfer during co-evolution with well-defined and constant ecological niches [16], [17], [18]. Moreover, although loss-of-function genetic variants are frequently associated with severe clinical phenotypes, several events are also present in healthy individuals' genomes, making it essential to assess their impact [19].

Hence, automated approaches, and especially gap-fill tools, must be used very carefully, taking into consideration the issues raised above. Otherwise, the offered automation can be a counterproductive solution for the manual curation steps performed during high-quality reconstructions. Furthermore, the usability of gap-fill approaches can be vastly improved.

To the best of our knowledge, the reconstruction of high-quality GEMs is frequently based on a parsimonious bottom-up approach involving manual curation and human intervention. In our view of a parsimonious bottom-up reconstruction, the metabolic network can be divided into smaller, yet insightful, modules based on the phenotype being studied. Then, recursive relations can be used to divide metabolic networks into smaller modules directly associated with the objective phenotype. FBA simulations applied over surrogate reactions designed explicitly for each module can unveil the minor manual curation tasks that often increment the reconstruction's quality and resolve the metabolic gaps for such module.

With this methodology in mind, Biological networks constraint-based *In Silico* Optimisation (*BioISO*) was designed to automatise the search for reactions and metabolites associated with a given objective, narrowing the search space. *BioISO* is the only tool combining a guided search with the evaluation of dead-end metabolites using FBA over surrogate reactions. More importantly, *BioISO* is a user-friendly gap-finding tool that allows users to fastly analyse gaps and errors. The results are then presented in a graphical user interface embedded in both a webserver and *merlin*, so that the debugging process can be easy to follow and repeat.

II Implementation

A BioISO*'s Algorithm*

BioISO requires a constraint-based model and the reaction to be evaluated, which defines the linear programming problem's objective function. A recursive relation-like algorithm is then used to build a hierarchical structure according to the metabolites and reactions associated with this objective.

BioISO is herein showcased through the analysis of a small-scale metabolic network, represented in Fig. 1, having 12 intracellular and two extracellular metabolites, 12 reactions, and two

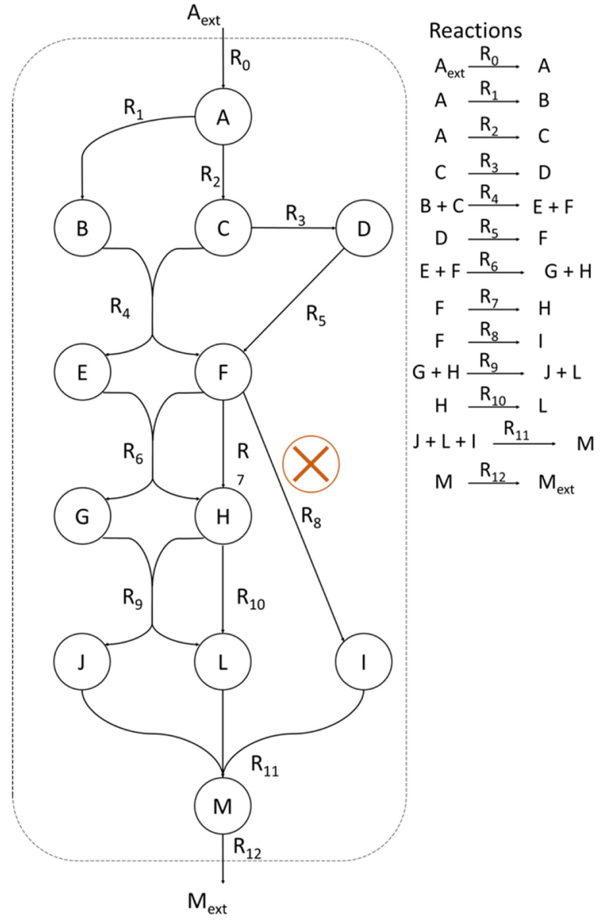

Fig. 1. Small-scale metabolic network. Metabolites and reactions are represented in the metabolic network as white nodes and black directed arrows. The extracellular boundary is represented as a dashed line. The reactions are listed alongside the metabolic network. In this metabolic network, the reaction identified by R_8 is considered missing, blocked, or incorrectly formulated.

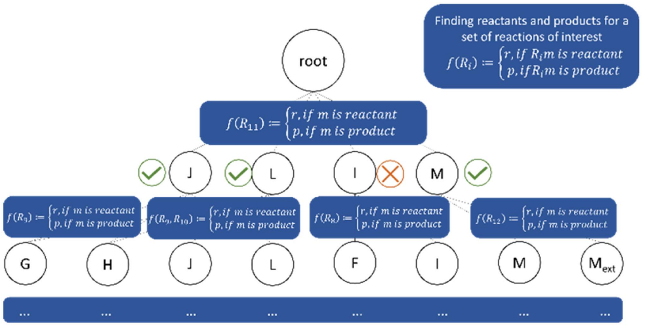
Finding reactants and products for a set of reactions of interest
$f(R_i) := \begin{cases} r, if\ R_i m\ is\ reactant \\ p, if\ R_i m\ is\ product \end{cases}$
root
$f(R_{11}) := \begin{cases} r, if\ m\ is\ reactant \\ p, if\ m\ is\ product \end{cases}$
J
L
I
M
$f(R_8) := \begin{cases} r, if\ m\ is\ reactant \\ p, if\ m\ is\ product \end{cases}$
$f(R_9, R_{10}) := \begin{cases} r, if\ m\ is\ reactant \\ p, if\ m\ is\ product \end{cases}$
$f(R_8) := \begin{cases} r, if\ m\ is\ reactant \\ p, if\ m\ is\ product \end{cases}$
$f(R_{12}) := \begin{cases} r, if\ m\ is\ reactant \\ p, if\ m\ is\ product \end{cases}$
G
H
J
L
F
I
M
M_{ext}
...

Fig. 2. Evaluation of reaction R_{11} with *BioISO*. *BioISO* finds the set of metabolites associated with reaction R_{11}, which in this case corresponds to metabolites *J, L, I,* and *M*. For the next call, *BioISO* finds metabolites *G, H, J, L, F, I, M,* and M_{ext} in the reactions R_8, R_9, R_{10} and R_{12} and so forth. The tick and *X* marks represent metabolites with positive and negative evaluation by *BioISO*, respectively. The functions to identify reactants and products are highlighted in the blue boxes under the metabolite.

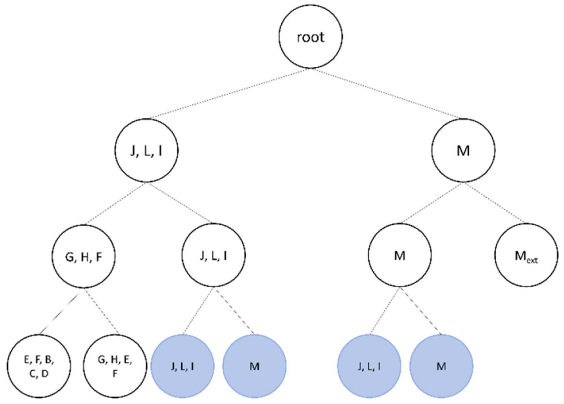

Fig. 3. The hierarchical tree-based structure, imposed by the recursive relation-like computational method implemented in *BioISO*, is outlined. *BioISO* finds the set of precursors and successors in the first level, which in this case correspond to metabolites *J, L, I,* and *M*, respectively. For the next level, *BioISO* finds the precursors *G, H, F* for the previous precursors *J, L, I*, which are also successors of themselves. On the other branches, *M* is its precursor, while M_{ext} is the successor. *BioISO* has implemented a cache memory system of all simulations performed during the recursion. Thus, nodes coloured in blue are only evaluated once, as they were already evaluated in those specific conditions.

compartments (extracellular and intracellular). In this metabolic network, the reaction identified by R_8 is considered missing, blocked, or incorrectly formulated, while identifier R_{11} refers to the reaction to be evaluated.

BioISO starts by finding the set of metabolites associated with the reaction submitted for evaluation, namely reaction R_{11}. The tool will discover metabolites *J, L, I,* and *M* as the subsequent nodes since these metabolites are involved in R_{11} (Fig. 2). A set of reactions is then created for each node and populated with the reactions associated with each metabolite. Thus, *BioISO* will retrieve four sets of reactions, one for each metabolite (Fig. 2).

Meanwhile, *BioISO* assembles a hierarchical tree-based structure, depicted in Fig. 3. The tool identifies as precursors (reactants) or successors (products) the metabolites associated with the submitted reaction (R11). Thus, *J, L,* and *I* (reactants) and *M* (product) involved in rection R_{11} were separated into two different branches: precursors and successors, respectively.

In the next recursive call, *BioISO* retrieves metabolites *G, H, F, M, J, L, I,* and M_{ext} from reactions R_8, R_9, R_{10}, and R_{12} (Fig. 2), while adding the precursors *G, H, F,* and *M*, and the successors *J, L, I,* and M_{ext} to the tree-based structure (Fig. 3). These reactions are either consuming or producing the metabolites identified in the previous step.

The stopping condition, namely *BioISO's* depth, represents the number of recursive calls performed during the metabolic network analysis. For instance, varying *BioISO*'s depth from 1 to 3 allows running the tool from shallow to guided or nearly exhaustive searches, depending on the metabolic network's size and arborescence.

The methodology for finding and assessing metabolites and reactions is detailed in algorithms Appendix C.1-4, available online. The first algorithm, labelled *BioISO* (algorithm Appendix C.1, available online), is the core logic supporting the methodology proposed in this work. *BioISO* uses algorithm Appendix C.2, available online, to find and evaluate (using FBA) reactions associated with nodes. More importantly, *BioISO* uses a more comprehensive approach to assess reactants (precursors) and products (successors), as demonstrated in algorithms Appendix C.3, available online, (*testReactant*) and Appendix C.4, available online, (*testProduct*), respectively.

In detail, a precursor (reactant) is considered a positive assessment if the metabolic model can produce it (connected metabolite). Thus, a reactant is a product elsewhere in the metabolic network; otherwise, it would not be available for the objective reaction. Hence, *BioISO* evaluates an unbalanced reaction explicitly designed to allow the metabolite accumulation in the metabolic model. The evaluation is successful if the model can attain non-zero flux in the FBA solution for this surrogate reaction.

As described in notation (1), when evaluating reaction R_{11}, *BioISO* will evaluate the precursor *I* by adding an unbalanced reaction R_{13}, which takes *I* as a reactant, and whose lower and upper bounds are set to zero and plus-infinity, respectively.

$$\begin{gathered} maximize \;\rightarrow\; v_{13} \\ subject \;\rightarrow\; S * v \;=\; 0 \\ subject \;\rightarrow\; \alpha_j \leq v_j \leq \beta_j,\; j \;=\; 1,\, \ldots,\, N \\ subject \;\rightarrow\; 0 \leq v_{13} \leq \; +\infty \end{gathered} \tag{1}$$

where:

1) v_{13} is the linear objective function for maximisation of reaction R_{13}.
2) *v* is the flux vector.
3) *S* is the stoichiometric matrix (columns represent reaction fluxes and rows the metabolites mass balances).
4) α and β are the lower and upper bounds, respectively.

Furthermore, a similar reaction is included in the model for each reactant to prevent the seldom cases in which all reactants are forcibly produced by reactions that produce/synthesise the assessed metabolite.

Likewise, *BioISO* creates unbalanced reactions that allow the uptake of all products associated with the evaluated reaction. These reactions are included in the model to prevent the unlikely scenario that the model forcibly needs to consume/metabolise such products to synthesise the precursor.

On the other hand, a successor (product) is considered a positive assessment if the metabolic model can consume it (connected metabolite). Thus, a product is a reactant elsewhere in the metabolic network. As described in the testing of precursor *I*, *BioISO* also creates an unbalanced reaction for the successor. However, this reaction is now explicitly designed to allow the metabolite uptake in the metabolic model. Thus, the minimisation of this uptake reaction is now the objective function of the FBA simulation. In other words, the model should metabolise/consume the precursor metabolite, obtaining an optimal non-zero flux solution through the unbalanced reaction.

A detailed description of the *BioISO* workflow to search and assert gaps is provided in the Appendix C, available online.

In short, the procedure to split the objective into two subproblems and evaluate both metabolites and reactions follows the workflow below:

1) collect the reactions associated with each metabolite to be evaluated.
2) maximise/minimise the reactions and assess the outcome of the FBA solution.
3) from such reactions, find the precursor (reactants) and successor (products) metabolites.
4) create unbalanced reactions allowing accumulation or uptake of the metabolites.
5) maximise/minimise the unbalanced reactions and assess the outcome of the FBA solutions.

An analysis of BioISO's relation-like algorithm's complexity, together with the recursion tree method visualisation, is also provided in Appendix C, available online.

B. BioISO*'s Applications*

BioISO is a package developed in Python$^{\text{TM}}$ 3 using the FBA framework implemented in *COBRApy* [20]. *BioISO* relies on *COBRApy* to read GEMs written in the System Biology Markup Language (SBML) [21]. The IBM CPLEX solver (v. 1210) is used by default to solve multiple linear programming problems formulated with the FBA framework, although any solver supported by *COBRApy* can be used. *BioISO*'s source code, validation procedures, and examples can be obtained from our group's GitHub at https://github.com/BioSystemsUM/BioISO.

A Dockerised Flask application has been implemented to make *BioISO* available to all scientific community at https://bioiso.bio.di.uminho.pt. This web service allows users to submit a GEM in the SBML file format and evaluate a specific reaction available in the model. BioISO's web service will then return a user-friendly web page highlighting the metabolic network's blocked reactions and dead-end metabolites according to the submitted reaction. Finally, the user is encouraged to navigate through the set of dead-end metabolites intuitively.

Besides the web service application, *BioISO* is also available as a plugin named *BioISO* for *merlin* [22], an open-source and user-friendly resource that hastens the reconstruction of GEMs. This plugin allows *BioISO* to supply an equally user-friendly view of the errors associated with a given model reconstructed within *merlin*.

Finally, instructions to run *BioISO* in the available applications and interpret the expected results are also available at https://bioiso.bio.di.uminho.pt/tutorial.

III Results

BioISO is aimed at identifying errors that emerge during the bottom-up reconstruction of high-quality GEMs. Errors such as missing or blocked reactions and dead-end metabolites are often met during model debugging and refinement. Thus, *BioISO* is based on a recursive-like algorithm to guide the search for metabolic gaps associated with a given objective. Throughout *BioISO*'s objective-oriented search, multiple FBA simulations are used to assert real metabolic gaps. Hence, we propose a tool capable of reducing large search spaces and asserting real metabolic gaps to accelerate time-consuming and laborious manual curation tasks.

Most state-of-the-art tools for debugging draft reconstructions aim to find and solve a wide range of problems. These tools are commonly used in automatic gap-find and gap-filling routines. For instance, *Meneco*, *gapFind/gapFill*, *fastGapFill*, *Gauge*, *Smiley,* and *Mirage* are gap-fill tools aimed at finding and solving errors accumulated during the draft reconstruction.

GapFind/gapFill, *fastGapFill,* and *Gauge* exhaustive-search tools attempt to assert gaps throughout the whole metabolic network. Then, these tools enumerate minimal solutions (set of reactions) to solve the highlighted gaps. Alternatively, Mirage and Smiley add new reactions to the model without an initial gap-scan, forcing model predictions to match the experimental data.

On the other hand, *Meneco*'s guide-search algorithm searches for gaps according to a set of seed and target metabolites. Then, this tool enumerates a minimal set of reactions that can restore the flux to all dead-end metabolites identified during the topological search.

Likewise, *BioISO* seeks dead-end metabolites downstream and upstream of a user-defined objective. Additionally, *BioISO* performs multiple FBA simulations of custom unbalanced reactions during the topological search to evaluate whether a given metabolite is being consumed or produced.

More importantly, *BioISO* is the only tool freely available to all scientists. That is, *BioISO* is the only user-friendly gap-finding tool, providing a graphical user interface embedded in both a webserver and *merlin*. Thus, our tool allows users to analyse gaps and errors without requiring coding skills or additional metabolic data such as growth phenotype data or biochemical databases. Moreover, *BioISO* is a ready-to-use and relatively fast method, allowing users to run this tool iteratively during model reconstruction.

A summary of all features used to compare *BioISO* with several gap-find and gap-filling tools is available in Appendix A and B, available online.

BioISO's validation includes three assessments:

1) *BioISO's* algorithm depth analysis.
2) Exhaustive-search *versus* guided-search.
3) BioMeneco – embedding *BioISO* in Meneco [6].

The first analysis was aimed at assessing BioISO's shallow, guided, or nearly exhaustive searches for metabolic gaps in five state-of-the-art models. The second analysis allowed us to assess the relevance of guided- and exhaustive-searches for gap-finding. In this assessment, we have compared *BioISO* and *Meneco* guide-searches against *fastGapFill* exhaustive-search. Finally, the last analysis showcases the outcome of setting *BioISO* as Meneco's gap-finding algorithm.

A. *BioISO's Algorithm Depth Analysis*

BioISO was used to analyse several published GEMs for two objective functions: growth and compound production maximisation. The sets of dead-end metabolites and blocked reactions were determined as described in the Implementation section. The workflow and methodology used to assess *BioISO*'s algorithm robustness are described in further detail in the Materials and Methods section together with Appendix D and E, available online.

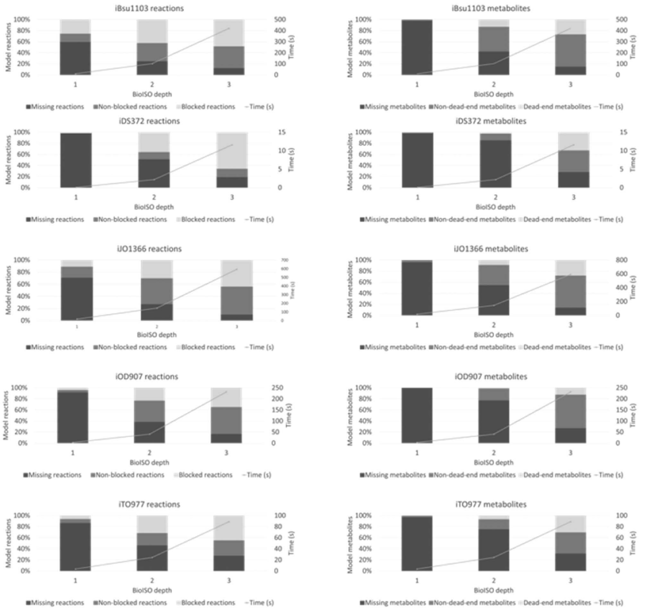

Fig. 4. Summary of the reactions (left panel) and metabolites (right panel) analysed by *BioISO* for published Genome-Scale Metabolic Models. *BioISO* was used to analyse 5 state-of-the-art models (iBsu1103, iDS372, iJO1366, iOD907, and iTO977) with added gaps. *BioISO's* analysis included different algorithm settings, namely varying the depth from 1 to 3 for each objective function, which will control the number of recursive calls for precursors and successors. This allowed running *BioISO's* algorithm for shallow, guided, or nearly exhaustive searches, depending on the size and arborescence of the metabolic network. *BioISO*'s computation time was recorded in seconds (s) together with missing (non-covered by *BioISO*) reactions and metabolites, non- and dead-end metabolites, non- and blocked reactions.

BioISO's analysis included different settings, namely varying the algorithm's depth from 1 to 3, allowing to set *BioISO* for shallow, guided, or nearly exhaustive searches.

BioISO's depth level of 1 assesses the nearest neighbours (successors and precursors metabolites) and their associated reactions. According to Fig. 4, *BioISO* analyses less than 50% of all reactions for growth maximisation. The number of blocked reactions is significantly reduced at depth level 1 (less than 12%), except for the iJO1366 [23] and iBsu1103 [24] models (Fig. 4 and Appendix E.1-2, available online). Likewise, *BioISO* only covers less than 10% of all metabolites for growth maximisation (Fig. 4). As a result, the number of dead-end metabolites found by *BioISO* at a depth level of 1 is less than 3 for all models (Fig. 4 and Appendix E.1-2, available online). The level of insight provided by *BioISO* for shallow searches is significantly reduced and similar to the *biomassPrecursorsCheck* tool from *COBRA Toolbox* [10] or *Meneco* [6].

Increasing the depth level to 2 allows evaluating more reactions. As demonstrated in Fig. 4, 50% or more of the reactions are assessed in all models. Whereas *BioISO* analyses nearly a quarter of all metabolites in the iOD907 [24] and iTO977 [25] models, this coverage increases up to 60% in the iJO1366 [25] and iBsu1103 [24] models. In contrast, only 15% of all metabolites have been covered by *BioISO* in the iDS372 [26] model. *BioISO* also detects more blocked reactions and dead-end metabolites for guided searches. The percentage of blocked reactions varies between 20% and 40%, and the percentage of dead-end metabolites between 2% and 12% (Fig. 4 and Appendix E.1-2, available online).

At a depth level of 3, a nearly exhaustive search is performed by *BioISO* evaluating most yet not all reactions and metabolites in the metabolic networks. In detail, *BioISO* analyses more than 70% of both metabolites and reactions for the growth maximisation (Fig. 4 and Appendix E.1-2, available online). Likewise, the percentage of detected blocked reactions and dead-end metabolites increases up to 65% and 30%, respectively.

As detailed in Appendix E.3-4, available online, similar results were obtained for the maximisation of compound production. However, the number of metabolites covered in the iTO977 model is considerably lower than the remaining models at depth levels of 2 and 3 (Appendix E.4, available online).

Fig. 4 also presents *BioISO's* computation time for each model during growth maximisation as a function of the depth level. *BioISO* was considerably faster for shallow searches (depth level of 1) in all models for growth (Fig. 4 and Appendix E.1-2, available online) and compound production (Appendix E.3-4, available online) maximisation. According to Fig. 4, *BioISO* required computation time for a depth level of 2 varies between 2 and 144 seconds during growth maximisation. During the compound production maximisation, *BioISO* takes between 4 and 74 seconds (Appendix E.3-4, available online). The computation time of *BioISO* increases significantly at the depth level of 3. At this depth, *BioISO's* computation time can attain around 600 and 405 seconds when maximising growth (Fig. 4 and Appendix E.1-2, available online) and compound production (Appendix E.3-4, available online), respectively.

Hence, *BioISO's* computation time is significantly dependent on the size of the covered search space. In turn, the covered search space increases with the depth of search and the model's size. Although navigating the network throughout the new metabolites and reactions might not be time-consuming, evaluating numerous metabolites and reactions using the FBA framework requires time.

The dead-end metabolites and blocked reactions ratios were calculated as denoted in (2), (3), (4) and (5) of the Materials and Methods section. Fig. 5 highlights the ratios of dead-end metabolites obtained for both growth and compound production analysis in all models. Both dead-end metabolites and blocked reactions ratios are also available in the Appendix E.1-4, available online.

At a depth level of 1, the ratio of blocked reactions for the objective-oriented search (br_{ooss}) varies between 0.3 and 0.9. In contrast, the homologous ratio for the whole-space search (br_{wss}) varies between 0.02 and 0.25 (Appendix E.1-4, available online).

Regarding the ratios of dead-end metabolites, *BioISO* attains markedly small ratios for the whole-space search *(*dem_{wss}*)* at a depth level of 1, namely obtaining ratios smaller than 0.1 in all models for both objective functions (Fig. 5 and Appendix E.1-4, available online). However, the ratio of dead-end metabolites for the objective-oriented search *(*dem_{ooss}*)* can peak up to 0.35 (Fig. 5 and Appendix E.1-2, available online) and 0.85 (Fig. 5 and Appendix E.6-4, available online) in the growth and compound production analysis, respectively.

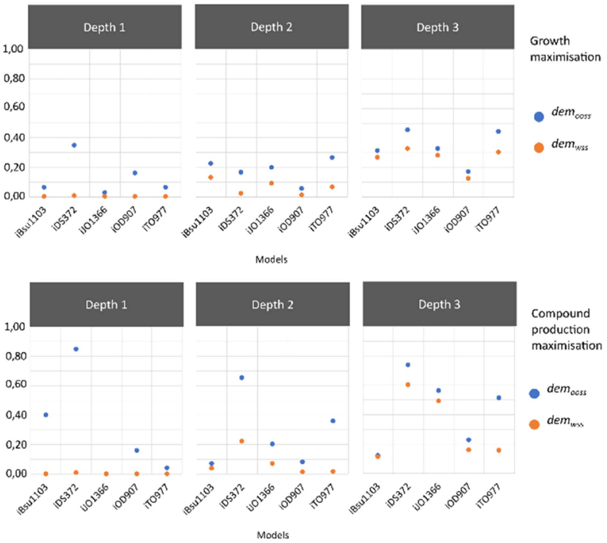

Fig. 5. Calculated ratios of dead-end metabolites for the maximisation of growth (upper panel) and compound production (bottom panel). *BioISO* was used to analyse 5 state-of-the-art models (iBsu1103, iDS372, iJO1366, iOD907, and iTO977) with added gaps. *BioISO's* analysis included different algorithm settings, namely varying the depth from 1 to 3 for each objective function, which will control the number of recursive calls for precursors and successors. This allowed running *BioISO's* algorithm for shallow, guided, or nearly exhaustive searches, depending on the size and arborescence of the metabolic network. dem_{ooss} and dem_{wss} stand for the ratios of dead-end metabolites for the objective-oriented search and whole search spaces.

The br_{wss} ratios increase significantly when raising the depth to level 2 (*BioISO* guided search), whereas br_{ooss} ones remain roughly the same as in the previous level. The br_{wss} ratio can vary from 0.23 to 0.43 (Appendix E.1-2, available online) and from 0.23 to 0.36 (Appendix E.3-4, available online) for the growth and compound production analysis, respectively.

The dem_{ooss} ratio tends to increase as a response to *BioISO*'s guided search (depth level of 2) in the iBsu1103, iJO1366, and iTO977 models during the growth maximisation analysis (Fig. 5). In contrast to the previous trend, the dem_{ooss} ratio tends to decrease in the iDS372 and iOD907 models. Regarding the maximisation of compound production, *BioISO* also attains higher dem_{ooss} ratios in both iJO1366 and iTO977 models at a depth level of 2 (guided search). Nevertheless, the dem_{ooss} ratio obtained in the iBsu1103 model is smaller in comparison to the value obtained for the shallow search (depth level of 1).

In general, the dem_{wss} ratio tends to increase as a response to *BioISO*'s guided search (depth level of 2) in all models for both objective functions, though not exceeding 0.221. As shown in Fig. 5 and Appendix E.1-2, available online, the dem_{wss} ratio ranges between 0.01 (iOD907 model) and 0.13 (iBsu1103 model) for the growth maximisation analysis. During the compound production maximisation analysis, the dem_{wss} ratio is less than 0.1 in all models except for the iDS372 model, where it peaks 0.221 (Fig. 5 and Appendix E.3-4, available online).

Using *BioISO* for nearly exhaustive searches (depth level of 3) returns br_{ooss} ratios between 0.41 and 0.81, whereas the br_{wss}

ratio varies between 0.35 and 0.66 for both objective functions (Appendix E.1-4, available online). As for detecting dead-end metabolites during the growth maximisation analysis, *BioISO's* nearly exhaustive search attains the highest dem_{ooss} and dem_{wss} ratios of 0.455 and 0.327 in the iDS372 model (Fig. 5), respectively. Regarding the compound production maximisation, *BioISO* peaked for a depth level of 3 dem_{ooss} and dem_{wss} ratios of 0.737 and 0.602 in the iDS372 model (Fig. 5), respectively.

Although the br_{ooss} ratio oscillates when rising depth, the br_{wss} tends to increase steadily. Similarly, the dem_{wss} also tends to increase with depth for both objective functions, whereas the dem_{ooss} ratio mimics the oscillatory behaviour of the br_{ooss} ratio. The oscillatory behaviour of the br_{ooss} and dem_{ooss} ratios is heavily pronounced between depths 1 and 2, which can be associated with the reduced level of detail that *BioISO* can provide for shallow searches.

The high br_{wss} ratios obtained for all levels are likely associated with the fact that *BioISO* does not prevent circular dependencies nor by-products accumulation when testing reactions. The interactive output of *BioISO* in both webserver and *merlin* guides the user through the dead-end metabolites (precursors and successors having an unsuccessful evaluation) while evaluating the reactions for guidance and further insight.

When testing metabolites, *BioISO's* strategy to prevent circular dependencies and by-product accumulation, as well as the isolated evaluation of precursors and successors, seems to have a greater impact on reducing the set of dead-end metabolites. The dem_{wss} ratio is significantly smaller for shallow and guided searchers across all models for both objective functions.

Furthermore, although *BioISO* has attained dem_{wss} ratios higher than 0.4 for two models during the compound production maximisation analysis with a depth level of 3, this ratio remains below 0.33 in all models during the growth maximisation analysis. The dem_{wss} ratios higher than 0.4 obtained with nearly exhaustive searches of *BioISO* may be associated with the factual metabolic gaps that do not need to be corrected or might not be associated with the desired phenotype.

For example, *BioISO* has systematically attained higher ratios for all metrics when assessing the iDS372 incomplete models for both objective functions. These higher scores may be associated with poor connectivity of most metabolites involved in the metabolic pathways analysed by *BioISO*, as parasitic organisms evolve in rich media, thus developing auxotrophies [16], [26], [27]. Hence, it is worth noticing that the identified dead-end metabolites might be associated with real metabolic gaps that should not be gap-filled.

In short, *BioISO* scores most of the smaller dem_{wss} ratios at the depth level of 2 (guided search). Moreover, the gap between dem_{wss} and dem_{ooss} ratios also starts to narrow for *BioISO*'s guided search. The small difference between both metrics suggests that most dead-end metabolites suggested by *BioISO* are a direct outcome of the actual network gaps introduced during the validation. Hence, *BioISO* can suggest a higher number of dead-end metabolites associated with the objective while maintaining the curation efforts at a minimum.

Therefore, a depth level of 2 was selected as the default level for running *BioISO* after analysing all dem_{wss}. Using this depth value, *BioISO* can guide the search to identify errors in a given metabolic network, without evaluating only the direct precursors and successors, such as *biomassPrecursorsCheck* [10] and Meneco [6], or the burden of evaluating the whole network, such as fastGapFill [8] and gapFind/gapFill [7].

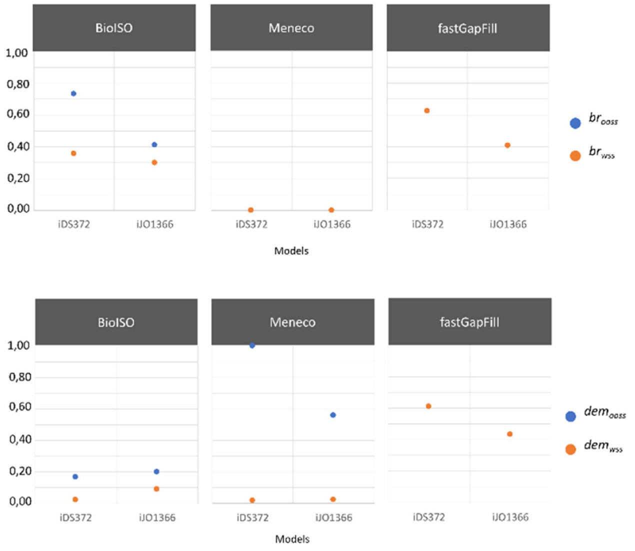

Fig. 6. Assessment of the relevance of guided (*BioISO* and *Meneco*) versus exhaustive searches (*fastGapFill*) for gap-finding. *BioISO*, *Meneco*, and *fastGapFill* were used to highlight gaps in two state-of-the-art models (iDS372 and iJO1366), with added gaps. The ratios of dead-end metabolites (dem_{ooss} and dem_{wss}) and blocked reactions (br_{ooss} and br_{wss}) for both objective-oriented search (*ooss*) and whole search (*wss*) spaces were then calculated for each tool.

Furthermore, a significant part of the metabolic network associated with a given objective is analysed by the tool for a guided search (depth level of 2), while the required computation time is significantly lower.

B. Exhaustive-Search versus Guided-Search

The exhaustive-search versus guided-search assessment was designed to compare the results of two guided-search tools, namely *BioISO* (guided search – depth level of 2) and *Meneco* [6], with *fastGapFill* [8] exhaustive-search application. The iJO1366 and iDS372 models obtained for the growth maximisation analysis were used in this assessment. The workflow and methodology used to compare exhaustive-searches against guided-searches are described in further detail in the Materials and Methods section, together with Appendix D, available online and E. Fig. 6 exhibits a summary of dead-end metabolites and blocked reactions ratios calculated for each tool.

According to Fig. 6 and Appendix E.5-9, available online, *Meneco* performed the poorest in identifying metabolic gaps. Besides the poor performance regarding the assessment of dead-end metabolites, it does not provide insights on blocked reactions.

The number of covered metabolites when using *Meneco* is the same as the number of metabolites selected for target metabolites, as this tool only evaluates target metabolites. No information is provided about other metabolites in the metabolic network.

Hence, although *Meneco* obtained the lowest dem_{wss} ratios in both models (Fig. 6), the tool suggests the absence of biosynthetic pathways to synthesise all metabolites in the covered search space, thus obtaining the highest dem_{ooss} ratios in both models (Fig. 6). In short, most metabolites analysed by this tool were evaluated as dead-end metabolites in the incomplete models.

FastGapFill provides, on the other hand, a level of insight much larger than *Meneco*, analysing all reactions and metabolites in all models (Fig. 6 and Appendix E.5-9, available online). However, the exhaustive-search tool is associated with a significant drawback. *fastGapFill* highlights numerous blocked reactions and dead-end metabolites according to Fig. 6, which might hinder a fast and precise identification of a *de facto* error in the network, such as the ones introduced in this validation procedure.

For example, *fastGapFill* has attained higher dem_{wss} ratios (Fig. 6) for the model of the less described and smaller genome's organism (*Streptococcus pneumoniae*), which has probably evolved through a combination of extensive loss-of-function events during the co-evolution with well-defined and constant ecological niches [16], [26], [27].

Although *Meneco* has obtained smaller dem_{wss} scores than *BioISO*, the level of insight provided by the gap-filling tool for both metabolic networks are significantly lower than the detail provided by our tool. When comparing the dem_{ooss} ratios, it is clear the lack of insight provided by Meneco, as most of the metabolites covered by *Meneco* are highlighted as dead-end metabolites. On the other hand, *BioISO* can be more effective and precise by suggesting fewer dead-ends out of the examined metabolites pool (Fig. 6 and Appendix E.5-9, available online).

According to Fig. 6, *BioISO* attained lower br_{wss} and dem_{wss} ratios than fastGapFill in both models. Hence, such smaller br_{wss} and dem_{wss} ratios suggest that *BioISO* is more capable of reducing the whole-space search to fewer dead-end metabolites than the gap-filling tool.

When debugging and validating the model for specific objective functions, such as growth maximisation, *BioISO* seems better suited for reducing the search space for errors and gaps in metabolic networks than the other tools analysed in this assessment. This advantage allows spending less time debugging unrealistic errors or gaps. Furthermore, as *BioISO* reduces the search space for errors, it also favours parsimonious alterations to the draft metabolic network. As a result, *BioISO* can be of paramount importance for the high-quality bottom-up reconstruction of GEMs during the manual curation stage.

However, it should be noticed that *Meneco* and *fastGapFill* have been designed to be essentially gap-fill tools. Thus, these tools use different approaches than *BioISO* to find errors.

C. BioMeneco – Embedding BioISO in Meneco

BioMeneco, *BioISO's* integration with Meneco [6], was developed to determine whether the former can improve the latter results by narrowing the search space for the gap-filling task. For that, reactions "R04568_C3_cytop" and "SO4tex" were removed from the iDS372 and iJO1366 models, respectively. Meneco was then used to generate potential solutions for restoring models' prediction of a growth phenotype based on *BioISO* suggestions for the set of targets (primary input for Meneco).

TABLE I
DEAD-END METABOLITES IDENTIFIED BY *BIOISO* IN THE IDS372 MODEL

Dead-End metabolite	Connected dead-end metabolites
C01356_cytop (Lipid)	C05980_cytop C06040_cytop C04046_cytop C00344_cytop
C06042_cytop (Lipoteichoic acid)	C00116_cytop C00162_cytop

Metabolites identified by BioISO as not being produced or consumed (dead-end metabolites) by the iDS372 model missing R04568_C3_cytop reaction.

All metabolites identified as not being produced or consumed by *BioISO* in the iDS372 model are reported in Table I. These metabolites have been selected for the set of target metabolites after a brief analysis of the *BioISO's* output.

In the iDS372 model, *BioISO* indicated that reaction "R04568_C3_cytop" might be associated with the synthesis of a precursor of the lipidic and lipoteichoic acid pathways. Most dead-end metabolites identified by *BioISO* were associated with metabolites "C01356_cytop" and "C06042_cytop", which are biomass precursors representing the lipid and lipoteichoic acid cellular biomass fractions, respectively. These suggestions are in agreement with the metabolites being synthesised by the reaction removed from the model. Reaction "R04568_C3_cytop" is associated with the synthesis of "trans-Tetradec-2-enoyl-[acp]" ("C05760_cytop"), which in turn is a precursor of the compound "Tetradecanoyl-[acp]" ("C05761_cytop") involved in the fatty acid biosynthesis pathway.

Besides the dead-end metabolites shown in Table I, *BioISO* suggested an unsuccessful evaluation of all remaining biomass precursors and successors. Nevertheless, only the lipid and lipoteichoic acid compounds were identified as dead-end metabolites. The remaining biomass precursors and successors refer to the special cases reported in the tutorial at https://bioiso.bio.di.uminho.pt/tutorial. Briefly, these metabolites were unsuccessfully evaluated due to a missing or impaired reaction downstream, namely the biomass reaction.

The corresponding precursors and successors of all neighbour metabolites were classified as non-dead-end metabolites, except for several precursors and successors of the lipid and lipoteichoic acid compounds.

All metabolites identified as not being produced or consumed by *BioISO* in the incomplete iJO1366 model, reported in Table II, were selected for the set of target metabolites.

TABLE II
DEAD-END METABOLITES IDENTIFIED BY *BIOISO* IN THE IJO1366 MODEL

Dead-End metabolite	Connected dead-end metabolites
2fe2s_c ([2Fe-2S] iron-sulphur cluster)	4fe4s_c lipopb_c iscu_DASH_2fe2s_c iscu_c sufbcd_DASH_2fe2s_c sufbcd_c
4fe4s_c ([4Fe-4S] iron-sulphur cluster)	iscu_DASH_4fe4s_c sufbcd_DASH_4fe4s_c 3fe4s_c
bmocogdp_c (bis-molybdopterin guanine dinucleotide)	bmoco1gdp_c
btn_c (Biotin)	btn_p btnso_c 2fe1s_c
so4_c (Sulphate)	so4_p

Metabolites identified by BioISO as not being produced or consumed (dead-end metabolites) by the iJO1366 model missing SO4tex reaction.

BioISO has indicated that the missing "SO4tex" reaction might be associated with synthesising a precursor for sulphur metabolism. Most dead-end metabolites identified by *BioISO* were linked to the iron-sulphur clusters, biotin, bis-molybdopterin guanine dinucleotide, and sulphate biomass precursors, which are all associated with the sulphur requirements of *E. coli*. These results are in line with the transport of sulphate to the periplasm by the removed reaction. The "SO4tex" reaction is responsible for transporting sulphate from the extracellular medium to the periplasm, which is then transported to the cytoplasm.

BioISO suggested more dead-end metabolites than the precursors described in Table II, absent from the set of "target" metabolites for the iJO1366 model. *BioISO* negatively evaluated the biomass precursors "mobd_c", "sheme_c", "cl_c", and "2ohph_c". Nevertheless, these metabolites have been ignored as dead-end metabolites, as they refer to the special cases reported in the tutorial at https://bioiso.bio.di.uminho.pt/tutorial. The precursors of these metabolites are being produced, and the successors are consumed, except for the biomass. Thus, *BioISO* highlighted these metabolites because the biomass reaction is, actually, the only consumption site available. These metabolites are an example of unsuccessful evaluations that should be easily detected in the user-friendly output returned by the web server and *merlin*. Moreover, the automatic tools would deal with such cases as perhaps regular gaps and incorrectly resolve them.

The gap-filling solutions suggested by Meneco for the incomplete iDS372 model are satisfactory. The proposed solutions could restore flux through the biomass reaction, and thus through all set of targets initially proposed. Meneco suggests adding reaction "R04568" (which was previously removed for this test) to restore the metabolic model.

Nevertheless, other solutions may add artefacts in the iDS372 model. Reactions "R11633", "R09085", "R11636", "R11634", "R11671" and "R00183" are equally recommended to restore flux throughout the set of targets. However, these reactions are not involved in the synthesis or consumption of missing biomass precursors or successors. Most reactions are involved in the synthesis of biomass precursors not affected by the removed reaction, such as the "R11636" ("dCTP" synthesis), "R09085" (carbon metabolism), "R11634" ("dATP "synthesis), and "R11633" ("dGTP" synthesis). Other reactions are involved in the synthesis of metabolites not required for *S. pneumoniae's* growth. Note that only reactions suggested in the pool named "One minimal completion" were considered. Nevertheless, Meneco provides a complete enumeration of all combinations of minimal completions.

BioMeneco recommended, on the other hand, a reduced pool of gap-filling solutions. In this case, *BioMeneco* suggested reaction "R04568", but now only three reactions ("R11671", "R00182", and "R09085") have been equally proposed. As neither RNA nor DNA were included in the set of target metabolites, all reactions previously suggested to restore the synthesis of purines and pyrimidines have been discarded.

Meneco restored the test iJO1366 model for six biomass precursors while indicating 35 "unreconstructable targets". Meneco identified the "so4_c" metabolite, which is one of the biomass precursors affected by the removal of the "SO4tex" reaction, as "reconstructable". Nevertheless, the remaining metabolites for which Meneco could restore flux were not affected by the removed reaction.

More importantly, Meneco's output does not comprise the "SO4tex" reaction in the set of gap-filling solutions to restore flux through all biomass precursors. More surprisingly, the "SO4tex" reaction was not included in any combination of minimal completions obtained through the complete enumeration of solutions. Furthermore, other potential solutions can lead to the introduction of artefacts in the iJO1366 model. For example, all reactions included in the "One minimal completion" set of solutions are transport reactions for biomass precursors not affected when reducing the iJO1366 model.

Interestingly, only the reaction "SO4tex" has been suggested by *BioMeneco* to restore flux through all missing biomass precursors and successors in the test iJO1366 model. Moreover, as none of the other biomass precursors was included in the set of target metabolites, all reactions involved in the transport of co-factors, ions, and amino acids were discarded from the "One minimal completion" pool.

Therefore, *BioISO* can be used to decrease large search spaces associated with model debugging procedures. Besides proposing a user-friendly application to guide the search for dead-end metabolites, we have showcased that *BioISO* can also facilitate high-quality bottom-up reconstructions by adjusting the guided-search gap-filling tool Meneco. For that, we suggest BioMeneco as an iterative process comprising two separate tasks:

1) running *BioISO* to identify the set of metabolites not being produced or consumed (dead-end metabolites).
2) running Meneco using the set of metabolites highlighted earlier as target metabolites, to obtain parsimonious solutions to complete draft metabolic networks.

IV Conclusion

BioISO is a user-friendly tool capable of performing guided searches of gaps in metabolic networks. This tool aims to help reconstruct high-quality genome-scale metabolic models by scientists without coding skills, leveraging bottom-up reconstructions that require manual curation and human intervention.

Several state-of-the-art gap-find and gap-fill tools have been compared with *BioISO*, which emerged as the only open-source tool ready to be used by everyone in the scientific community. Moreover, *BioISO* is not associated with the significant drawbacks of using a gap-filling method, such as poor usability, the requirement for additional data, and recommending biological artefacts due to the lack of evidence for the solutions.

BioISO has been validated with GEMs available in the literature [23], [24], [26], [28], [29]. *BioISO's* validation comprehended three assessments: algorithm depth analysis – assessment of shallow, guided or nearly exhaustive searches with *BioISO*; exhaustive-search (fastGapFill [8]) versus guided-search (*BioISO* and Meneco [6]); embedding *BioISO* in Meneco [6] - the outcome of setting *BioISO* as Meneco's gap-finding algorithm.

The ratio of dead-end metabolites obtained by *BioISO* for objective-oriented and whole-search spaces increases with the depth of search to all tested metabolic networks. Nevertheless, *BioISO* attains lower ratios for shallow (depth level of 1) and guided searches (depth level of 2). This suggests that *BioISO* can highlight a reduced number of dead-end metabolites for a significant part of the metabolic network associated with a given objective. Thus, *BioISO*'s guided-search, with a depth level of 2, should yield the best trade-off between analysing a significant portion of the whole metabolic network, while still highlighting the *de facto* dead-end metabolites for a given objective.

Although *BioISO* has not been tested in GEMs of high complexity organisms (e.g., human), the tool can still be used to evaluate large metabolic networks. Since *BioISO* performs a guided-search oriented to the objective phenotype, it might be required to run the tool several times towards the identification of dead-end metabolites far from the reaction being evaluated.

In addition, *BioISO* does not consider reaction reversibility to assert precursors (reactants) and successors (products). Hence, one should evaluate a model version with the directionality of interest for the reactions being evaluated. Alternatively, one can also perform *BioISO* analysis using a version of the GEM containing only irreversible reactions by duplicating reversible reactions into the two directions.

Although *BioISO* has attained lower dead-end metabolites ratios in the whole-space search than fastGapFill, Meneco has scored even smaller ratios. Nevertheless, the level of detail provided by Meneco is significantly lower, as most of the metabolites covered by Meneco are highlighted as dead-end metabolites. On the other hand, *BioISO* can be more effective and precise by suggesting fewer dead-ends from the examined metabolites pool.

When debugging and validating the model for specific objective functions, such as growth maximisation, *BioISO* seems better suited for reducing the search space for errors and gaps in metabolic networks than the other tools analysed in this assessment.

When using *BioISO* to pre-process Meneco's set of targets, the latter suggested the correct minimal completions. *BioISO* can improve Meneco's gap-finding algorithm, facilitating Meneco's integration with a high-quality bottom-up reconstruction workflow by following an iterative process comprising two separate tasks: running *BioISO* to identify dead-end metabolites; running Meneco using the set of metabolites highlighted as target metabolites, to obtain parsimonious solutions.

V Materials and Methods

A. *BioISO's Algorithm Depth Analysis*

BioISO's algorithm depth analysis was performed in parallel for both objective functions, namely growth and compound production maximisation. This assessment allowed setting shallow, guided, or nearly exhaustive searches with *BioISO* to assess the algorithm's robustness. *BioISO*'s algorithm depth analysis was performed in five state-of-the-art GEMs: iDS372 (*Streptococcus pneumoniae*) [26]; iJO1366 (*Escherichia coli*) [23]; iBsu1103 (*Bacillus subtilis*) [24]; iTO977 (*Saccharomyces cerevisiae*) [29]; iOD907 (*Kluyveromyces lactis*) [28].

For instance, five incomplete models were created for the growth maximisation analysis by removing the following reactions from the *E. coli* iJO1366 model [23], one at a time: SDPTA; IMPC; MEPCT; NNDPR; SERAT. The incomplete models were evaluated by setting *BioISO*'s algorithm depth level at 1, and growth maximisation as the objective function (Ec_biomass_iJO1366_core_53p95M). This procedure was repeated for the remaining models, depth levels of 2 and 3, and compound production maximisation analysis.

Then, two metrics were proposed to evaluate the gap-finding performance, namely the ratio of dead-end metabolites and the ratio of blocked reactions. These metrics were used to quantify the search space associated with model debugging of gaps and errors, which in a worst-case scenario includes all metabolites and reactions of GEM. As shown in (2), the ratio of dead-end metabolites for the objective-oriented search space (dem_{ooss}) is a function of the number of metabolites that a guided-search tool evaluates as unsuccessful (dead-end metabolite) divided by the size of the objective-oriented search space (*ooss*), also referred as the number of covered/evaluated metabolites and reactions. The ratio of dead-end metabolites for the whole search space (dem_{wss}) is a function of the number of found dead-end metabolites divided by the *wss*, as described in (3). Equations (4) and [(5)] describe a similar approach to calculate the ratio of blocked reactions for the objective-oriented search (br_{ooss}) and the whole search (br_{wss}) spaces, respectively.

$$dem_{ooss} = \frac{\sum DeadEndMetabolites}{\sum CoveredMetabolies} \tag{2}$$

$$dem_{wss} = \frac{\sum DeadEndMetabolites}{\sum Metabolies} \tag{3}$$

$$br_{ooss} = \frac{\sum BlockedReactions}{\sum CoveredReactions} \tag{4}$$

$$br_{wss} = \frac{\sum BlockedReactions}{\sum Reactions} \tag{5}$$

Objective functions, settings, evaluation metrics, and methodologies used to introduce gaps in state-of-the-art GEMs used during the algorithm depth analysis can also be consulted in detail at Appendix D and E, available online.

B. Exhaustive-Search versus Guided-Search

The exhaustive-search versus guided-search analysis was performed for both iDS372 [26] and iJO1366 [23] models to assess the relevance of guided (*BioISO* and *Meneco* [6]) and exhaustive searches (*fastGapFill* [8]) for gap-finding.

BioISO was used as described in the previous section for a depth level of 2.

All metabolites available in the extracellular compartment of the iDS372 or iJO1366 models were used as seed metabolites to run *Meneco* gap-finding methodology. Likewise, the set of target metabolites was comprised of precursors and successors of the evaluated reactions. The set of dead-end metabolites was determined through *Meneco's* '*get_unproducible*' method. Note that *Meneco* cannot assert blocked reactions. Thus, the set of blocked reactions could not be determined.

FastGapFill was used to assess the whole search space by accounting for errors and gaps. *fastGapFills' 'gapFind'* and *'findBlockedReaction'* methods were used to determine dead-end metabolites and blocked reactions, respectively, in the incomplete models.

The metrics described in the previous section were then used to assess the tools' performance, namely the ratio of dead-end metabolites and the ratio of blocked reactions. Appendix D and E, available online, present all details about the assessment of *BioISO* with *Meneco* and *fastGapFill*.

C. BioMeneco – Embedding BioISO in Meneco

The BioMeneco analysis was performed for the iDS372 [26] and iJO1366 [23] models to assess the integration of *BioISO* as *Meneco's* gap-finding algorithm.

Meneco's topological search finds dead-end metabolites, so the gaps associated with them can be filled with reactions from a universal database. The novelty of *Meneco* is that it allows selecting which gaps should be filled by tweaking the set of target metabolites. Hence, *BioMeneco*, *BioISO's* integration with *Meneco*, was performed to assess whether *BioISO* can suggest the right set of targets to be used as input in *Meneco*.

Reactions "R04568_C3_cytop" and "SO4tex" were removed from the iDS372 and iJO1366 models, respectively, to perform this assessment for growth validation. Meneco was then used to generate potential solutions for both models using two sets of "target" metabolites in parallel:

1) The set of "target" metabolites comprised precursors and successors of the evaluated reaction in each model.
2) The set of "target" metabolites was formulated based on the identification of dead-end metabolites by *BioISO*.

BiGG [14] universal database was used as the source of metabolic reactions for the test iJO1366 model, while KEGG [13] was used to solve gaps in the test iDS372 model.

References

[1] I. Thiele and B. Ø. Palsson, "A protocol for generating a high-quality genome-scale metabolic reconstruction.," *Nature Protoc.*, vol. 5, no. 1, pp. 93–121, 2010, doi: 10.1038/nprot.2009.203.

[2] C. S. Henry, M. DeJongh, A. A. Best, P. M. Frybarger, B. Linsay, and R. L. Stevens, "High-throughput generation, optimization and analysis of genome-scale metabolic models," *Nat. Biotechnol.*, vol. 28, no. 9, pp. 977–982, Sep. 2010, doi: 10.1038/nbt.1672.

[3] O. Dias and I. Rocha, "Systems biology in fungi," in *Molecular Biology of Food and Water Borne Mycotoxigenic and Mycotic Fungi*, R. Paterson, Ed. Boca Raton, FL, USA: CRC Press, 2015, pp. 69–92.

[4] J. D. Orth, I. Thiele, and B. O. Palsson, "What is flux balance analysis?," *Nat Biotechnol.*, vol. 28, no. 3, pp. 245–248, 2010, doi: 10.1038/nbt.1614.

[5] 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 Res.*, vol. 46, no. 15, pp. 7542–7553, Sep. 2018, doi: 10.1093/nar/gky537.

[6] S. Prigent et al., "Meneco, a topology-based gap-filling tool applicable to degraded genome-wide metabolic networks," *PLOS Comput. Biol.*, vol. 13, no. 1, Jan. 2017, Art. no. e1005276, doi: 10.1371/journal.pcbi.1005276.

[7] V. Satish Kumar, M. S. Dasika, and C. D. Maranas, "Optimization based automated curation of metabolic reconstructions," *BMC Bioinf.*, vol. 8, no. 1, Jun. 2007, Art. no. 212, doi: 10.1186/1471-2105-8-212.

[8] I. Thiele, N. Vlassis, and R. M. T. Fleming, "FASTGAPFILL: Efficient gap filling in metabolic networks," *Bioinformatics*, vol. 30, no. 17, pp. 2529–2531, Sep. 2014, doi: 10.1093/bioinformatics/btu321.

[9] Z. Hosseini and S. A. Marashi, "Discovering missing reactions of metabolic networks by using gene co-expression data," *Sci. Rep.*, vol. 7, no. 1, pp. 1–12, Feb. 2017, doi: 10.1038/srep41774.

[10] L. Heirendt et al., "Creation and analysis of biochemical constraint-based models using the COBRA Toolbox v.3.0," *Nature Protoc.*, vol. 14, no. 3, pp. 639–702, Mar. 2019, doi: 10.1038/s41596-018-0098-2.

[11] J. L. Reed et al., "Systems approach to refining genome annotation," *Proc. Nat. Acad. Sci. USA*, vol. 103, no. 46, pp. 17480–17484, Nov. 2006, doi: 10.1073/pnas.0603364103.

[12] E. Vitkin and T. Shlomi, "MIRAGE: A functional genomics-based approach for metabolic network model reconstruction and its application to cyanobacteria networks," *Genome Biol.*, vol. 13, no. 11, Nov. 2012, Art. no. R111, doi: 10.1186/gb-2012-13-11-r111.

[13] M. Kanehisa, M. Furumichi, M. Tanabe, Y. Sato, and K. Morishima, "KEGG: New perspectives on genomes, pathways, diseases, and drugs," *Nucleic Acids Res.*, vol. 45, no. D1, pp. D353–D361, Jan. 2017, doi: 10.1093/nar/gkw1092.

[14] Z. A. King et al., "BiGG Models: A platform for integrating, standardizing and sharing genome-scale models," *Nucleic Acids Res.*, vol. 44, no. D1, pp. D515–D522, Jan. 2016, doi: 10.1093/nar/gkv1049.

[15] R. Caspi et al., "The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome databases," *Nucleic Acids Res.*, vol. 42, no. D1, pp. D459–D471, Jan. 2014, doi: 10.1093/nar/gkt1103.

[16] K. S. Makarova et al., "Comparative genomics of the lactic acid bacteria.," *Proc. Nat. Acad. Sci. USA*, vol. 103, no. 42, pp. 15611–15616, 2006, doi: 10.1073/pnas.0607117103.

[17] A. Bolotin et al., "Complete sequence and comparative genome analysis of the dairy bacterium Streptococcus thermophilus.," *Nat. Biotechnol.*, vol. 22, no. 12, pp. 1554–1558, 2004, doi: 10.1038/nbt1034.

[18] M. van de Guchte et al., “The complete genome sequence of Lactobacillus bulgaricus reveals extensive and ongoing reductive evolution.,” *Proc. Nat. Acad. Sci. USA*, vol. 103, no. 24, pp. 9274–9279, 2006, doi: 10.1073/pnas.0603024103.

[19] K. A. Pagel et al., “When loss-of-function is loss of function: Assessing mutational signatures and impact of loss-of-function genetic variants,” *Bioinformatics*, vol. 33, no. 14, pp. i389–i398, Jul. 2017, doi: 10.1093/bioinformatics/btx272.

[20] A. Ebrahim, J. A. Lerman, B. O. Palsson, and D. R. Hyduke, “COBRApy: Constraints-based reconstruction and analysis for python,” *BMC Syst. Biol.*, vol. 7, no. 1, Aug. 2013, Art. no. 74, doi: 10.1186/1752-0509-7-74.

[21] M. Hucka et al., “The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models,” *Bioinformatics*, vol. 19, no. 4, pp. 524–531, Mar. 2003.

[22] O. Dias, M. Rocha, E. C. Ferreira, and I. Rocha, “Reconstructing genome-scale metabolic models with merlin,” *Nucleic Acids Res.*, vol. 43, no. 8, pp. 3899–3910, Apr. 2015, doi: 10.1093/nar/gkv294.

[23] J. D. Orth et al., “A comprehensive genome-scale reconstruction of Escherichia coli metabolism—2011,” *Mol. Syst. Biol.*, vol. 7, no. 1, Jan. 2011, Art. no. 535, doi: 10.1038/msb.2011.65.

[24] C. S. Henry, J. F. Zinner, M. P. Cohoon, and R. L. Stevens, “iBsu1103: A new genome-scale metabolic model of Bacillus subtilis based on SEED annotations,” *Genome Biol.*, vol. 10, no. 6, Jun. 2009, Art. no. R69, doi: 10.1186/gb-2009-10-6-r69.

[25] J. D. Orth et al., “A comprehensive genome-scale reconstruction of Escherichia coli metabolism–2011,” *Mol. Syst. Biol.*, vol. 7, Oct. 2011, Art. no. 535, doi: 10.1038/msb.2011.65.

[26] O. Dias, J. Saraiva, C. Faria, M. Ramirez, F. Pinto, and I. Rocha, “iDS372, a phenotypically reconciled model for the metabolism of streptococcus pneumoniae strain R6,” *Front. Microbiol.*, vol. 10, Jun. 2019, Art. no. 1283, doi: 10.3389/fmicb.2019.01283.

[27] J. Hoskins et al., “Genome of the bacterium Streptococcus pneumoniae strain R6,” *J. Bacteriol.*, vol. 183, no. 19, pp. 5709–5717, Oct. 2001, doi: 10.1128/JB.183.19.5709-5717.2001.

[28] O. Dias, R. Pereira, A. K. Gombert, E. C. Ferreira, and I. Rocha, “iOD907, the first genome-scale metabolic model for the milk yeast Kluyveromyces lactis,” *Biotechnol. J.*, vol. 9, no. 6, pp. 776–790, Jun. 2014, doi: 10.1002/biot.201300242.

[29] T. Österlund, I. Nookaew, S. Bordel, and J. Nielsen, “Mapping condition-dependent regulation of metabolism in yeast through genome-scale modeling,” *BMC Syst. Biol.*, vol. 7, no. 1, Apr. 2013, Art. no. 36, doi: 10.1186/1752-0509-7-36.

Fernando Cruz received the PhD degree in biomedical engineering from the University of Minho. He is currently working with OmniumAI.

João Capela received the master’s degree in bioinformatics. He is currently working toward the PhD degree with the University of Minho.

Eugénio C. Ferreira received the PhD degree in chemical engineering from the University of Minho. He is the vice-rector with the University of Minho. He is also a full professor with the Department of Biological Engineering and director with the PhD program on bioengineering with the University of Minho.

Miguel Rocha received the master’s degree in bioinformatics from the University of Minho, and the PhD degree in computer science from the University of Minho. He is an Associate Professor with the University of Minho.

Oscar Dias received the master’s degree in bioinformatics from the University of Minho, and the PhD degree in chemical and biological engineering from the University of Minho. He is an assistant researcher with the University of Minho. He is also a member of the Coordinating Committee.