# XomicsToModel: Omics data integration and generation of thermodynamically consistent metabolic models

15th December 2022

German Preciat[1,2*], Agnieszka B. Wegrzyn[1*], Ines Thiele[3,4,5,6], Thomas Hankemeier[1†] and Ronan M.T. Fleming[1,3†]

**Abstract**

Constraint-based modelling can mechanistically simulate the behaviour of a biochemical system, permitting hypotheses generation, experimental design and interpretation of experimental data, with numerous applications, including modelling of metabolism. Given a generic model, several methods have been developed to extract a context-specific, genome-scale metabolic model by incorporating information used to identify metabolic processes and gene activities in a given context. However, existing model extraction algorithms are unable to ensure that the context-specific model is thermodynamically feasible. This protocol introduces XomicsToModel, a semi-automated pipeline that integrates bibliomic, transcriptomic, proteomic, and metabolomic data with a generic genome-scale metabolic reconstruction, or model, to extract a context-specific, genome-scale metabolic model that is stoichiometrically, thermodynamically and flux consistent. The XomicsToModel pipeline is exemplified for extraction of a specific metabolic model from a generic metabolic model, but it enables omics data integration and extraction of physicochemically consistent mechanistic models from any generic biochemical network. With all input data fully prepared, algorithmic completion of the pipeline takes ~10 min, however manual review of intermediate results may also be required, e.g., when inconsistent input data lead to an infeasible model.

**Keywords**

Constraint-based modelling, omics, data integration, thermodynamic consistency, systems biology.

[1]Metabolomics and Analytics Centre, Leiden Academic Centre for Drug Research, Leiden University, Einsteinweg, Leiden, The Netherlands,

[2]Centro Universitario de Ciencias Exactas e Ingenierías, University of Guadalajara, Guadalajara, Mexico,
[3]School of Medicine, National University of Ireland, University Road, Galway, Ireland,
[4]Ryan Institute, National University of Galway, Galway, Ireland,
[5]Division of Microbiology, National University of Galway, Galway, Ireland,
[6]APC Microbiome Ireland, Ireland.

[*]These authors contributed equally to this work.

[†]Correspondence should be addressed to Thomas Hankemeier (hankemeier@lacdr.leidenuniv.nl) and Ronan M.T. Fleming (ronan.mt.fleming@gmail.com).

# INTRODUCTION

## Introduction to genome-scale metabolic model extraction

The main goal of a genome-scale metabolic model is to represent all known metabolic functions and predict physiochemically and biochemically realistic metabolic fluxes in living systems [23]. One can distinguish four complementary approaches to the development of a genome-scale metabolic model for a single organism or anatomical region of interest, e.g., organ, tissue or cell type. In the first instance, if no network reconstruction is available, a high-quality genome-scale metabolic reconstruction of a single organism can be generated from scratch by following established protocols [27]. Subsequently, various metabolic models can be derived from a single reconstruction, by computational application of different combinations of mathematical modelling assumptions. In the second instance, if a network reconstruction, or a model, is available for an organism with orthologous genes, methods to generate metabolic models for related microbial [20] and mammalian [33] species have been proposed. In the third instance, given a generic metabolic reconstruction, or model, for a multi-cellular organism, containing metabolic reactions from any anatomical region, there are several methods for derivation of a model that is specific for a particular anatomical region [21]. The latter are also referred to as methods for tailoring a genome-scale metabolic model . When objectively compared in detail, it was found that the model derivation method most strongly affected the accuracy of gene-essentiality predictions [21]. In the fourth instance, given a universal metabolic reaction database, containing metabolic reactions from multiple organisms, an organism-specific model can be derived using one of the several existing methods , e.g., CarveMe [17]. In all except the first instance, one is given a generic reconstruction, or model, and the method extracts a specific model, which is a subset of the reconstruction.

A specific model may be specific to a particular organism or a particular anatomical region. For example, given a generic human metabolic model, one could extract a hepatocyte model [16], with the capability to model different contexts, depending on the constraints subsequently applied to it. A specific model may also be context-specific, which is a particular organism or anatomical region in a particular environmental and internal context. From the same starting point, with different input data, one could extract a context-specific model, e.g., a hepatocyte model specific to a fasting state [31]. The distinction between cell-type and context-specific is more a gradation of specificity, paralleled by increased constraints and decreased volume of the feasible steady state solution set, which can be quantified for convex sets [13].

Established model extraction methods ensure that a specific model is flux-consistent, that is, each reaction admits a non-zero steady-state flux. Consider Flux Balance Analysis [22], which requires the solution to the following optimisation problem

$$\begin{aligned} \max_{v \in \mathbb{R}^r} \quad & \rho(v) := c^T v \\ \text{s.t.} \quad & Sv = 0, \\ & l \leq v \leq u, \end{aligned} \tag{1}$$

where $v \in \mathbb{R}^r$ is the net rate of each reaction, $\rho(v) := c^T v$ is a biologically motivated linear objective, specified by the coefficient vector $c \in \mathbb{R}^r$. The matrix $S \in \mathbb{R}^{m \times r}$ is a stoichiometric matrix, where $m$ is the number of metabolites, and $r$ is the number of internal and exchange reactions. The constraint $Sv = 0$ represents the assumption of steady-state, that is 1) production = consumption for metabolites not exchanged across the boundary of the system and 2) production + uptake = consumption + secretion for metabolites exchanged across the boundary of the system. The inequalities $l \leq v \leq u$ denote box constraints from the lower ($l \in \mathbb{R}^r$) and upper bounds ($u \in \mathbb{R}^r$) on reaction rates.

The $j^{th}$ reaction in a network is said to be flux inconsistent if, $v_j = 0$ for all steady state flux vectors in the feasible set $\Omega := \{Sv = 0,\ l \leq v \leq u\}$. Since Flux Balance Analysis only predicts steady state flux, it is misleading to include unidentified flux inconsistent reactions in a model for Flux Balance Analysis because one must be able to distinguish between a reaction that cannot carry steady state

flux (independent of the objective chosen) from a reaction that does not carry steady state flux in a particular Flux Balance Analysis solution (dependent of the objective chosen). Therefore, flux inconsistent reactions should be identified in a model before interpreting the results of methods such as Flux Balance Analysis. Note that, in a reconstruction, a flux inconsistent reaction may be supported by experimental evidence, and serves as a starting point for further refinement, e.g., by gap filling [28].

It is well established that Flux Balance Analysis requires additional constraints to ensure that a steady state flux vector is also thermodynamically feasible [3, 15], i.e., consistent with energy conservation and the second law of thermodynamics [9]. The critical necessity at genome-scale, is to incorporate any additional constraints, or terms in the objective, in a mathematical form that, when computationally implemented, retains the computational efficiency and certificates available with linear optimisation. As exemplified in Figure 1, a stoichiometric matrix may be split into two sets of columns $S := [N, B]$, where the matrix $N \in \mathbb{R}^{m \times n}$ represents internal reactions, which do conserve mass, and the matrix $B \in \mathbb{R}^{m \times k}$ represents external reactions, which do not conserve mass, e.g., reaction equation $A \to \varnothing$. The latter are modelling constructs to represent the exchange of metabolites with the environment. By splitting internal net fluxes into unidirectional fluxes, and maximising the entropy of unidirectional fluxes, one can compute steady state, thermodynamically feasible fluxes in genome-scale biochemical models using convex optimisation [10], which retains the efficiency as well as feasibility certificates available with linear optimisation [6]. However, this approach assumes the model admits a thermodynamically feasible flux. We define a model to be *thermodynamically consistent* if each of its reactions admits a *nonzero* thermodynamically feasible flux. A flux vector may be thermodynamically feasible but the flux is zero for some reaction, which is not enough to establish that reaction to be thermodynamically consistent.

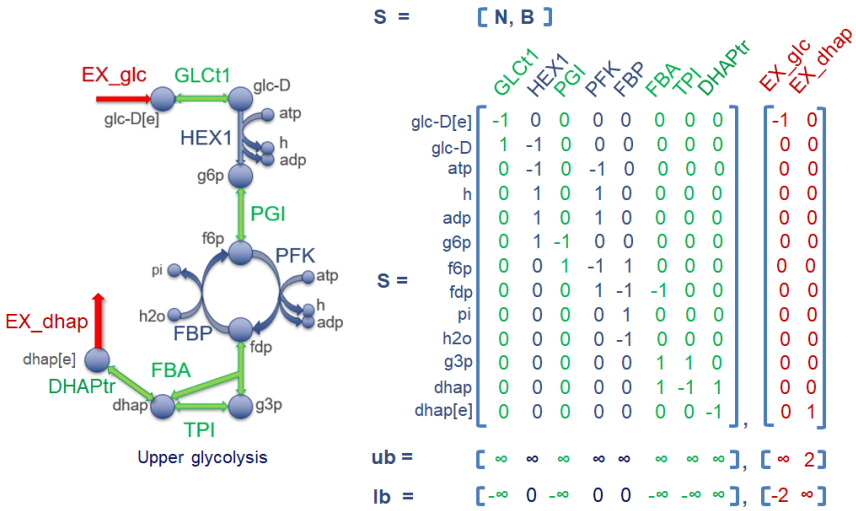



**Figure 1:** Network diagram of upper glycolysis (left) with reactions (arrows, upper case labels) and metabolites (nodes, lower case labels) with corresponding labels. The corresponding Stoichiometric matrix (right, $S \in \mathbb{R}^{13 \times 10}$, $N \in \mathbb{R}^{13 \times 8}$, $B \in \mathbb{R}^{13 \times 2}$), with reversible reactions shown in green, non-reversible reactions shown in blue, and exchange reactions shown in red. The upper bound ($ub$) of internal reactions is unconstrained, whereas the lower bound ($lb$) is limited by reaction directionality or by the maximum uptake rate, which can be seen as a constraint on uptake of extracellular glucose (glc_D[e]) or secretion of dihydroxyacetone phosphate (dhap[e]) from the environment. All metabolite and reaction abbreviations are with respect to the namespace in www.vmh.life[19].

A thermodynamically feasible flux does not exist when at least one combination of bounds on reaction rates forces net flux around a stoichiometrically balanced cycle [8]. Consider the following stoichiometrically balanced cycle of reactions $A \rightleftharpoons B \rightleftharpoons C \rightleftharpoons A$. If the bounds are such that $A \to B \to C \to A$ is the only feasible net flux, then the network containing this cycle cannot carry a thermodynamically feasible steady state net flux. The second law of thermodynamics requires that a chemical potential difference between substrates and products is required for net flux. Let $\mu_A$ denote the chemical potential of metabolite $A$. Given net flux $A \to B$ the second law of thermodynamics

requires $\mu_A > \mu_B$ and similarly $\mu_B > \mu_C$ and $\mu_C > \mu_A$. However, the first pair of inequalities imply $\mu_A > \mu_C$, which is inconsistent with the last, unless one assigns more than one chemical potential to $C$ at the same instant. However, to do so would be inconsistent with energy conservation, which requires each row of a stoichiometric matrix to be associated with a single chemical potential, assuming that each row corresponds to a well mixed compartment [9].

Stoichiometrically balanced cycles are biochemically faithful network topological features that are omnipresent in genome-scale models. It is not the presence of a stoichiometrically balanced cycle, per say, that presents a problem. It is the thermodynamically inconsistent specification of bounds, which force net flux around a stoichiometrically balanced cycle and prevents the prediction of a thermodynamically feasible net flux. As recognised by Desouki et al. [8], provided that a model does not contain a combination of thermodynamically inconsistent bounds, minimisation of the one-norm of net flux, subject to additional constraints maintaining the direction of net flux, is guaranteed to remove that part of a steady state flux vector that is thermodynamically infeasible. This approach is attractive, as it is based on linear optimisation, but it has yet to be leveraged for generation of thermodynamically consistent models. Generation of thermodynamically consistent models should, in principle, increase the biochemical fidelity of constraint-based models as it opens up the possibility of leveraging established methods to efficiently compute thermodynamically feasible steady state fluxes [10, 8], that assume an input model is thermodynamically consistent. Given a generic reconstruction or model, all established model extraction algorithms, including GIMME [4], iMAT [35], MBA [16], mCADRE [34], FastCore [32], and CarveMe [17], and also gap filling algorithms [28], extract a specific model based on either a binary (present/absent) or weighted assignment of reactions desired to be present in a specific model, where each reaction is (net) flux consistent. However, the models generated from these algorithms are not guaranteed to admit thermodynamically feasible net flux.

## Development of the protocol

**Overview** In this protocol we present the XomicsToModel pipeline, which, given a generic reconstruction and omics data, enables extraction of a context-specific, genome-scale metabolic model, which is stoichiometrically, thermodynamically and flux consistent. The pipeline was developed to generate a context-specific model of a dopaminergic neuronal metabolism [24], given a generic genome-scale human metabolic reconstruction, Recon3D [7], and transcriptomic, metabolomic and bibliomic data. Substantia nigra dopaminergic neurons are the most susceptible to degeneration in Parkinson's disease, a progressive, neurodegenerative movement disorder, and its biochemical mechanisms remain poorly understood [2]. Finally, although the operation of XomicsToModel pipeline is exemplified for extraction of a specific metabolic model from a generic metabolic model, its implementation is envisaged to enable omics data integration and extraction of a physicochemically consistent mechanistic model from any generic biochemical network.

**Input** The input to the XomicsToModel pipeline is a biochemical network and a set of omics data. The biochemical network may be a reconstruction, or a model, and is not required to be stoichiometrically, thermodynamically or flux consistent. The XomicsToModel pipeline allows a flexible and modular integration of transcriptomic, proteomic, and metabolomic data, as well as bibliomc data abstracted from literature curation. In each case, the input data may be qualitative (present, absent, unspecified), semi-quantiative, quantitative, or combinations thereof. The XomicsToModel pipeline is complemented by functions to automatically import omics data.

The application of the XomicsToModel pipeline to extract a dopaminergic neuronal metabolic model [24] demonstrates its flexibility with respect to incorporation of a variety of qualitative and quantitative constraints. For example, the presence or absence of metabolites in the culture medium, as well as quantitative metabolite exchange reaction rates, which were applied using quadratic optimisation to set exchange reaction bounds, weighted by the inverse of measurement uncertainty. Optionally, the XomicsToModel pipeline enables extension of an input reconstruction or model, with manually specified metabolic reactions, in the case where a generic model is missing certain key pathways relevant for a system of interest.

**Constraint relaxation** The XomicsToModel pipeline incorporates a series of tests for flux and optionally thermodynamic consistency after each step that reduces the size of the feasible set, e.g., after removal of generic metabolites or reactions assigned not to be present in a specific model. This approach ensures the early detection of infeasible constraints resulting from inaccurate or inconsistent experimental data. Detection of inconsistency is followed by algorithmic relaxation of constraints (e.g. bounds) to render the draft model feasible. Specifically, the XomicsToModel pipeline automatically searches for the minimal number of constraint relaxations required to admit a flux-consistent model in the case that inconsistent or incorrect omics data renders a draft model infeasible. This approach can be used to feedback into data processing or input to avoid model infeasibility (Figure 3), which is often a challenge with omics data integration.

The application of the XomicsToModel pipeline to extract a dopaminergic neuronal metabolic model [24] presented a particular challenge for constraint-based modelling because the extracted model was required to be representative of neurons that do not grow. Therefore, the maximisation of biomass growth could not be used as an objective. Instead, a set of coupling constraints [29] were added to represent cell maintenance requirements, e.g., for the turnover of key metabolites. However, if incorrectly scaled with respect to constraints on exchange reactions, the application of coupling constraints may generate an infeasible draft model. Therefore, an algorithmic proposition of a minimal number of constraint relaxations accelerates the identification of biochemically inconsistent constraints and refinement of input data.

**Model extraction** The XomicsToModel pipeline is compatible with various model extraction algorithms (cf Table 1), with an established interface to FastCore [32]. In addition, the XomicsToModel pipeline builds upon thermoKernel, a novel algorithm used to extract the dopaminergic neuronal metabolic model [24], but applicable for the extraction of any context-specific model that is required to be stoichiometrically, thermodynamically and flux consistent. The ability of thermoKernel to enforce thermodynamic consistency during the model extraction process opens up new possibilities for data integration. Thermodynamic consistency is implemented by constraining the possible relationships between reaction flux and metabolite chemical potential (detailed below). Consequently, it is possible to directly specify a ternary (presence, absence, unspecified) or weighted assignment of metabolites desired to be present in a specific model. More generally, it is possible to specify a ternary or weighted assignment of metabolites and reactions desired to be present in a specific model.

This enables the thermoKernel algorithm to extract a thermodynamically consistent model that is a trade-off between ternary or weighted assignments of rows and columns of a stoichiometric matrix. The application of the XomicsToModel pipeline to extract a dopaminergic neuronal metabolic model [24] demonstrates this flexibility with ternary and weighted specification of metabolites and reactions desired to be present in the dopaminergic neuronal model. The XomicsToModel enables integration of, for example, qualitative metabolomic data, where the metabolite presence can now be applied directly as context-specific data input.

**Ensemble modelling** To exploit the XomicsToModel pipeline flexibility, a supplementary XomicsToMultipleModels function is provided to generate an ensemble of context-specific genome-scale models by varying the specific data and technical parameters used in the model extraction process. Moreover, supplementary functions are provided to estimate the predictive capacity of an extracted model, given independent data. These features will facilitate analysis by complementary software with capabilities for machine learning from model ensembles [18]. The application of the XomicsToModel pipeline to extract an ensemble of dopaminergic neuronal metabolic models is presented elsewhere [24].

## Applications of the XomicsToModel pipeline

The XomicsToModel pipeline applies to any situation where one has a generic (or universal) reconstruction (or model) and seeks to extract a specific subset of it, based on transcriptomic, proteomic,

or metabolomic data, or combinations thereof. There is no restriction with respect to the identity of the species of the input reconstruction or model, nor to the number of species represented in either the input or output models. The most computationally intensive steps are the identification of the largest thermodynamically consistent subset of an input reconstruction or model and the extraction of a thermodynamically consistent subset of minimal size, both of which are achieved with the thermoKernel algorithm. As the thermoKernel algorithm is based on a sequence of linear optimisation problems, its performance scales accordingly. Ternary specification of metabolite presence/absence/unspecified, reaction activity/inactivity/unspecified, gene activity/inactivity/unspecified, and combinations thereof are all possible. Similarly, weighted specification, to bias for or against the inclusion of metabolites, reactions, or both, based on input transcriptomic, proteomic, or metabolomic data is also possible, also in combination with a ternary specification. The XomicsToModel pipeline is interfaced via a function, but multiple options can be specified to enable modular usage of the XomicsToModel pipeline to implement model extraction priorities reflective of a wide variety of scenarios. For example, one can switch from thermoKernel to a different model extraction algorithm, such as FastCore[32], as desired. As such, it is envisaged that the XomicsToModel pipeline will have widespread applicability.

## Comparison with other methods

A systematic comparative evaluation of model extraction methods by Opdam et al. [21] has shown that using various data types, such as omics data, for model construction and validation, as well as careful selection of gene expression cut-offs, lead to higher model accuracy when comparing with experimental exometabolomic data. Table 1 compares other algorithms for extraction of a context-specific model, given a generic model of a single organism, with the XomicsToModel pipeline. Overall, in comparison with similar methods, the XomicsToModel pipeline provides several additional capabilities in terms of its ability to represent additional constraints, its flexibility concerning data integration and its ability to suggest relaxations to recover from the application of inconsistent constraints.

**Table 1:** Comparison of technical features of algorithms for extraction of a context-specific model, given a generic model of a single organism. MILP, mixed integer linear programming; FVA, flux variability analysis, fastFVA[12], computational acceleration of flux variability analysis; QC-LP, quasiconcave sequence of linear programs; DCA-LP, difference of convex function sequence of linear programs. $*$ denotes subsequently implemented in the COBRA Toolbox [14].

| | GIMME | iMAT | MBA | INIT | mCADRE | FastCore | XomicsToModel |
|---|---|---|---|---|---|---|---|
| Citation | [4] | [35] | [16] | [1] | [34] | [32] | N/A |
| Year | 2008 | 2010 | 2010 | 2012 | 2012 | 2014 | 2022 |
| Active metabolite list | × | × | × | × | × | × | ✓ |
| Metabolite weights $\in \mathbb{R}$ | × | × | × | × | ✓ | × | ✓ |
| Active reaction list | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Reaction weights $\in \mathbb{R}$ | × | × | ✓ | ✓ | ✓ | × | ✓ |
| Coupling constraints | × | × | × | × | × | ✓* | ✓ |
| Thermodynamic consistency | × | × | × | × | × | × | ✓ |
| Constraint relaxation | × | × | × | × | × | × | ✓ |
| Algorithmic approach | MILP | MILP | FVA | MILP | fastFVA | QC-LP | DCA-LP |

## Experimental Design

To run the XomicsToModel pipeline, users must first obtain and prepare a generic reconstruction or model and the context-specific data used to extract the model. Furthermore, one must either accept the default set of technical parameters or specify the parameters to suit a particular context (as detailed below).

**Generic reconstruction or model:** A generic genome-scale reconstruction or model represents a metabolic network constructed from the amalgamation and manual literature curation of metabolic reactions that occur in various cell types or organisms.

**Context-specific data:** Context-specific data represents the genotype or phenotype of a specific biological system. It can be obtained through a review of the existing literature or derived experimentally from a biological system or both. The information can be entered manually or using the function preprocessingOmicsModel described in the supplementary information. The following context-specific information is currently supported by the XomicsToModel pipeline:

- **Bibliomic data:** Data derived from a manual reconstruction following a review of the existing literature. This includes data on the genes, reactions, or metabolites known to be present or absent in the studied biological system. Furthermore, bibliomics data can define a set of coupled reactions or a set of constraints for model reactions based on phenotypic observations.
- **Transcriptomic data:** Measured gene expression levels in the studied biological system. It is used to estimate qualitatively and quantitatively the activity of reactions associated with the detected genes based on the gene-protein-reaction association (GPR, Figure 2). Transcriptomic data can be provided in fragments per kilobase million (FPKM) or raw counts.
- **Proteomic data:** Measured protein levels in the studied biological system. Similarly to transcriptomics data, it can be used to estimate quantitatively which reactions in the metabolic model should be considered active based on the gene-protein-reaction association (Figure 2).
- **Metabolomic data:** The average and standard deviation of metabolite concentrations measured in cell media, biofluids, tissues, or organisms translated into flux units, which may be $mmol/gDW/h$, $\mu mol/gDW/h$ or $nmol/gDW/h$ depending on one's choice. Metabolites detected experimentally can be assigned to be present in the metabolic model. Measured uptakes and secretions rates in growth media or biofluids can be used to quantitatively constrain the uptakes and secretions of the model. Furthermore, growth conditions information such as growth media composition can also be provided to constrain available uptakes in the model.

Note that while the XomicsToModel supports input of all of these different type of context-specific information simultaneously, to generate a model it is sufficient to have any subset of these different types of context-specific information, e.g., if there is no proteomic data, it is still possible to generate a context-specific model.

**Technical parameters:** With these options, technical constraints can be added to the model, as well as setting the parameters for model extraction or debugging. If they are not specified, default values are used.

- **Bounds:** Parameters that define the minimum and maximum flux value in the model or the minimum non-zero flux value tolerance.
- **Exchange reactions:** Instructions to close or leave open demand and sink reactions and whether exchange reactions should be added based on the metabolomic data.
- **Extraction options:** Parameters required by solvers and for the context-specific model extraction algorithms.
- **Data-specific parameters:** Parameters that define the minimum level of transcript/protein to be considered as present in the model (threshold) and whether the transcripts below the defined threshold should be removed from the generic reconstruction or model.
- **Debugging options:** Debugging parameters that allow the user to evaluate the output of consecutive steps within the pipeline after they have been completed.

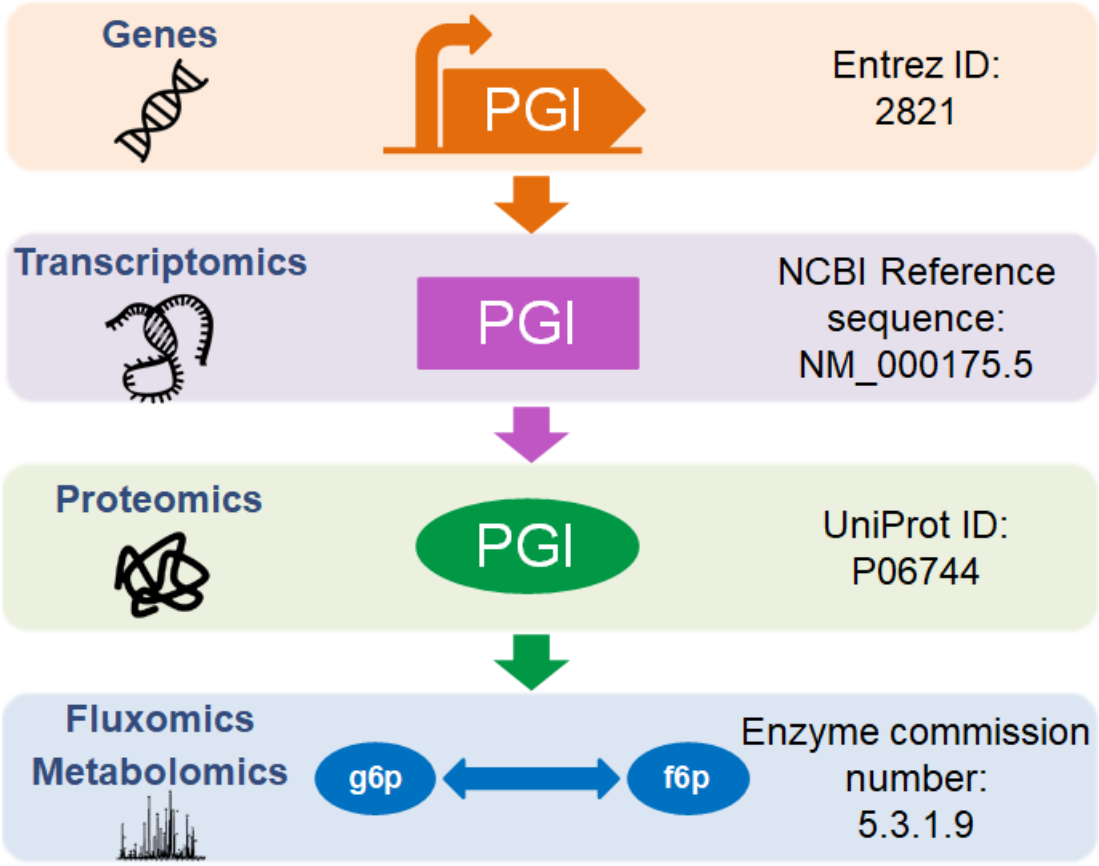



**Figure 2:** The gene-protein-reaction association or GPRs are boolean operators that describe the interactions of genes, transcripts, proteins, and reactions. For the reaction glucose-6-phosphate isomerase (PGI) to occur, the gene PGI must be activated (Entrez identifier: 2821) leading to the transcription of the PGI mRNA (NCBI Reference Sequence: NM 000175.5). The mRNA is then translated into the glycolytic enzyme PGI (UniProt identifier: P06744). Finally, the PGI enzyme (Enzyme Commission Number: 5.3.1.9) catalyse a reaction that interconverts glucose-6-phosphate (VMH identifier: g6p) and fructose-6-phosphate (VMH identifier: f6p)

### Required expertise

The pipeline is written in MATLAB, a programming language that is easy to learn but also powerful due to the numerous toolboxes for numerical and symbolic computing. This protocol can be implemented by anyone who has basic MATLAB programming skills and the fundamentals of constraint-based modelling. The computational load associated with this protocol is determined by the network's size. A desktop personal computer is sufficient to generate a stoichiometrically, thermodynamically and flux consistent, context-specific, genome-scale metabolic model.

### Limitations

This protocol focuses on extracting a context-specific genome-scale model from an existing reconstruction by integrating omics data from a biological system. It does not cover generation of a high-quality genome-scale metabolic reconstruction from scratch Norsigian et al. [20] and Thiele and Palsson [27], or the analysis of existing models [14]. Additionally, the integration of omics data presents several challenges since they are generated using a variety of platforms, affecting storage and data formats significantly. The integration of omics data necessitates data in specific formats. Therefore, individual omics data must be pre-processed. Furthermore, experimental errors, such as data processing or measurement errors, can propagate through the extraction of a metabolic network that is not a faithful representation of the original biochemical system. In this situation, a significant loss of predictive capacity will be evident. Therefore, it is not recommended to extract a context-specific model and claim that it has high predictive accuracy without commensurate comparison with some independent experimental data.

# MATERIALS

## Equipment

### Input data

The COBRA Toolbox supports commonly used data formats for model description such as Systems Biology Markup Language (SBML) and Excel sheets (.xlsx). MATLAB supports a wide range of text and spreadsheet formats that can be used to provide context-specific data for the XomicsToModel pipeline.

### Required hardware

- A computer with at least 8 GB of RAM and any 64-bit Intel or AMD processor. ▲ **CRITICAL** Depending on the size of the model more processing power and more memory might be needed.
- A hard drive with at least 10 GB of space available.

### Required software

- An operating system that is MATLAB qualified (https://mathworks.com/support/sysreq.html). ▲ **CRITICAL** To make sure that the operating system is compatible with the MATLAB version check the requirements at https://mathworks.com/support/sysreq/previous_releases.html.
- MATLAB (MathWorks, https://mathworks.com/products/matlab.html), version R2021+. Install MATLAB and its license by following the official installation instructions (https://mathworks.com/help/install/ug/installmathworks-software.html). ! **CAUTION** Version R2021+ or above is recommended for running MATLAB live scripts (.mlx files). ▲ **CRITICAL** No support is provided for versions older than R2021. MATLAB is released on a twice-yearly schedule, and the latest releases of MATLAB may not be compatible with the existing solver interfaces, necessitating an update of the MATLAB interface provided by the solver developers, an update of the COBRA Toolbox, or both.
- The COBRA Toolbox version 3.4 [14], or above. To install the COBRA Toolbox follow the instructions on https://github.com/opencobra/cobratoolbox. ! **CAUTION** If an installation of COBRA Toolbox is already present, update the repository from MATLAB, via terminal, or Git Bash. ▲ **CRITICAL** Check that all of the system requirements in https://opencobra.github.io/cobratoolbox/docs/requirements.html are met.

### Solvers

Table 2 provides an overview of the optimisation solvers supported by the XomicsToModel pipeline.

**Table 2:** An overview of the optimisation solvers XomicsToModel supports.

| Name | Version | Interface |
|---|---|---|
| GLPK | 2.7+ | glpk |
| GUROBI | 7.0+ | gurobi |
| ILOG CPLEX | 10-12.10 | ibm_cplex |
| MATLAB | R2014b+ | matlab |

## Equipment setup COBRA Toolbox

If one is not a developer of COBRA Toolbox code, update the COBRA Toolbox from within MATLAB by running the following command:

```
>> updateCobraToolbox
```

Or, update from the terminal (or Git Bash) by running the following from within the COBRA Toolbox directory.

```
$ cd cobratoolbox # change to the local cobratoolbox directory
$ git checkout master # switch to the master branch
$ git pull origin master # retrieve changes
```

In the case that the update of the COBRA Toolbox fails or cannot be completed, clone the repository again. ▲ **CRITICAL** The COBRA Toolbox can be updated as described above only if no changes have been made to the code in one's local clone of the official online repository. To contribute any local edits to the COBRA Toolbox code into the official online repository please follow the MATLAB.devTools guidelines [14] (https://opencobra.github.io/MATLAB.devTools/stable/) .

## PROCEDURE

The XomicsToModel pipeline is implemented in a sequence of twenty-one steps (Figure 3). To generate a thermodynamic-flux-consistent, context-specific, genome-scale metabolic model, the pipeline requires three inputs: a generic COBRA reconstruction or model, context-specific information and technical parameters. Below, for each step, all the user-defined options used are listed and described.

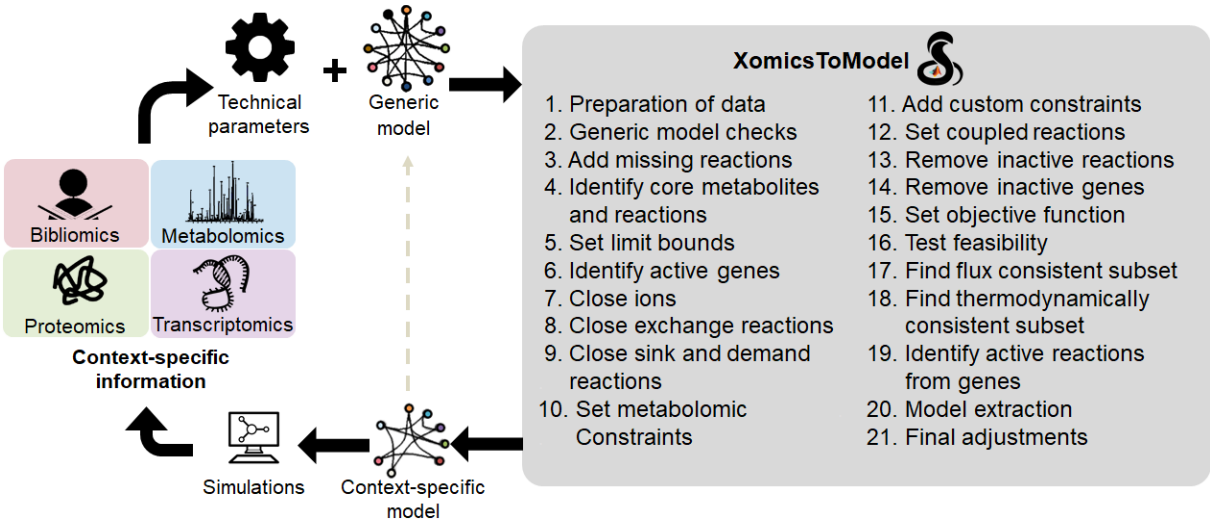



**Figure 3:** Overview of the XomicsToModel pipeline.

It is possible to create genome-scale models iteratively and systematically using the XomicsToModel pipeline. The generated models can be used to design or interpret the information based on bibliomic data from manual literature curation (red), as well as metabolomic (blue), proteomic (green) or transcriptomic (purple) data. Furthermore, experimental data can be used to validate or refine the context-specific model. The XomicsToModel pipeline (grey) extracts a new model by integrating a generic model with context-specific information based on the technical parameters such as the extraction algorithm, method for identifying active genes, or maximum reaction rates.

To extract a context-specific model from a generic model using omics data, first launch the COBRA Toolbox [14] and specify the optimisation solver to be used. The XomicsToModel pipeline can be run, if at least one input variable, a generic model is provided ,which must have the properties specified in https://github.com/opencobra/cobratoolbox/blob/master/docs/source/notes/COBRAModelFields.md. The user-defined context-specific information, specificData, and technical parameters, param, are optional, and if not specified, default values for the required fields will be assigned.

```
>> initCobraToolbox
>> changeCobraSolver('gurobi', 'LP');
>> changeCobraSolver('gurobi', 'QP');
>> [contextSpecificModel, modelGenerationReport] = XomicsToModel(genericModel, ...
    specificData, param)
```

**? TROUBLESHOOTING**

The name and types of the fields for the specificData and param variables must be identical to how they are described in each step for the XomicsToModel pipeline to identify them. The supplementary information describes a tool for automatically generating the specificData variable, as well as a tool for creating multiple models varying the user-defined options and for testing the accuracy of the extracted model.

## 1 | Preparation of data ● TIMING $\sim 5$ s

**Description:**

The first step performs data harmonisation of the user-defined options in specificData and param, and generic model's fields to comply with naming conventions and formats expected by the XomicsToModel pipeline. Missing specificData and param fields are identified and default values are assigned. The COBRA Toolbox function getCobraSolverParams is used to identify the primal feasibility tolerance. Furthermore, the draft model is created based on the generic model. In addition, if a filename for the diary has been specified in param.diaryFilename the diary file is opened, where many intermediate results of the pipeline are printed for debugging.

**Usable variables:**

- param.diaryFilename - The name (and location) of a diary file with the printed pipeline output (Default: 0).

## 2 | Generic model checks ● TIMING $\sim 1$ s

**Description:**

In this section, the draft model fields that may cause inconsistency when adding or removing reactions are removed. For example, Recon3D [7], includes a cell compartment representing the mitochondrial intra-membrane space, but this is not necessary when using an algorithm to compute thermodynamically feasible fluxes. Therefore, if 'thermoKernel' is chosen as the tissue-specific solver, this compartment is removed and metabolites are reassigned to the cytosol.

Regardless of the optimisation objective selected, every solution to Equation 1 must satisfy the steady state constraints $Sv = 0$, reaction rate bounds $l \leq v \leq u$, and optionally coupling constraints $Cv \leq d$, implying that the system of inequalities is feasible; if the model is infeasible in this step, an error will be generated **? TROUBLESHOOTING** .

**Usable variables:**

- param.tissueSpecificSolver - The name of the tissue-specific solver to be used to extract the context-specific model (Possible options: 'thermoKernel' and 'fastcore'; Default: 'thermoKernel').
- param.printLevel - Level of verbose that should be printed (Default: 0).

## 3 | Add missing reactions ● TIMING $\sim 30$ s

**Description:**

Reactions specified in the specificData.rxns2add, are added to the draft model using the function addReaction. Multiple data can be included but only the reaction identifier and the reaction formula are mandatory. **▲ CRITICAL STEP** When a metabolic reaction is added, default bounds are set based on the characters used in the reaction formula to separate the substrates and products.

- Forward (->): $l_j = 0$; $u_j =$param.TolMaxBoundary.
- Reverse (<-): $l_j =$param.TolMinBoundary; $u_j = 0$.
- Reversible (<=>) $l_j =$param.TolMinBoundary; $u_j =$param.TolMaxBoundary.

The default values for missing reaction names, metabolic pathways, and gene rules are *Custom reaction*, *Miscellaneous*, and an empty cell, respectively. **! CAUTION** If a reaction is already present in the draft model, it will be replaced by the new reaction. By associating a new gene identifier with a reaction, a new gene is added to the draft model.

Given a generic reconstruction or model, the findStoichConsistentSubset function approximately solves the problem to extract the largest subset of it that is stoichiometrically consistent [14]. Inconsistent metabolites and reactions are removed, however, if this affects feasibility, an error is generated **? TROUBLESHOOTING** . A set of reactions is stoichiometrically consistent if every reaction in that set is mass balanced [11]. Recall again that, in the split $S := [N, B]$, the matrix $N$ represents stoichiometrically consistent internal reactions, while the matrix $B$ represents stoichiometrically inconsistent external reactions. It is vital to appreciate that the problem to extract the largest stoichiometrically consistent subset is combinatorially complex, and therefore, at genome-scale, only approximation to the actual solution is achievable in practice. The quality of the approximation is improved if heuristics are used to warm-start the algorithm with a specification of the reactions that are accepted as being stoichiometrically inconsistent, e.g., external reactions, as specified as false in model.SIntRxnBool. If model.SIntRxnBool is not provided, the function findSExRxnInd attempts to heuristically distinguish internal and external reactions, based on reaction naming conventions (e.g. 'EX_' prefix for an exchange reaction) and reaction stoichiometry (any reaction with only one nonzero stoichiometric coefficient in the corresponding row is an external reaction). Also in this step, the gene-protein-reaction rules are updated for the newly added reactions, and the reaction-gene-matrix is regenerated. In addition, different boolean vectors indicating stoichiometrically consistent and inconsistent metabolites and reactions are added to the draft model. If param.printLevel is greater than zero, a table with a summary of the draft model's stoichiometric consistency will be printed (Table 3).

**Usable variables:**

- specificData.rxns2add - Table containing the identifier of the reaction to be added, its name, the reaction formula, the metabolic pathway to which it belongs, the gene rules to which the reaction is subject, and the references. (Default: empty).

| rxns | rxnNames | rxnFormulas | subSystems | rxnGrRules | rxnReferences |
|---|---|---|---|---|---|
| newRxn1 | Oxidation | A + O -> AO | Glycolysis | gene1 or gene2 | PMID: **** |
| newRxn2 | Reduction | AH + B <=> A + BH | Glycolysis | gene1 | PMID: **** |
| : | : | : | : | : | : |

- param.TolMinBoundary - The reaction flux lower bound minimum value (Default: -1000).
- param.TolMaxBoundary - The reaction flux upper bound maximum value (Default: 1000).
- param.debug - Logical, should the pipeline save its progress for debugging (Default: false).
- model.SIntRxnBool - Logical vector, true if a reaction is heuristically considered an internal reaction, false for an external reactions (Default: false).

## 4 | Identify core metabolites and reactions ● TIMING $\sim 10$ s

**Description:**

The XomicsToModel pipeline uses several fields from the specificData variable to identify the set of metabolites and reactions that must be active in the draft model. These optional fields include the reaction specified as the objective function, the metabolic reactions to add, the metabolites and reactions identified as active based on bibliomic data, the reactions whose bounds are restricted based on bibliomic data, the coupled reactions (See Step 12), the exchange reactions for metabolites present in the growth media, and the exchange reactions for metabolites shown to be secreted or uptaken based on exometabolomic data. All of these reactions are added to the active reaction list, which will be appended or shortened if necessary to keep the model stoichiometrically, thermodynamically and flux consistent.

### Usable variables:

- param.setObjective - Cell string indicating the linear objective function to optimise (Default: none).
- specificData.absentMetabolites - List of metabolites known to be absent based on the bibliomic data (Default: empty).
- specificData.activeReactions - List of reactions know to be active based on bibliomic data (Default: empty).
- specificData.coupledRxns - Table containing information about the coupled reactions. This includes the identifier for the coupling constraint (couplingConstraintID), the list of coupled reactions (coupledRxnsList), the coefficients of those reactions (c, given in the same order), the right hand side of the constraint (d), the constraint sense or the directionality of the constraint (dsence), and the reference (Default: empty).

| couplingConstraintID | coupledRxnsList | c | d | dsense | reference |
|---|---|---|---|---|---|
| cRxn1 | crxn1, crxn2 | 1 1 | 2.0250 | G | PMID: **** |
| cRxn2 | crxn1, crxn3 | 1 -1 | 0.2265 | G | PMID: **** |
| : | : | : | : | : | |

- specificData.exoMet - Table with measured exchange fluxes, e.g., obtained from a exometabolomic analysis of fresh and spent media. It includes the reaction identifier, the reaction name, the measured mean flux, standard deviation of the measured flux, the flux units, and the platform used to measure it.

| rxns | rxnNames | mean | SD | units | platform |
|---|---|---|---|---|---|
| EX_mMet1 | Exchange of mMet1 | 100 | 8x$10^{-4}$ | $\mu mol/gDW/h$ | LC-MS |
| EX_mMet2 | Exchange of mMet2 | 4x$10^{-3}$ | 3x$10^{-4}$ | $\mu mol/gDW/h$ | GC-MS |
| : | : | | | | : |

- specificData.inactiveReactions - List of reactions known to be inactive based on bibliomic data (Default: empty).
- specificData.mediaData - Table containing the fresh media concentrations. Contains the reaction identifier, the maximum uptake ($\mu mol/gDW/h$) assuming it is equal to the concentration of the metabolite in fresh medium divided by the length of time before medium exchange, and the medium concentration ($\mu mol/l$; Default: empty).

| rxns | mediumMaxUptake | mediumConcentrations |
|---|---|---|
| EX_met1 | -220 | 1345 |
| EX_met2 | -14916 | 191021 |
| : | : | : |

- specificData.presentMetabolites - List of metabolites known to be present based on the bibliomic data (Default: empty).
- specificData.rxns2add - Table containing the identifier of the reaction to be added, its name, the reaction formula, the metabolic pathway to which it belongs, the gene rules to which the reaction is subject, and the references. (Default: empty).

| rxns | rxnNames | rxnFormulas | subSystems | rxnGrRules | rxnReferences |
|---|---|---|---|---|---|
| newRxn1 | Oxidation | A + O -> AO | Glycolysis | gene1 or gene2 | PMID: **** |
| newRxn2 | Reduction | AH + B <=> A + BH | Glycolysis | gene1 | PMID: **** |
| : | : | : | : | : | : |

- specificData.rxns2constrain - Table containing the reaction identifier, the updated lower bound (lb), the updated upper bound (ub), a constraint description, and any notes such as references or special cases (Default: empty).

| rxns | lb | ub | constraintDescription | Notes |
|---|---|---|---|---|
| rxn1 | 0 | | Close uptake | PMID: **** |
| rxn2 | 0.0.956 | 1000 | Turnover constraints | PMID: **** |
| : | : | : | : | : |

## 5 | Set limit bounds ● TIMING $\sim 1$ s

**Description:**
Considering the constraints $l \leq v \leq u$ shown in Equation 1, where $l$ is the lower bound and $u$ the upper bound, each lower bound in the draft model such that $l_j = min(l)$ and $l_j \leq TolMinBoundary$ is set to param.TolMinBoundary and each upper bound such that $u_j = max(u)$ and $u_j \geq TolMaxBoundary$ is set to param.TolMaxBoundary. **! CAUTION** There should be no lower bound that exceeds an upper bound. The model's feasibility is tested with the new bounds **? TROUBLESHOOTING** . If param.printLevel is greater than zero, the modified bounds will be printed.

**Usable variables:**
- param.TolMinBoundary - The reaction flux lower bound minimum value (Default: -1000).
- param.TolMaxBoundary - The reaction flux upper bound maximum value (Default: 1000).

## 6 | Identify active genes ● TIMING $\sim 10^2$ s

**Description:**
The list of active genes is specified by NCBI Entrez gene identifiers, which are unique integers for genes or other loci. **! CAUTION** A gene must be present in the draft model to be considered as active. Missing genes can be added using the gene-protein-reaction rules in specificData.rxns2add.rxnGrRules (Step 3). The active genes list is composed based on the data from multiple sources:

- Bibliomics: Genes classified as active or inactive based on published data.
- Proteomics: Genes that, based on the list of detected proteins, are active based on their (normalised) peak areas. If $peakArea \geq$ param.thresholdP, the gene is considered active. **! CAUTION** Proteomics data is usually annotated with UniprotKB. However, for data integration with the model, UniprotKBs need to be converted into corresponding Entrez identifiers.
- Transcriptomics: Genes that, based on the list of detected transcripts, are active based on their expression levels (FPKM, Fragments Per Kilobase of transcript per Million mapped reads or raw counts). If $\log(expressionLevel) \geq$param.thresholdT, the gene is considered active.

Furthermore, the logarithm of the levels of gene expression in specificData.transcriptomicData are used to link gene expression levels to associated reactions using the function mapExpressionToReactions. In short, the resulting model.expressionRxns field is based on the gene relationship described in the gene-protein-reaction rules: an **AND** in the gene-protein-reaction rules is assigned $min(lb)$, and an **OR** is assigned a $max(ub)$. For genes that are not found in the transcriptomics data, no expression value is added (NaN is default). If param.inactiveGenesTranscriptomics is used, the genes that fall below the specified threshold are added to the list specificData.inactiveGenes. In the event of a discrepancy, for example, an inactive gene according to manual literature curation but active according to proteomic or transcriptomic data, preference will be given based on the option specified by param.curationOverOmics. By default, bibliomic data takes precedence over other omics data, i.e., param.curationOverOmics = true. Genes not found in the draft model are removed from the list of active genes. Two fields are added to the model to indicate the gene expression value of the genes present in the draft model model.geneExpVal and the expression of the corresponding reactions (model.expressionRxns) based on the function mapExpressionToReactions. If param.printLevel is greater than zero, a histogram of reaction expression is printed.

**Usable variables:**
- specificData.transcriptomicData - Table with a column with Entrez ID's and a column for the corresponding transcriptomics expression value in FPKM or raw counts (Default: empty).

| gene | expVal |
|---|---|
| entrezID1 | 220 |
| entrezID2 | 14916 |
| : | : |

- param.thresholdT - The transcriptomic cutoff threshold for determining whether or not a gene is active (Default: $log2(1)$).
- specificData.proteomics - Table with a column with Entrez ID's and a column for the corresponding protein levels (Default: empty).

| gene | expVal |
|---|---|
| entrezID1 | 24 |
| entrezID2 | 78 |
| : | : |

- param.thresholdP - The proteomic cutoff threshold for determining whether or not a gene is active (Default: $log2(1)$).
- specificData.activeGenes - List of Entrez identifier of genes that are known to be active based on the bibliomic data (Default: empty).
- param.curationOverOmics - Logical, indicates whether curated data should take priority over omics data (Default: false).
- param.inactiveGenesTranscriptomics - Logical, indicate if inactive genes in the transcriptomic analysis should be added to the list of inactive genes (Default: true).
- specificData.inactiveGenes - List of Entrez identifier of genes known to be inactive based on the bibliomics data (Default: empty).

## 7 | Close ions ● TIMING $\sim 1$ s

**Description:**

Reactions involving ion exchange can be closed in this step if param.closeIons = true, in which case the lower and upper bounds for ion exchange reactions are closed, preventing net exchange of ions with the environment. By doing so, it is possible to defer modelling of ion exchange, e.g., to represent action potentials in neurons since they represent a dynamic trajectory of ion concentrations, while prioritising metabolic activities.

**Usable variables:**

- param.closeIons - Logical, it determines whether or not ion exchange reactions are closed. (Default: false).

## 8 | Close exchange reactions ● TIMING $\sim 1$ s

**Description:**

If the variable param.closeUptakes = true, the $lb$ of the exchange reactions in the draft model are closed (set to 0). This will allow the draft model to only take up the metabolites specified by the bibliomics, growth media, or metabolomics data. It is advised to set param.closeUptakes to true for biological systems derived from cell cultures. If param.printLevel is greater than zero, the new constraints will be printed along with statistics such as the size of the stoichiometric matrix, the number of exchange reactions closed, the number of exchange reactions in active reactions, and the number of exchange reactions in reactions to constrain based on manual literature curation.

**Usable variables:**

- specificData.rxns2constrain - Table containing the reaction identifier, the updated lower bound, the updated upper bound, a constraint description, and any notes such as references or special cases (Default: empty).

| rxns | lb | ub | constraintDescription | Notes |
|---|---|---|---|---|
| rxn1 | 0 | | Close uptake | PMID: **** |
| rxn2 | 0.0.956 | 1000 | Turnover constraints | PMID: **** |
| : | : | : | : | : |

- param.closeUptakes - Logical, decide whether or not all of the uptakes in the draft model will be closed (Default: false).

## 9 | Close sink and demand reactions ● TIMING $\sim 10$ s

**Description:**

The demand and sink reactions are closed based on the value option indicated in param.nonCoreSinksDemands. Only the demand and sink reactions in specificData.activeReactions will remain open. If param.printLevel is greater than zero, the number and names of closed demand or sink reactions are printed.

The solution of a feasible model must satisfy the constraints $Sv = 0$ and $lb \leq v \leq ub$ and optionally coupling constraints $Cv \leq d$. The newly added constraints may not all be consistent with a steady state flux at the same time, implying that the system of inequalities and the model are infeasible, which could be caused by an incorrectly specified reaction bound. To solve the infeasibility we use relaxedFBA, which is an optimisation problem that approximately relaxes the minimum number of reaction bounds in order to make a Flux Balance Analysis problem feasible. The relaxation options are set in param.relaxOptions. The old bounds from spent media metabolites are saved in two new fields in the draft model, along with a field identifying the exchange reactions that were relaxed. relaxedFBA is a flexible function for relaxing.

**Usable variables:**

- specificData.activeReactions - List of reactions known to be active based on bibliomic data (Default: empty).
- param.nonCoreSinksDemands - The type of sink or demand reaction to close is indicated by a string (Possible options: 'closeReversible', 'closeForward', 'closeReverse', 'closeAll' and 'closeNone'; Default: 'closeNone').
- param.relaxOptions - A structure array with the relaxation options (Possible options are described in detail in Cobra Toolbox 3.0 protocol, Step 23 [14], Default: param.relaxOptions.steadyStateRelax = 0).

## 10 | Set exometabolic constraints ● TIMING $\sim 10$ s

**Description:**

The extracellular metabolic constraints can be obtained from two different sources: fresh cell culture media metabolite concentrations (as specified by the manufacturer, or as specified by combining known quantities of metabolically specified media), or quantitative extracellular metabolomic measurements of fresh and spent media (assuming a metabolomic steady state), or both. The difference between spent and fresh media concentrations for extracellular metabolites is assumed to equal the rate of exchange of the metabolite between the cells and cell culture media (discussed further below). Based on ones overall objective, constraints from fresh cell culture media metabolite concentrations or extracellular metabolomic measurements can either be integrated before or after the extraction algorithm step (Step 20) by using the param.growthMediaBeforeModelExtraction and param.metabolomicsBeforeModelExtraction parameters respectively. If the constraints are added before the extraction, the draft model's solution space will be shrunk as a result of these constraints, whereas if it is done after the extraction, the context-specific model is extracted from a larger solution space with only the reaction directionality, steady-state and perhaps coupling constraints, given a particular network input into the model extraction step. These are important factors to consider when specifying the set of reactions to extract. Setting param.metabolomicsBeforeModelExtraction=true is the default and helps to ensure that the model that enters (and leaves) the extraction step is consistent with metabolomic data. Setting param.metabolomicsBeforeModelExtraction=false usually results in a less constrained model, which can be useful if one wishes to change the exometabolomic constraints,

e.g., to model perturbations. However, care must be taken to ensure that reactions that exchange metabolites corresponding to exometabolomic measurements are still present in the extracted model.

**Cell culture data:** The uptake rates for exchange reactions corresponding to metabolites present in the media are adjusted based on their concentration, modifying only the lower bounds $lb$. The field $\mathrm{mediumConcentrations}$ in $\mathrm{specificData.mediaData}$ can be measured or directly obtained from the product specifications. The function $\mathrm{preprocessingOmicsModel}$, described in the supplementary information, can be used to calculate the maximum medium uptake rates based on the fresh media metabolite concentrations.

**Metabolomics:** The metabolomic constraints are obtained from quantitative metabolomic experiments using biological samples from two time-point measurements. The quality of the data should be assessed carefully prior to the integration with the model by following the general community standards (measurement accuracy, precision, blank effect, linearity of the calibration line, relative errors of calibration line fit etc.) [25]. Furthermore, outlier samples should be identified (for example, based on the median of detected metabolite levels) and excluded. Next, obtained data should be transformed into metabolic rates and normalised for the biomass content, e.g., per gram of dry weight ($gDW$) with the following formula:

$$v_{met} := \frac{(\bar{x_1} - \bar{x_0}) \times f}{\tau \times p}$$

where $v_{met}$ is the rate of exchange of a given metabolite, $\bar{x_0}$ is the mean initial metabolite concentration (e.g., in $\mu mol/L$), $\bar{x_1}$ is the mean metabolite concentration at the second time point ($\mu mol/L$), $\tau$ is the time interval between the two measurements (in hours), $p$ is the protein concentration in the sample ($g/L$), and $f$ is the protein fraction of the total dry weight (g protein/g DW). It should be noted that the biomass content can also be estimated based on the cell count and an estimate of the dry weight mass of a cell or by direct measurements of dry weight mass (tissue).

To propagate the the error associated with exchange flux measurement, one can propagate the standard deviation associated with each measured value using

$$\sigma_{\Delta x} := \sqrt{(\sigma\bar{x_1})^2 + (\sigma\bar{x_0})^2},$$

$$\sigma_v := \frac{|v_{met}| \cdot pf \cdot \sqrt{\left(\frac{\sigma\Delta x}{\Delta x}\right)^2 + \left(\frac{\sigma p}{\bar{p}}\right)^2}}{\tau} \tag{2}$$

where $\sigma_{\Delta x}$ is the standard deviation in the estimation of the change in metabolite concentration between initial ($\bar{x_0}$) and final measurement ($\bar{x_1}$) given the measurement uncertainties, and $\sigma_v$ is the standard deviation in the exchange reaction rate, taking into account the standard deviation in metabolite concentration change estimation ($\sigma_{\Delta x}$ ) and standard deviation in protein estimation ($\sigma_p$) . As the standard deviation in protein estimation is the same for each metabolite, when fitting a model that is linear in the rate of each reaction, one can consider this to be zero.

Measured metabolites should be mapped into the model namespace where metabolite identifiers and/or exchange reaction identifiers for each metabolite should be found (ideally by cross-mapping standard metabolite identifiers such as InChI, ChEBI, HMDB etc.), and included in the final metabolomics input data. The metabolomic constraints are set by fitting the bounds of the draft model to the exometabolomic reaction rates while allowing relaxation of net flux bounds based on the quadratic optimisation problem presented in Preciat et al. [24],

$$
\begin{aligned}
\min_{v,p,q\in\mathbb{R}^n} \quad & (v_{exp}-v)^T \mathrm{diag}(w_{exp})(v_{exp}-v) + p^T \mathrm{diag}(w_l)p + q^T \mathrm{diag}(w_u)q \\
\text{s.t.} \quad & Sv = 0, \\
& Cv \leq d, \\
& \underline{v} - p \leq v \leq \overline{v} + q, \\
& 0 \leq p, \\
& 0 \leq q,
\end{aligned} \tag{3}
$$

where $p \in \mathbb{R}^n_{\geq 0}$ and $q \in \mathbb{R}^n_{\geq 0}$ are non-negative variables that allow the lower ($\underline{v}$) and upper bound ($\overline{v}$) constraints to be relaxed. This problem always returns a steady-state flux $v \in \mathbb{R}^n$ and allows for different information to be input as parameters, including the penalisation of deviation from experimental fluxes ($w_{exp} \in \mathbb{R}^n_{\geq 0}$), penalising relaxation of lower bounds ($w_l \in \mathbb{R}^n_{\geq 0}$), and penalising relaxation of upper bounds ($w_u \in \mathbb{R}^n_{\geq 0}$). The weights for penalising the experimental fluxes are set in param.metabolomicWeights and are derived from the mean ('mean', default), standard deviation ($\sigma_v$ in (2), 'SD'), or relative standard deviation ('RSD') of the experimental reaction flux. For example, param.metabolomicWeights=SD; sets the penalty on deviation from experimental measurement to be the inverse of one plus the variance:

$$
v_{exp} := \frac{1}{(1+\sigma^2_{exp})} \tag{4}
$$

where $\sigma_{exp} \in \mathbb{R}^n_{\geq 0}$ is the standard deviation in measured fluxes. This approach increases the penalty on deviation from an experimentally measured mean flux where the variance is lower.

To avoid numerical errors in model analysis, a default flux value is used if the experimental data exceeds the specified boundary limits: the minimum reaction rate (param.TolMinBoundary), the maximum reaction rate (param.TolMaxBoundary) as well as the absolute minimum flux allowed, as defined in param.boundPrecisionLimit, i.e. any $|lb|$ or $|ub| <$ param.boundPrecisionLimit is considered as zero. If param.printLevel is greater than zero, the new metabolomically derived bounds on reaction fluxes are printed. Finally, the feasibility of the draft model is tested, and if it is not feasible, e.g., if there are inconsistencies or errors in the exometabolomic data, some reaction bounds will be relaxed using relaxedFBA.

### Usable variables:

- specificData.mediaData - Table containing the fresh media concentrations. Contains the reaction identifier, the maximum uptake ($\mu mol/gDW/h$) based on the concentration of the metabolite and the concentration ($\mu mol/l$; Default: empty).

| rxns | mediumMaxUptake | mediumConcentrations |
|---|---|---|
| EX_met1 | -220 | 1345 |
| EX_met2 | -14916 | 191021 |
| : | : | : |

- specificData.exoMet - Table with the fluxes obtained from exometabolomic experiments. It includes the reaction identifier, the reaction name, the measured mean flux, standard deviation of the measured flux, the flux units, and the platform used to measure it.

| rxns | rxnNames | mean | SD | units | platform |
|---|---|---|---|---|---|
| EX_mMet1 | Exchange of mMet1 | 100 | 8x10$^{-4}$ | $\mu mol/gDW/h$ | LC-MS |
| EX_mMet2 | Exchange of mMet2 | 4x10$^{-3}$ | 3x10$^{-4}$ | $\mu mol/gDW/h$ | GC-MS |
| : | : | | | | : |

- param.metabolomicWeights - String indicating the type of weights to be applied for metabolomics fitting (Possible options: 'SD', 'mean' and 'RSD'; Default: 'mean').
- specificData.essentialAA - List exchange reactions of essential amino acids (Default: empty). Must never be secreted, even in a relaxed FBA model.
- param.TolMinBoundary - The reaction flux lower bound minimum value (Default: -1000)

- param.TolMaxBoundary - The reaction flux lower bound maximum value (Default: 1000)
- param.boundPrecisionLimit - Precision of flux estimate, if the absolute value of the lower bound or the upper bound is lower than the *boundPrecisionLimit* but higher than 0 the value will be set to the *boundPrecisionLimit* (Default: primal LP feasibility tolerance).

```
>> param.boundPrecisionLimit = getCobraSolverParams('LP', 'feasTol');
```

- param.growthMediaBeforeReactionRemoval - Logical, should the cell culture data be added before (true) or after (false) the model extraction (Default: true).
- param.metabolomicsBeforeModelExtraction - Logical, should the metabolomic data be added before (true) or after (false) the model extraction (Default: true).

## 11 | Add custom constraints ● TIMING ~ 10 s

**Description:**

Internal and external constraints on reactions can be added using the data in table specificData.rxns2constrain, which will be used to change the lower and upper bounds of the reactions in the draft model. Demand reactions will be ignored to ensure thermodynamic consistency if 'thermoKernel' is chosen as the tissue-specific solver. No change is made if the bound is empty. If the bounds in the table exceed the established limit bounds, they will be adjusted to what the user or default data specifies. If param.printLevel is greater than zero, the number of closed reactions, as well as the number of open demand or sink reactions, will be printed. Finally, the feasibility of the draft model is tested, and if it is not feasible some reactions will be relaxed via relaxedFBA.

**Usable variables:**

- specificData.rxns2constrain - Table containing the reaction identifier, the updated lower bound, the updated upper bound, a constraint description, and any notes such as references or special cases (Default: empty).

| rxns | lb | ub | constrainDescription | Notes |
|---|---|---|---|---|
| rxn1 | 0 | | Close uptake | PMID: **** |
| rxn2 | 0.0.956 | 1000 | Turnover contraints | PMID: **** |
| : | : | : | : | : |

- param.TolMinBoundary - The reaction flux lower bound minimum value (Default: -1000)
- param.TolMaxBoundary - The reaction flux upper bound maximum value (Default: 1000)
- param.boundPrecisionLimit - Precision of flux estimate, if the absolute value of the lower bound or the upper bound is lower than the *boundPrecisionLimit* but higher than 0 the value will be set to the *boundPrecisionLimit* (Default: primal feasibility tolerance).

```
>> param.boundPrecisionLimit = getCobraSolverParams('LP', 'feasTol');
```

- param.tissueSpecificSolver - The name of the tissue-specific solver to be used to extract the context-specific model (Possible options: 'thermoKernel' and 'fastcore'; Default: 'thermoKernel').

## 12 | Set coupled reactions ● TIMING 10 s

**Description:**

Often published biochemical data specifies metabolites that are known to be produced or metabolised by the biological system. However, a model frequently includes multiple production or degradation pathways. To ensure that the net production or consumption of the metabolite is represented correctly in the model the concept of coupled reactions can be used [29]. In non-growing, non-dividing cells, such as neurons, these constraints can be used to replace the biomass reaction to set constraints that specify specific biochemical requirements for cell maintenance [24]. In short, all reactions known to produce ($c = 1$) or degrade ($c = -1$) the metabolite are identified and listed in the specificData.coupledRxns (column coupledRxnsList), and the sum of their net fluxes is set to be greater than ($csence = G$) the known net production/degradation of the metabolite ($d$). This ensures that the specified reactions are active in the model and the total net flux through them

satisfies the set conditions, but the exact flux partitioning between the reactions is not specified. To this end, addCOBRAConstraints is used to add these constraints to the draft model via the inequality $c_1 * v(1) + c_2 * v(2) * ... * c_i * v(i) \geq d$ or $c_1 * v(1) + c_2 * v(2) * ... * c_i * v(i) \leq d$ based on the specification in model.csense. That is, csense ('E', equality, 'G' greater than, 'L' less than. The vector $c \in \{-1, 1\}^i$ contains the coefficients indicating the directionality of each coupled reaction (1 for production, $-1$ for degradation), and $d$ is the constraint's right-hand side $C \cdot v \leq d$ specifying the net value of the production/degradation of the metabolite **? TROUBLESHOOTING** . The coupled constraints are added to the draft model by including fields such as the constraint matrix containing reaction coefficients, the constraint identifiers, the constraint senses, and the constraint right-hand side values. If param.printLevel is greater than zero, the information about the added coupled reactions will be printed.

**Usable variables:**

- param.addCoupledRxns - Logical, should the coupled constraints be added (Default: false). **! CAUTION** If it is TRUE and the table coupledRxns is empty, the step is not performed.
- specificData.coupledRxns - Table containing information about the coupled reactions. This includes the coupled reaction identifier, the list of coupled reactions, the coefficients of those reactions, the constraint, the sense or the directionality of the constraint, and the reference (Default: empty).

| couplingConstraintID | coupledRxnsList | c | d | csence | reference |
|---|---|---|---|---|---|
| cRxn1 | crxn1, crxn2 | 1 1 | 2.0250 | G | PMID: **** |
| cRxn2 | crxn1, crxn3 | 1 -1 | 0.2265 | G | PMID: **** |
| : | : | : | : | : | |

## 13 | Remove inactive reactions ● TIMING 10 s

**Description:**

The metabolic reactions assigned as inactive by manual literature curation are removed from the draft model. In the event of a discrepancy between the datasets, for example, a reaction assigned to be inactive based on manual literature curation but active according to Step 4, preference will be given based on the value of param.curationOverOmics (true - prioritise manual literature curation, false- prioritise experimental data). After reaction removal, the feasibility of the model is tested, and if it fails, some reaction bounds will be relaxed via relaxedFBA. If param.printLevel is greater than zero, the number of reactions removed and the reactions that were assigned as inactive and were removed will be printed.

**Usable variables:**

- specificData.inactiveReactions - List of reactions known to be inactive based on manual literature curation (Default: empty).
- specificData.rxns2remove - List of reactions to remove (Default: empty).
- param.curationOverOmics - Logical, indicates whether curated data should take priority over omics data (Default: false).

## 14 | Remove inactive genes ● TIMING $\sim 10^2$ s

**Description:**

Genes assigned as inactive in bibliomic or transcriptome data are removed from the draft model, together with the reactions affected by their deletion (identified using the function deleteModelGenes). However, any reaction assigned to be active in Step 4 will not be deleted at this step. Furthermore, if removing reactions linked to inactive genes compromises the draft model's feasibility, the bounds of the fewest number of those reactions are relaxed via relaxedFBA , and the rest are removed. To ensure that the model is feasible, the relaxed reactions are added to the list of active reactions. In the event of a discrepancy between the datasets, for example, a gene assigned to be inactive based

on the manual literature curation but active based on the proteomics data, or transcriptiomics data, or both (Step 6), preference will be given based on the value of param.curationOverOmics variable. If param.printLevel is greater than zero, the deleted reactions and genes will be printed, as well as those that were not deleted.

**Usable variables:**

- param.curationOverOmics - Logical, indicates whether curated data should take priority over omics data (Default: false).
- specificData.inactiveGenes - List of Entrez identifiers of genes known to be inactive based on the bibliomics data (Default: empty).

## 15 | Set objective function ● TIMING 1 s

**Description:**

The linear objective $\varphi(v) := c^T v$ used in Flux Balance Analysis [22] can be set in this step (but is not required), where $c \in \mathbb{R}^n$ represents the biologically inspired linear objective to find the optimal flux vector. The objective function in param.setObjective represents the reaction whose flux is to be maximised, such as ATP or biomass production, or minimised, such as energy consumption. However, alternative biological objectives can also be used as shown in Preciat et al. [24].

**Usable variables:**

- param.setObjective - Cell string indicating the linear objective function to optimise (Default: none).

## 16 | Test feasibility ● TIMING $\sim 10$ s

**Description:**

Several constraints have been integrated into the draft model up to this point; in this step, the final feasibility check is performed before the consistency checks. If the draft model is not feasible, some reactions will be relaxed via relaxedFBA.

## 17 | Find flux consistent subset ● TIMING $\sim 10^2$ s

**Description:**

The goal of this step is to find and extract the largest subset of the draft model that is flux consistent, i.e. each metabolite and reaction can carry a non-zero steady-state flux. The options param.fluxCCmethod and param.fluxEpsilon indicate the algorithm that will be used to identify the flux consistent subset of the draft model and to determine the minimum flux magnitude accepted. All reactions that are flux inconsistent, and all metabolites that exclusively participate in flux inconsistent reactions, are removed. If a flux inconsistent metabolite or reaction is identified as active in specificData.presentMetabolites or in the list of active reactions from Step 4, it is removed from the list. If param.printLevel is greater than zero, it prints the dimensions of the stoichiometric matrix before and after checking flux consistency, as well as the constraints for flux consistent reactions and metabolites. In addition, different boolean vectors indicating the flux consistent and inconsistent metabolites and reactions are added to the draft model.

**Usable variables:**

- specificData.presentMetabolites - List of metabolites known to be active based on the bibliomics data (Default: empty).
- param.fluxCCmethod - String with the name of the algorithm to be used for the flux consistency check (Possible options: 'swiftcc', 'fastcc' or 'dc', Default: 'fastcc').
- param.fluxEpsilon - Minimum non-zero flux value accepted for tolerance (Default: Primal feasibility tolerance).

```
>> feasTol = getCobraSolverParams('LP', 'feasTol');
>> param.fluxEpsilon = feasTol * 10;
```

## 18 | Find thermodynamically consistent subset ● TIMING $\sim 10$ s

**Description:**
Any stoichiometric matrix $S$ may be split into one subset of columns corresponding to internal and external reactions, $S = [N, B]$, where internal reactions are stoichiometrically consistent, that is $\exists \ell \in \mathbb{R}^m_{>0}$ such that $N^T \ell = 0$, and external reactions are not stoichiometrically consistent, that is $\nexists \ell \in \mathbb{R}^m_{>0}$ such that $B^T \ell = 0$, as they represent net exchange of mass across the boundary of the system. A steady state flux vector, $v \in \mathbb{R}^r$ such that $Sv = 0$, may be split into an internal reaction flux vector $z \in \mathbb{R}^n$ and an external reaction flux vector $w \in \mathbb{R}^k$, where $Nz + Bw = 0$. In a given internal reaction flux vector $z \in \mathbb{R}^n$, an internal reaction flux $z_j$ is said to to be thermodynamically consistent if there exists a $y \in \mathbb{R}^m$ such that

$$\begin{aligned} z_j > 0 \quad &\Rightarrow \quad N_j^T y < 0, \\ z_j < 0 \quad &\Rightarrow \quad N_j^T y > 0, \\ z_j = 0 \quad &\Rightarrow \quad N_j^T y = 0, \end{aligned}$$

where one may interpret $N_j^T y$ as the thermodynamic driving force for reaction $j$, and $y$ may be interpreted as a vector proportional to the chemical potentials of each metabolite. This is a relaxation of the constraint $\mathrm{sign}(z_j) = -\mathrm{sign}(N_j^T y)$, since $N_j^T y \neq 0 \not\Rightarrow z_j \neq 0$, which is appropriate for a biochemical reaction because, besides the existence of a thermodynamic driving force, the presence of an enzyme may be required for appreciable net flux within a given reaction. The procedure for finding the thermodynamically feasible subset of a given model is described in detail elsewhere (Fleming et al, Cardinality optimisation in constraint-based modelling: Application to human metabolism, 2022, submitted).

This step is only performed if param.tissueSpecificSolver='thermoKernel' is chosen as the extraction algorithm since the performance of 'thermoKernel' is accelerated if it is provided a thermodynamically consistent model prior to extracting a subset of it. An approximation to the largest thermodynamically consistent subset is implemented by 'findThermoConsistentFluxSubset', which requires model.S, model.lb and model.ub as essential inputs. To accelerate this step one may optionally specify the metabolites and reactions that are flux consistent using model.fluxConsistentMetBool and model.fluxConsistentRxnBool.

All reactions that are not thermodynamically consistent are removed from the draft model. All metabolites that are exclusively involved in thermodynamically inconsistent reactions are also removed from the draft model. If a thermodynamically inconsistent metabolite or reaction is identified as active in specificData.presentMetabolites or in the list of active reactions from Step 4, it is removed from the list. Additionally, all orphan genes are removed from the model and the reaction-gene-matrix is regenerated. Boolean vectors indicating thermodynamically consistent metabolites and reactions are added to the draft model, model.thermoFluxConsistentMetBool and model.thermoFluxConsistentRxnBool respectively.

If param.printLevel is greater than zero, it prints the dimensions of the stoichiometric matrix before and after checking thermodynamic consistency, as well as the constraints for thermodynamic consistent reactions and metabolites.

- param.tissueSpecificSolver - The name of the tissue-specific solver to be used to extract the context-specific model (Possible options: 'thermoKernel' and 'fastcore'; Default: 'thermoKernel').
- param.thermoFluxEpsilon - Flux epsilon used in thermoKernel (Default: feasibility tolerance).
- param.iterationMethod - Enables different iteration methods to be employed when exploring the thermodynamically consistent set.

## 19 | Identify active reactions from genes ● TIMING $\sim 10$ s

**Description:**
The list of active reactions up to this point has been based on the context-specific reactions and flux consistency data. The relationship between context-specific genes and metabolic reactions is

established in this step. Each identified reaction is added to the final list of active reactions used to extract the context-specific model. The pipeline determines the gene-reaction relationship using either of the two options in param.activeGenesApproach:

'oneRxnPerActiveGene': At least one reaction per active gene is included (default).

'deleteModelGenes': Find a list of reactions, whose rates are affected by the deletion of active genes and include them all as active reactions.

Changing this parameter has a large effect on the size of the extracted models. Per active gene, 'oneRxnPerActiveGene' adds at least one corresponding reaction to the extracted model, whereas 'deleteModelGenes' adds all corresponding reactions, where the presence of the gene product is essential for nonzero flux through each of the reactions corresponding. Experience with generation of whole-body metabolic models [30] and extraction of a neuronal model [24] from Recon3D supports the use of 'oneRxnPerActiveGene' as the default.

**Usable variables:**

- param.activeGenesApproach - String with the name of the active genes approach will be used (Possible options; 'oneRxnPerActiveGene', 'deleteModelGenes', Default: 'oneRxnPerActiveGene').
- param.printLevel - Level of verbose that should be printed (Default: 0).
- param.debug - Logical, should the function save its progress for debugging (Default: false).

## 20 | Model extraction ● TIMING $\sim 10^2$ s

**Description:**

In this step, given a draft model from step 3.19, a specific model is extracted using the createTissueSpecificModel interface. Two extraction algorithms may be used for this step, as indicated in the param.tissueSpecificSolver: 'fastCore', which extracts a flux consistent context-specific model [32], and 'thermoKernel', the default algorithm, which extracts a stoichiometrically, thermodynamically and flux consistent context-specific model. The 'FastCore' algorithm has been previously described [32]. Below the approach of the 'thermoKernel' is described.

**thermoKernel: Extraction of a thermodynamically consistent context-specific model** Besides defining lists of present metabolites (core metabolites) and active reactions (core reactions), with 'thermoKernel' one also has the option, to set weights on metabolites and reactions. Negative weights promote inclusion of metabolites and reactions in a specific model, positive weights do the opposite, while a zero weight does not penalise for exclusion or inclusion. By default, 'thermoKernel' sets a fixed penalty for inclusion of any metabolite or reaction not in a core set. If gene expression data is provided, this fixed penalty is equal to the median of the log of the reaction expression value model.expressionRxns, otherwise it is one. Importantly, a weight of zero is set for any metabolite ranked in the top 100 for metabolite connectivity. This is to avoid exclusion or inclusion of a reaction if it contains cofactors, and rather focus on determining exclusion or inclusion based on metabolites that participate in a small number of reactions. In addition, if param.activeGenesApproach = oneRxnPerActiveGene, all metabolites and reactions in the core set are set a weight of equal to the *negative* of the median of the log of the reaction expression value model.expressionRxns, or -1 if model.expressionRxns is not provided. If model.expressionRxns is provided, but a reaction is missing gene expression data, it is assumed the corresponding reaction weight of equal to the *negative* of the median of the log of the reaction expression values model.expressionRxns.

We assume that a metabolite is present if it is either produced or consumed at a non-zero rate by a corresponding *net* flux vector, and absent otherwise. Any stoichiometric matrix may be split into $N = R - F$ were $F_{i,j}$ and $R_{i,j}$ are the stoichiometric numbers of the $i^{th}$ molecule consumed and produced in the $j^{th}$ directed reaction, respectively. In terms of forward reaction flux $p \in \mathbb{R}^n_{\geq 0}$ and reverse reaction flux $q \in \mathbb{R}^n_{\geq 0}$, the rate of consumption of each metabolite is $Fp + Rq$ and the rate of

production of each metabolite is $Fq+Rp$. Therefore, we observe that $s := (F+R)p+(F+R)q \geq 0$ represents the sum of the rate of production and consumption of each metabolite. If metabolite $i$ is not present in a network, then $s_i = 0$.

We define thermodynamically consistency as the requirement that any net flux to be driven by a change in chemical potential for the corresponding reaction, that is

$$\begin{aligned} z_j > 0 \quad &\Rightarrow \quad N_j^T y < 0, \\ z_j < 0 \quad &\Rightarrow \quad N_j^T y > 0, \\ z_j = 0 \quad &\Rightarrow \quad N_j^T y = 0, \end{aligned}$$

where $y$ is interpreted as a vector proportional to the chemical potentials of each metabolite. However, the change of chemical potential for a reaction may be non-zero, without a corresponding nonzero flux, representing the absence of an enzyme for the corresponding reaction. This is a relaxation of the constraint $\mathrm{sign}(z_j) = -\mathrm{sign}(N_j^T y)$, since $N_j^T y \neq 0 \not\Rightarrow z_j \neq 0$. To compute a thermodynamically consistent flux that simultaneously quantitatively balances incentives for the presence of metabolites and the activity of reactions, with disincentives for absence of metabolites and inactivity of reactions, 'thermoKernel' implements the following optimisation problem

$$\begin{aligned} \min_{z,w,p,q,s,r} \quad & \beta 1^T (p+q) + g^T \|z\|_0 + h^T \|s\|_0 + f^T \|r\|_0 \\ \text{s.t.} \quad & Nz + Bw = 0, \\ & z = p - q, \\ & (F+R)(p+q) = s, \\ & (F+R)z = r, \\ & \underline{z} \leq z \leq \overline{z}, \\ & \underline{w} \leq w \leq \overline{w}, \\ & 0 \leq p, \\ & 0 \leq q. \end{aligned} \tag{5}$$

The objective term $\beta 1^T (p+q)$ minimises the absolute value of internal reaction flux $z \in \mathbb{R}^n$, which promotes thermodynamic consistency [8]. The scalar weight $\beta > 0$ is chosen to trade off between cardinality optimisation of internal reaction fluxes, which explores the flux consistent subset of the model, and minimising the one norm of the sum of forward and reverse reaction rates, which promotes thermodynamic consistency. The parameter vector $g \in \mathbb{R}^n$ is a vector of internal reaction rates. The term $g^T \|z\|_0$ denotes the weighted zero-norm of internal reaction fluxes, where the zero-norm for each reaction is weighted individually, that is

$$g^T \|z\|_0 := \sum_{j=1}^{n} g_j^T \|z_j\|_0 .$$

When $g_j < 0$, or $g_j > 0$, then net flux is incentivised or penalised, in reaction $j$, respectively.

The parameter vectors $h, f \in \mathbb{R}^m$ are a vectors of metabolite weights. When $h_i > 0$, the objective term $h_i^T \|s_i\|_0$ minimises the zero-norm of the rate of production and consumption of metabolite $i$. When $h_i < 0$, the objective term $h_i^T \|s_i\|_0$ maximises the zero-norm of the rate of production and consumption of metabolite $i$. However, if even if $s_i > 0$ then a metabolite may not be produced or consumed at a non-zero rate by a corresponding *net* flux vector since $p = q \neq 0$ is a solution that also implies $s > 0$. Therefore, we introduce the variable $r := (F+R)z$ as an approximation to the sum of production and consumption due to net reaction flux, which is $(F+R)\,|z|$, but is non-trivial to implement. If $r_j := (F+R)_j z > 0$ then $(F+R)_j\,|z| > 0$ as desired. Therefore, when $f_i < 0$, the objective term $f_j^T \|r_j\|_0$ approximates maximisation of the weighted zero-norm of the sum of production and consumption of metabolite $i$ due to non-zero net reaction flux. In practice, we set $f_j = -\min(h_j, 0)$ so that optimisation of the cardinality of $s_j$ and $r_j$ is concordant.

Note that $\underline{z} \in \{0, -\infty\}^n$ and $\overline{z} \in \{0, \infty\}^n$ denote lower and upper bounds on internal reaction fluxes while $\underline{w}, \overline{w} \in \mathbb{R}^k$ denote lower and upper bounds on external reaction fluxes [8]. Internal

reaction bounds can enforce reaction directionality constraints in a manner consistent with thermodynamic constraints, but bounds that force internal reaction fluxes to be active may result in thermodynamically inconsistent fluxes. Problem 5 computes a thermodynamically feasible flux that simultaneously quantitatively balances incentives for presence of metabolites and activity of reactions, with disincentives for absence of metabolites and inactivity of reactions. This step is also compatible with a requirement that the extracted model be able to optimise a linear combination of net fluxes (not shown), as in flux balance analysis [22]. It is also compatible with coupling constraints [29] that impose linear inequality constraints on reaction rates, $Cv \leq d$, but depending on how they are formulated, such constraints may or may not interfere with implementation thermodynamic flux consistency constraints.

However, it is usually the case that more than one thermodynamically feasible flux is required to ensure activity of a set of desired reactions and presence of a set of metabolites. Therefore, we adopt a randomly greedy strategy inspired by Tefagh and Boyd [26] whereby we choose a random vector from a uniform rectangular set, $d \in \{\mathbb{R}^n | \operatorname{unif}(0,1)\}$, then update weights on cardinality optimisation of reactions in the next iterate $g(n+1)$ based on those of the previous iterate $g(n)$, with the heuristic

$$g_j(n+1) \quad := \quad \begin{cases} g_j(0) \times (n+1) & g_j(n) < 0,\ z_j(n) = 0,\ d \geq 0.5, \\ 0 & g_j(n) < 0,\ z_j(n) = 0,\ d < 0.5, \\ 0 & g_j(n) < 0,\ z_j(n) \neq 0, \\ g_j(0) & g_j(n) > 0, \\ 0 & g_j(n) = 0. \end{cases}$$

This approach successively increases the incentive for activity of internal reactions that are desired to be active, but have not yet been active in a previous iterate. The strategy is the same to incentivise presence of metabolites. The iterations either conclude when all incentivised reactions and metabolites are active and present, respectively, or when a pre-specified maximum number of iterations is reached. Each iteration of Problem 5 is efficiently solved using a sequence of linear optimisation problems, as described in detail elsewhere (Fleming et al, Cardinality optimisation in constraint-based modelling: Application to human metabolism, 2022, submitted).

**Any model extraction algorithm:** If param.printLevel is greater than zero, the size of the stoichiometric matrix, the metabolites and reactions removed by the extraction solver, and the core metabolites and reactions removed by the extraction solver are all printed.

**Usable variables:**

- param.activeGenesApproach - String with the name of the active genes approach will be used (Possible options: 'oneRxnPerActiveGene', 'deleteModelGenes', Default: oneRxnPerActiveGene).
- param.tissueSpecificSolver - The name of the tissue-specific solver to be used to extract the context-specific model (Possible options: 'thermoKernel' and 'fastCore'; Default: 'thermoKernel').
- param.weightsFromOmics - Should gene weights be assigned based on the omics data (Default: 0).
- param.thermoFluxEpsilon - Flux epsilon used in thermoKernel (Default: feasibility tolerance).
- param.fluxEpsilon - Flux consistency tolerance value (Default: primal feasibility tolerance).

```
>> feasTol = getCobraSolverParams('LP', 'feasTol');
>> param.fluxEpsilon = feasTol * 10;
```

## 21 | Final adjustments ● TIMING ~1 s

In the last step, the flux and thermodynamic consistency of the model is checked again. Also specific data used in the extraction of the specific model are added to the model structure. If a field with metabolic reaction formulas did not already exist in the draft model, it is now added (using printRxnFormula function [5]). Also, two vectors are added that indicate which reactions were specified to be active or inactive, as well as which metabolites were specified to be present or absent. Finally, the fields of the context-specific model are reordered into a standard format.

# ANTICIPATED RESULTS

### Preparation of data

1 | The parameters that will be used by the 'XomicsToModel' pipeline are printed, including the default values for options that have not been declared.

### Generic model check

2 | Any modifications made at this step are indicated; for example, if the generic model contains DM_atp_c_ reaction it is re-named to ATPM as it represents ATP **m**aintenance and to ensure that it is not closed together with standard demand reactions for the model extraction with 'thermoKernel'. Furthermore, if the compartment representing mitochondrial intramembrane space ([i]) is removed (if 'thermoKernel' is chosen) the modified reactions are also printed out.

### Add missing reactions

3 | The reactions and formulas added in this step are printed, together with the results of the stoichiometric consistency test. An example of the stoichiometric consistency test for Recon3D [7] can be seen in Table 3

**Table 3:** Recon3D [7] stoichiometric consistency summary

| Summary of stoichiometric consistency | | |
|---|---|---|
| 5835 | 10600 | totals |
| 0 | 1809 | heuristically external |
| 5835 | 8791 | heuristically internal |
| 5835 | 8791 | ... of which are stoichiometrically consistent |
| 0 | 0 | ... of which are stoichiometrically inconsistent. |
| 0 | 0 | ... of which are of unknown consistency. |
| 0 | 0 | heuristically internal and stoichiometrically inconsistent or unknown consistency. |
| 0 | 0 | ... of which are elementally imbalanced (inclusively involved metabolite). |
| 0 | 0 | ... of which are elementally imbalanced (exclusively involved metabolite). |
| 5835 | 8791 | Confirmed stoichiometrically consistent by leak/siphon testing. |

### Identify active genes

6 | A set of active genes was identified using bibliomic, transcriptomic, and proteomic data. In addition, a message is displayed indicating the number of genes that have no expression information due to a lack of transcriptomic data.

### Set limit bounds

5 | A draft model with the updated maximum upper bounds and minimum lower bounds.

### Identify core metabolites and reactions

4 | A set of active reactions identified based on bibliomic, metabolomic, and cell culture data.

### Close ions

7 | A draft model in which the upper and lower bounds of ion exchange reactions such as calcium, potassium, or sodium, among others, are set to zero.

### Close exchange reactions

**8** | A draft model where all the lower bounds of exchange reactions are set to zero to allow only fresh media metabolites to be taken up. A message is shown reporting model statistics such as the stoichiometric matrix dimensions, the number of closed reactions, the number of all exchange reactions, and the number of exchange reactions in the active reactions.

### Close sink and demand reactions

**9** | The sink and demand reactions of the draft model are set to zero and a message is shown reporting the number of closed non-core sink/demand reactions and the method used in param.nonCoreSinksDemands.

### Set metabolic constraints

**10** | A draft model with added cell culture constraints, or exometabolomic constraints, or both. In addition, a message is displayed reporting the number of exometabolomic and media uptake constraints. It includes exometabolomic data statistics, the fitting procedure for each exometabolomic reaction, the relaxed reactions and the summary of the fitting.

### Add custom constraints

**11** | A draft model with custom constraints based on bibliomic data. If 'thermoKernel' is being used, demand reactions included in specificData.rxns2constrain, i.e. with prefix DM_, will be ignored and the message indicating this is being printed. Furthermore, a message is given showing a list of reaction identifiers that cannot be constrained because they are not present in the model, the number of constrained reactions, and whether or not the model was still feasible after the addition of custom constraints.

### Set coupled reactions

**12** | A draft model that includes new fields that correspond to the coupled reactions. A message is shown including the details of the coupled constraints imposed on the draft model.

### Remove inactive reactions

**13** | A draft model, with inactive reactions removed (based on bibliomic data). A message including the number of reactions that were removed from the model, a list of reactions that were not present in the draft model prior to the reaction removal, as well as the results of the model feasibility check.

### Remove inactive genes

**14** | A draft model with inactive genes removed and a detailed description of the number of genes that were removed or missing in draft model prior to this step. Based on the value of param.curationOverOmics a message is given about the genes that were kept or removed from the manually curated set of inactive genes due to the conflict with the provided omics data (measured as active in transcriptomics and/or proteomics). Furthermore, information is given about the number of genes that were not removed since their removal would lead to an infeasible model. Last, model feasibility check results are given, including any reaction that was relaxed if the model was not feasible.

### Set objective function

**15** | A message including the specified linear objective function, or a lack of thereof.

### Test feasibility

**16** | This step generates a draft model with the bounds relaxed if the draft model is not feasible.

### Find flux consistent subset

**17** | A draft model with flux consistent reactions and metabolites is printed, along with a report containing model statistics such as stoichiometric matrix dimensions, flux consistent and inconsistent reactions and metabolites.

### Find thermodynamically consistent subset

**18** | A thermodynamically consistent draft model. Additionally, a message is given with the parameters used to identify the thermodynamically consistent subset. It is followed by a detailed report from the cardinality optimisation. Last, the stoichiometric matrix dimensions, thermodynamically consistent and inconsistent reactions, and metabolites are printed out.

### Identify active reactions from genes

**19** | A set of reactions that must be active based on the active genes.

### Model extraction

**20** | An extracted genome-scale metabolic model. If 'fastCore' is selected as the extraction algorithm a message is printed out with the extraction summary and the size of the new model. If 'thermoKernel' is selected as the extraction algorithm, a message is printed out with the parameters used in the 'thermoKernel' function, as well as 'optCardThermo' and 'optimizeCardinality' functions called by the 'thermoKernel' function, an extraction summary, and statistics for the new model.

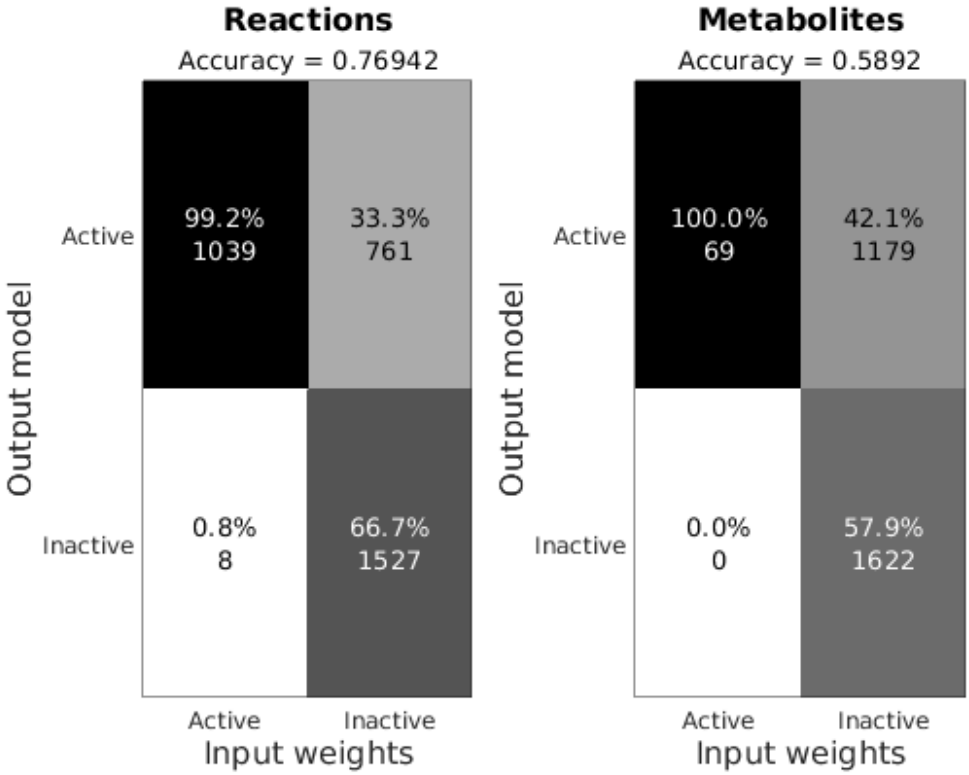



**Figure 4:** Comparison of specified and actual metabolites and reactions in an extracted model. Confusion matrices between specified input weights for reactions and metabolites supposed to be active and inactive versus output model reactions and metabolites that are actually active and inactive, obtained while generating a context-specific model of a dopaminergic neuronal metabolism [24]. The end result may be a trade-off between incentives to be active (negative weights), incentives to be inactive (positive weights) and ambivalence (zero weights). When a reaction is incentivised to be present but the corresponding metabolites are incentivised to be inactive, the optimal solution at each iteration of 'thermoKernel' will depend on the relative weights on the reactions versus metabolites. This plot is automatically generated by 'XomicsToModel' param.plotThermoKernelStats = 1;' when param.tissueSpecificSolver = 'thermoKernel'.

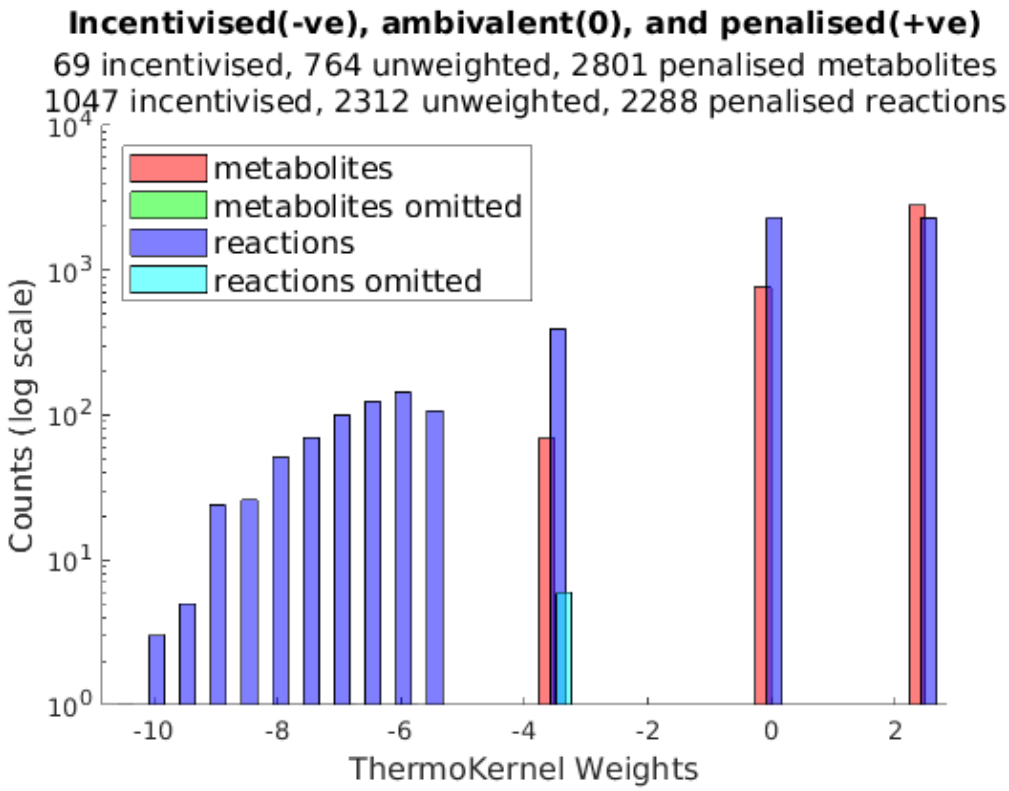



**Figure 5:** Comparison of weights on active and inactive reactions and metabolites

Almost all reactions incentivised to be active (negative weights) are active, with a small proportion inactive that should be active (reactions omitted, light blue), obtained while generating a context-specific model of a dopaminergic neuronal metabolism [24]. All metabolites incentivised to be active (negative weights) are active, with a none inactive that should be active (metabolites omitted, green in legend, no bar in plot). This plot is automatically generated by 'XomicsToModel' if param.plotThermoKernelWeights = 1;' when param.tissueSpecificSolver = 'thermoKernel'.

### Final

21 | The final context-specific flux consistent genome-scale metabolic model. If 'thermoKernel' was chosen, then the model is also thermodynamically consistent.

## ? TROUBLESHOOTING

| Step | Problem | Possible reason | Solution |
|---|---|---|---|
| All | Unexpected behaviour within a key step. | Erroneous or inconsistent input data. Exceptional use case for XomicsToModel not observed previously. | Set param.debug = true;and run XomicsToModel again. All results generated prior to a particular step are saved in a MATLAB workspace file (.mat) corresponding to that step. Therefore, in the case of unexpected results, rather than running the entire pipeline again, one can load the MATLAB workspace file at the desired step then restart XomicsToModel in MATLAB debugging mode as required. |
| 2 | Infeasible draft input model. | The draft model may be infeasible for a variety of reasons, including flux or stoichiometric inconsistency, or because its bounds are over-constrained. | The XomicsToModel pipeline aims to generate a feasible stoichiometrically-flux consistent model; if this is not accomplished, either the model or the included information needs to be modified.<br>The fastGapFill function can add missing reactions to make the draft model feasible. Reactions can be added using the addReaction function after additional manual literature curation.<br>If it is over-constrained, relaxedFBA relaxes the minimum number of reaction bounds in the draft model to make it feasible. |
| 3 | RXN1 in specificData. rxns2add.rxns requires a reaction formula in specificData.rxns2add. rxnFormulas | New metabolic reactions in the draft model require the addition of the metabolic reaction formula. | Place the metabolic reaction formula in the specificData.rxns2add.rxnFormulas for each reaction in the following format:<br>'met1 + 2 met2 -> met3 + met4'<br>where the type of reaction is indicated with the proper characters (Forward ($->$); Reverse ($<-$); Bidirectional ($<=>$)) |
| 11 | Lower bounds greater than upper bounds | A lower bound of a reaction cannot be greater than an upper bound $lb \not> ub$. | Check the bounds included in the options variable including:<br>specificData.rxns2add,<br>param.TolMinBoundary,<br>param.TolMaxBoundary,<br>specificData.rxns2constrain and<br>specificData.exoMet described in Sections 3, 5, 10 and 10. |
| 12 | Missing fields in the table. | To add the coupled reactions, no element of the table specificData. coupledRxns can be left empty. | Fill in the empty cells with the missing information as described in the Section 12. |

| Step | Problem | Possible reason | Solution |
|---|---|---|---|
| 20 | Extracted model contains reactions that do not admit a thermodynamically feasible net flux. | The objective term $\beta 1^T(p+q)$, which minimises the absolute value of internal reaction flux $z \in \mathbb{R}^n$, which promotes thermodynamic consistency, is too small. | Increase the value of the trade-off parameter ( model.beta) on minimisation of one-norm of internal fluxes. That is, the value of $\beta$ should be such that $\beta 1^T(p+q) \geq g^T \|z\|_0 + h^T \|s\|_0 + f^T \|r\|_0$at the optimum of each iteration within the thermoKernel algorithm. |
| 20 | Extracted model does not contain the anticipated metabolites or reactions | Anticipated metabolites or reactions are not part of the flux (or thermodynamically flux consistent subset). Alternatively, weighing on metabolites or reactions too small relative to $\beta$. | Check whether anticipated metabolites or reactions were eliminated prior to the model extraction step. Next check if increasing the magnitude of the (negative) weights on metabolites or reactions increases the fraction of anticipated metabolites included in the extracted model. Check to make sure that specificData.rxns2constrain contains bounds of sufficient magnitude, i.e., at least 100 times the feasibility tolerance of the solver, so typically of magnitude 1e-4 or greater. |
| 20 | Extracted model does not admit sufficient flux through a specified biomass reaction. | | Ensure param.biomassRxn is set to the correct reaction abbreviation. Add a sufficient lower bound on biomass reaction flux as an additional constraint that is compatible with a steady state flux in the input generic model. specificData.rxns2constrain |

## ● TIMING

Step 1, Preparation of data: $\sim 5$ s
Step 2, Generic model check: $\sim 1$ s
Step 3, Add missing reactions: $\sim 30$ s
Step 5, Set limit bounds: $\sim 1$ s
Step 6, Identify active genes: $\sim 10^2$ s
Step 4, Identify core metabolites and reactions: $\sim 10$ s
Step 7, Close ions: $\sim 1$ s
Step 8, Close exchange reactions: $\sim 1$ s
Step 9, Close sink and demand reactions: $\sim 10$ s
Step 10, Set metabolic constraints: $\sim 10$ s
Step 11, Add custom constraints: $\sim 10$ s
Step 12, Set coupled reactions: $10$ s
Step 13, Remove inactive reactions: $10$ s
Step 14, Remove inactive genes: $\sim 10^2$ s
Step 15, Set objective function: $1$ s
Step 16, Test feasibility: $\sim 10$ s
Step 17, Find flux consistent subset: $\sim 10^2$ s
Step 18, Find thermodynamically consistent subset: $\sim 10$ s
Step 19, Identify active reactions from genes: $\sim 10$ s
Step 20, Model extraction: $\sim 10^2$ s
Step 21, Final: $\sim 1$ s

---

**ACKNOWLEDGEMENTS** The authors would like to thank Hanneke Leegwater for her helpful feedback while revising the manuscript. GP, AW, TH and RF received funding from the European Union's Horizon 2020 research and innovation programme, for the SysMedPD project, under grant agreement No. 668738, and from the Dutch National Institutes of Health (ZonMw) TKI-LSH Neuromet project (LSHM18092). AW, TH and RF received funding from Dutch Research Council for the The Netherlands X-Omics Initiative, project 184.034.019. IT received funding from the European Research Council under the European Unions Horizon 2020 research and innovation programme (grant agreement No 757922), and by the National Institute on Aging grants (1RF1AG058942-01 and 1U19AG063744-01). RF received funding from the U.S. Department of Energy, Office of Science, Biological and Environmental Research Program, award DE-SC0010429.

**AUTHOR CONTRIBUTIONS** CRediT (Contributor Roles Taxonomy) author statement

| Author | Contributions |
|---|---|
| German Preciat | Conceptualisation; Methodology; Software; Tutorial; Writing - Original draft. |
| Agnieszka Wegrzyn | Conceptualisation; Methodology; Software; Data Curation Writing - review and editing; |
| Ines Thiele | Conceptualisation; Methodology; Software; Funding acquisition |
| Thomas Hankemeier | Resources; Supervision; Funding acquisition |
| Ronan M.T. Fleming | Conceptualisation; Methodology; Software; Writing - Original Draft; Writing - review and editing; Supervision; Funding acquisition |

**COMPETING FINANCIAL INTERESTS** The authors declare that they have no competing financial interests.

## References


[1] Rasmus Agren et al. "Reconstruction of Genome-Scale Active Metabolic Networks for 69 Human Cell Types and 16 Cancer Types Using INIT". In: *PLOS Computational Biology* 8.5 (May 2012), e1002518. DOI: 10.1371/journal.pcbi.1002518.

[2] R. Balestrino and A. H. V. Schapira. "Parkinson Disease". In: *European Journal of Neurology* 27.1 (Jan. 2020), pp. 27–42. DOI: 10.1111/ene.14108.

[3] D. A. Beard, S.-D. Liang and H. Qian. "Energy Balance for Analysis of Complex Metabolic Networks." In: *Biophysical Journal* 83.1 (2002), pp. 79–86. DOI: 10.1016/S0006-3495(02)75150-3.

[4] Scott A. Becker and Bernhard O. Palsson. "Context-Specific Metabolic Networks Are Consistent with Experiments". In: *PLOS Computational Biology* 4.5 (May 2008), e1000082. DOI: 10.1371/journal.pcbi.1000082.

[5] Scott A. Becker et al. "Quantitative Prediction of Cellular Metabolism with Constraint-Based Models: The COBRA Toolbox". In: *Nature Protocols* 2.3 (Mar. 2007), pp. 727–738. DOI: 10.1038/nprot.2007.99.

[6] Stephen P Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge, UK; New York: Cambridge University Press, 2004.

[7] Elizabeth Brunk et al. "Recon3D Enables a Three-Dimensional View of Gene Variation in Human Metabolism". In: *Nature Biotechnology* 36 (Feb. 2018), p. 272.

[8] Abdelmoneim Amer Desouki et al. "CycleFreeFlux: Efficient Removal of Thermodynamically Infeasible Loops from Flux Distributions". In: *Bioinformatics* 31.13 (July 2015), pp. 2159–2165. DOI: 10.1093/bioinformatics/btv096.

[9] Dill, Ken and Bromberg, Sarina. *Molecular Driving Forces Statistical Thermodynamics in Biology, Chemistry, Physics, and Nanoscience*. Garland Science, 2011.

[10] R. M. T. Fleming et al. "A Variational Principle for Computing Nonequilibrium Fluxes and Potentials in Genome-Scale Biochemical Networks." In: *Journal of Theoretical Biology* 292 (2011), pp. 71–77. DOI: 10.1016/j.jtbi.2011.09.029.

[11] Albert Gevorgyan, Mark G. Poolman and David A. Fell. "Detection of Stoichiometric Inconsistencies in Biomolecular Models." In: *Bioinformatics* 24.19 (2008), pp. 2245–51. DOI: 10.1093/bioinformatics/btn425.

[12] Steinn Gudmundsson and Ines Thiele. "Computationally Efficient Flux Variability Analysis". In: *BMC Bioinformatics* 11.1 (Sept. 2010), p. 489. DOI: 10.1186/1471-2105-11-489.

[13] Hulda S. Haraldsdóttir et al. "CHRR: Coordinate Hit-and-Run with Rounding for Uniform Sampling of Constraint-Based Models". In: *Bioinformatics* 33.11 (Jan. 2017), pp. 1741–1743. DOI: 10.1093/bioinformatics/btx052.

[14] Laurent Heirendt et al. "Creation and Analysis of Biochemical Constraint-Based Models Using the COBRA Toolbox v.3.0". In: *Nature Protocols* 14.3 (Mar. 2019), p. 639. DOI: 10.1038/s41596-018-0098-2.

[15] Christopher S. Henry, Linda J. Broadbelt and Vassily Hatzimanikatis. "Thermodynamics-Based Metabolic Flux Analysis". In: *Biophysical Journal* 92.5 (Mar. 2007), pp. 1792–1805. DOI: 10.1529/biophysj.106.093138.

[16] Livnat Jerby, Tomer Shlomi and Eytan Ruppin. "Computational Reconstruction of Tissue-Specific Metabolic Models: Application to Human Liver Metabolism". In: *Molecular Systems Biology* 6.1 (Jan. 2010). DOI: 10.1038/msb.2010.56.

[17] Daniel Machado et al. "Fast Automated Reconstruction of Genome-Scale Metabolic Models for Microbial Species and Communities". In: *Nucleic Acids Research* 46.15 (Sept. 2018), pp. 7542–7553. DOI: 10.1093/nar/gky537.

[18] Gregory L. Medlock, Thomas J. Moutinho and Jason A. Papin. "Medusa: Software to Build and Analyze Ensembles of Genome-Scale Metabolic Network Reconstructions". In: *PLOS Computational Biology* 16.4 (Apr. 2020), e1007847. DOI: 10.1371/journal.pcbi.1007847.

[19] Alberto Noronha et al. "The Virtual Metabolic Human Database: Integrating Human and Gut Microbiome Metabolism with Nutrition and Disease". In: *Nucleic Acids Research* 47.D1 (Jan. 2019), pp. D614–D624. DOI: 10.1093/nar/gky992.

[20] Charles J. Norsigian et al. "A Workflow for Generating Multi-Strain Genome-Scale Metabolic Models of Prokaryotes". In: *Nature Protocols* 15.1 (Jan. 2020), pp. 1–14. DOI: 10.1038/s41596-019-0254-3.

[21] Sjoerd Opdam et al. "A Systematic Evaluation of Methods for Tailoring Genome-Scale Metabolic Models". In: *Cell Systems* 4.3 (Mar. 2017), 318–329.e6. DOI: 10.1016/j.cels.2017.01.010.

[22] Jeffrey D. Orth, Ines Thiele and Bernhard Ø Palsson. "What Is Flux Balance Analysis?" In: *Nature Biotechnology* 28.3 (Mar. 2010), pp. 245–248. DOI: 10.1038/nbt.1614.

[23] Bernhard Ø Palsson. *Systems Biology: Constraint-based Reconstruction and Analysis*. Cambridge, England: Cambridge University Press, Jan. 2015.

[24] German Preciat et al. *Mechanistic Model-Driven Exometabolomic Characterisation of Human Dopaminergic Neuronal Metabolism*. July 2021. DOI: 10.1101/2021.06.30.450562.

[25] Vinod P. Shah et al. "Bioanalytical Method Validation–a Revisit with a Decade of Progress". In: *Pharmaceutical Research* 17.12 (2000), pp. 1551–1557. DOI: 10.1023/A:1007669411738.

[26] Mojtaba Tefagh and Stephen P. Boyd. "SWIFTCORE: A Tool for the Context-Specific Reconstruction of Genome-Scale Metabolic Networks". In: *BMC Bioinformatics* 21.1 (Apr. 2020), p. 140. DOI: 10.1186/s12859-020-3440-y.

[27] I. Thiele and B. Ø. Palsson. "A Protocol for Generating a High-Quality Genome-Scale Metabolic Reconstruction." In: *Nature Protocolss* 5.1 (2010), pp. 93–121. DOI: 10.1038/nprot.2009.203.

[28] Ines Thiele, Nikos Vlassis and Ronan M. T. Fleming. "fastGapFill: Efficient Gap Filling in Metabolic Networks". In: *Bioinformatics (Oxford, England)* 30.17 (Sept. 2014), pp. 2529–2531. DOI: 10.1093/bioinformatics/btu321.

[29] Ines Thiele et al. "Functional Characterization of Alternate Optimal Solutions of Escherichia Coli's Transcriptional and Translational Machinery". In: *Biophysical Journal* 98.10 (May 2010), pp. 2072–2081. DOI: 10.1016/j.bpj.2010.01.060.

[30] Ines Thiele et al. "Personalized Whole-Body Models Integrate Metabolism, Physiology, and the Gut Microbiome". In: *Molecular Systems Biology* 16.e8982 (2020), p. 24.

[31] Kalyan C. Vinnakota et al. "Network Modeling of Liver Metabolism to Predict Plasma Metabolite Changes During Short-Term Fasting in the Laboratory Rat". In: *Frontiers in Physiology* 10 (2019), p. 161. DOI: 10.3389/fphys.2019.00161.

[32] Nikos Vlassis, Maria Pires Pacheco and Thomas Sauter. "Fast Reconstruction of Compact Context-Specific Metabolic Network Models". In: *PLoS Comput Biol* 10.1 (Jan. 2014), e1003424. DOI: 10.1371/journal.pcbi.1003424.

[33] Hao Wang et al. "Genome-Scale Metabolic Network Reconstruction of Model Animals as a Platform for Translational Research". In: *Proceedings of the National Academy of Sciences* 118.30 (July 2021). DOI: 10.1073/pnas.2102344118.

[34] Yuliang Wang, James A. Eddy and Nathan D. Price. "Reconstruction of Genome-Scale Metabolic Models for 126 Human Tissues Using mCADRE". In: *BMC Systems Biology* 6.1 (Dec. 2012), p. 153. DOI: 10.1186/1752-0509-6-153.

[35] Hadas Zur, Eytan Ruppin and Tomer Shlomi. "iMAT: An Integrative Metabolic Analysis Tool". In: *Bioinformatics* 26.24 (Dec. 2010), pp. 3140–3142. DOI: 10.1093/bioinformatics/btq602.

## Supplementary information

### Supplementary Manual 1 - Additional Tools

A brief explanation of how to use the additional tools for the XomicsToModel pipeline.

### Supplementary Manual 2 - XomicsToModel tutorial

A html version of a tutorial demonstrating the use of the XomicsToModel pipeline. Tutorial available as an executable MATLAB live-script from:
https://github.com/opencobra/COBRA.tutorials/dataIntegration/XomicsToModel.

# Supplementary Information 1 - Additional tools

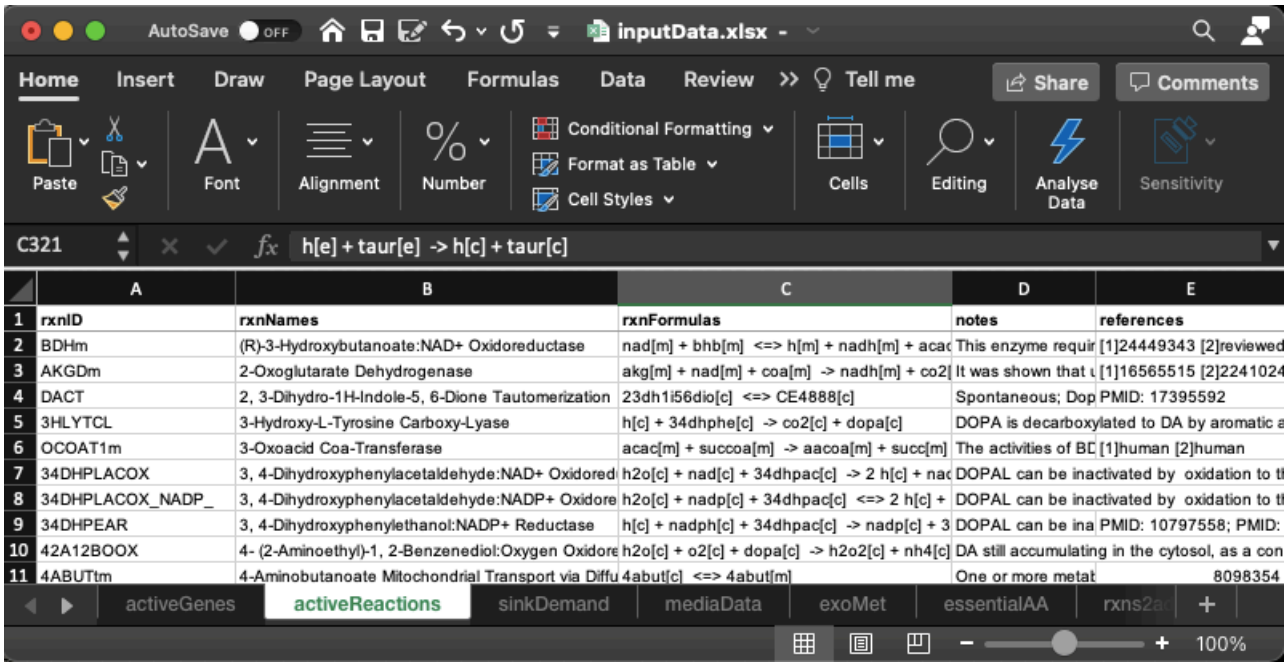
## Preprocessing Omics Data

**Description:**

In order to facilitate data integration, the function preprocessingOmicsModel prepares the context-specific and technical information that will be used by the XomicsToModel pipeline. It is used by running:

```
>> specificData = preprocessingOmicsModel(inputData, setMinActiveFlux, ...
    setMaxActiveFlux, specificData)
```

The variable inputData is a string describing the location of the file (path) and the file name for the context-specific data. Each sheet in the table is added as a field in the specificData variable and, if necessary, converted to the corresponding format. ▲ **CRITICAL STEP** The names of the sheets must be identical to the fields described in each step and as shown in Figure S1, otherwise, the data will be ignored by the XomicsToModel pipeline.




| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | rxnID | rxnNames | rxnFormulas | notes | references |
| 2 | BDHm | (R)-3-Hydroxybutanoate:NAD+ Oxidoreductase | nad[m] + bhb[m] <=> h[m] + nadh[m] + acac | This enzyme requir | [1]24449343 [2]reviewed |
| 3 | AKGDm | 2-Oxoglutarate Dehydrogenase | akg[m] + nad[m] + coa[m] -> nadh[m] + co2[ | It was shown that u | [1]16565515 [2]2241024 |
| 4 | DACT | 2, 3-Dihydro-1H-Indole-5, 6-Dione Tautomerization | 23dh1i56dio[c] <=> CE4888[c] | Spontaneous; Dop | PMID: 17395592 |
| 5 | 3HLYTCL | 3-Hydroxy-L-Tyrosine Carboxy-Lyase | h[c] + 34dhphe[c] -> co2[c] + dopa[c] | DOPA is decarboxylated to DA by aromatic a | |
| 6 | OCOAT1m | 3-Oxoacid Coa-Transferase | acac[m] + succoa[m] -> aacoa[m] + succ[m] | The activities of BD | [1]human [2]human |
| 7 | 34DHPLACOX | 3, 4-Dihydroxyphenylacetaldehyde:NAD+ Oxidored | h2o[c] + nad[c] + 34dhpac[c] -> 2 h[c] + nac | DOPAL can be inactivated by oxidation to tl | |
| 8 | 34DHPLACOX_NADP_ | 3, 4-Dihydroxyphenylacetaldehyde:NADP+ Oxidore | h2o[c] + nadp[c] + 34dhpac[c] <=> 2 h[c] + | DOPAL can be inactivated by oxidation to tl | |
| 9 | 34DHPEAR | 3, 4-Dihydroxyphenylethanol:NADP+ Reductase | h[c] + nadph[c] + 34dhpac[c] -> nadp[c] + 3 | DOPAL can be ina | PMID: 10797558; PMID: |
| 10 | 42A12BOOX | 4- (2-Aminoethyl)-1, 2-Benzenediol:Oxygen Oxidore | h2o[c] + o2[c] + dopa[c] -> h2o2[c] + nh4[c] | DA still accumulating in the cytosol, as a con | |
| 11 | 4ABUTtm | 4-Aminobutanoate Mitochondrial Transport via Diffu | 4abut[c] <=> 4abut[m] | One or more metat | 8098354 |

**Figure S1:** An example of the inputData file.

If Force activity is indicated in the column specificData.rxns2constrain.constrainDescription, but no $lb$ or $ub$ values are provided, the parameters setMinActiveFlux and setMaxActiveFlux are used to make the flux of a reaction $n_j$ to be $v_j > 0$ (reaction will be forced to be active). The lower bound of the reaction is then set to the feasibility tolerance multiplied by 100, while the upper bound is changed to the maximum feasibility tolerance (1000, or param.TolMaxBoundary if is present). If $lb$ or $ub$ values are provided in the data file they will not be changed.

Furthermore, the media uptakes can also be calculated in this step based on the initial metabolite concentration in the media (defined either by the manufacturer or from measurements). It is necessary to have the cell culture information provided in table specificData.cellCultureData for this step. The following formula shows how the uptakes are calculated, and which variables need to be included

$$\frac{metaboliteConcentration(umol/L) \times uptakeSign \times volume(L) \times proteinFraction(g/gDW)}{interval(hr) \times averageProteinConcentration(g/L) \times assayVolume(L)}$$

Lastly, it is also possible to include an existing version of the specificData variable and then preprocessingOmicsModel updates it.

### Used variables:

- specificData.rxns2constrain - Table containing the reaction identifier, the updated lower bound, the updated upper bound, a description for the constraint and notes such as references or special cases (Default: empty).

| rxns | lb | ub | constraintDescription | Notes |
|---|---|---|---|---|
| rxn1 | 0.001 | | Force activity | PMID: **** |

- specificData.cellCultureData - Table containing the cell culture data used to calculate the uptake flux. Includes well volume ($mL$), time interval cultured ($hrs$), average protein concentration ($g/L$), assay volume ($L$), protein fraction (dimensionless), and the sign for uptakes (Default: empty).

| volume | interval | averageProteinConcentration | assayVolume | proteinFraction | uptakeSign |
|---|---|---|---|---|---|
| 0.002 | 48 | 0.3989 | 0.00045 | 0.706 | -1 |

- param.TolMaxBoundary - The reaction flux upper bound maximum value (Default: 1000).

## XomicsToMultipleModels

### Description:

Taking advantage of the flexibility of the XomicsToModel pipeline, the XomicsToMultipleModels function can be used to generate an ensemble of context-specific genome-scale models by varying the options used in the function. The function is used to represent experiments with varying time points or measurement platforms, different cases (i.e. mutant vs WT), use different generic models, or identify the conditions that are most similar to a training set. . For each model, a directory with the names of the different parameters will be created in which the diary and/or variables from debugging can be saved. The parameters include different optimization solvers, transcriptomic threshold or active genes approach.

To run the XomicsToMultipleModels function the command used is:

```
>> directories = XomicsToMultipleModels(modelGenerationConditions, param, ...
    replaceModels)
```

### Used variables:

- modelGenerationConditions.activeGenesApproach - The different approached described in Step 19 (Possible options: 'deleteModelGenes' and 'oneRxnPerActiveGene'; default: 'oneRxnperActiveGene');
- modelGenerationConditions.boundstoRelaxExoMet - The type of bounds that can be relaxed, upper bounds, lower bounds or both ('b'; possible options: 'u', 'l' and 'b'; default: 'b');
- modelGenerationConditions.closeIons - Indicate whether the ions are open or closed (Possible options: true and false; default: false);
- modelGenerationConditions.cobraSolver - Optimisation solvers supported by the pipeline. Possible options: 'glpk', 'gurobi', 'ibm_cplex', 'matlab'; default: 'gurobi');
- modelGenerationConditions.genericModel - Generic COBRA model(s);
- modelGenerationConditions.inactiveGenesTranscriptomics - Use inactive transcriptomic genes or not (Possible options: true and false; default: false);
- modelGenerationConditions.specificData - Specific data to be used (Default: empty);
- modelGenerationConditions.limitBounds - Boundary on the model (Default: 1000);
- modelGenerationConditions.tissueSpecificSolver - Extraction solver (Possible options: 'fastCore'[32] and 'thermoKernel'; default: 'thermoKernel');
- modelGenerationConditions.outputDir - Directory where the models will be generated (Default: current directory);
- modelGenerationConditions.transcriptomicThreshold - Transcriptomic thresholds that are defined by the user (Default: $log2(2)$);
- param - Variable with fixed parameters (Default: empty structure array);
- replaceModels - Logical, It is used to determine whether or not the models of an existing directory should be replaced (Default: false).

## modelPredictiveCapacity

### Description:

The function modelPredictiveCapacity can be used to verify a model's predictive capacity when evaluating one or more models. Three tests can be performed by the function: flux consistency, thermodynamic flux consistency, and predictions against a training set.

During flux consistency and thermodynamic flux consistency tests, it is determined whether active or inactive metabolites and reactions have a consistent flux. On the other hand, the predictive capacity of the model to qualitatively and quantitatively predict fluxes is determined by comparing it to a training data set with constraints for different reactions. The function modelPredictiveCapacity relaxes the model's upper and lower bounds for the reactions in the training set, then the objective functions supported by the function are used to predict the directionality of the fluxes as well as their euclidean distance and Spearman correlation to the training set.

modelPredictiveCapacity returns two variables; the variable comparisonData contains detailed information about the comparisons between predicted fluxes and the training dataset. It displays a table containing the consistency of metabolites and reactions, the predicted and measured fluxes (training dataset), and the final comparison. Furthermore, the variable summary contains a summary of the results.

The data used for Figure **??** was obtained using modelPredictiveCapacity.

```
>> [comparisonData, summary] = modelPredictiveCapacity(model, param)
```

### Used variables:

- model - A Cobra model to be tested;
- param.tests - Array with the tests run on the model (Possible options: 'fluxConsistent': Flux consistency; 'thermoConsistentFlux': Thermodynamic flux consistency; 'flux': Objective function comparison based on a training set; 'all': Do all tests; Default: 'all').;
- param.activeInactiveRxn - $nx1$ with entries $\{1, -1, 0\}$ depending on whether a reaction must be active, inactive, or unspecified respectively;
- param.presentAbsentMet - $nx1$ with entries $\{1, -1, 0\}$ depending on whether a metabolite must be present, absent, or unspecified respectively;
- param.trainingSet - Table with the training set. It includes the reaction identifier, the reaction name, the measured mean flux and standard deviation of the flux (Required for param.tests = 'flux').

| rxns | rxnNames | mean | SD |
|---|---|---|---|
| EX_mMet1 | Exchange of mMet1 | 100 | 8x10$^{-4}$ |
| EX_mMet2 | Exchange of mMet2 | 4x10$^{-3}$ | 3x10$^{-4}$ |
| : | : | | |

- param.objectives - List of objective functions to be tested (Required for param.tests = 'flux'; Default: 'all').

## debugXomicsToModel

### Description:

The function debugXomicsToModel can be used to examine the debug files, optionally generated when param.debug =1;, by the XomicsToModel function, to determine when metabolites, reactions, and genes of interest are removed or retained during the XomicsToModel generation process.

```
>> debugXomicsToModel(genericModel, directory, modelGenerationReport, coreData)
```

### Used variables:

- genericModel - Generic COBRA model used as input to the XomicsToModel function.
- directory - directory where the intermediate results were saved when param.debug =1;.
- modelGenerationReport - report generated by XomicsToModel. Alternatively, provide the coreData structure defined below to focus on a subset of genes, reactions, metabolites or combinations thereof.
- coreData.genes - cell array of Entrez gene identifiers of interest.
- coreData.rxns - cell array of reaction identifiers of interest.
- coreData.mets - cell array of metabolite identifiers of interest.