XomicsToModel: Multiomics data integration and generation of thermodynamically consistent metabolic models

8th November 2021

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

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 multi-omic 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, multi-omics, data integration, thermodynamic consistency, systems biology.

[1]Division of Systems Biomedicine and Pharmacology, Leiden Academic Centre for Drug Research, Leiden University, Einsteinweg, Leiden, The Netherlands,

[2]School of Medicine, National University of Ireland, University Road, Galway, Ireland,

[3]Ryan Institute, National University of Galway, Galway, Ireland,

[4]Division of Microbiology, National University of Galway, Galway, Ireland,

[5]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 [28]. 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 [34] species have been proposed. In the third instance, given a generic metabolic reconstruction, or model, for a multi-cellular organism containing metabolic reactions from all anatomical regions, there are several methods for deriving a model that is specific for a particular anatomical region. In literature, they are also referred to as methods for tailoring a genome-scale metabolic model [21]. 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 exmple, 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 [32]. 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 reliably quantified if the set is convex [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

$$\begin{aligned} \max_{v\in\mathbb{R}^n} \quad & \rho(v) \coloneqq c^T v \\ \text{s.t.} \quad & Sv = 0, \\ & lb \leq v \leq ub, \end{aligned} \tag{1}$$

where $v \in \mathbb{R}^n$ is the net rate of each reaction, $\rho(v) \coloneqq c^T v$ is a biologically motivated linear objective, specified by the coefficient vector $c \in \mathbb{R}^n$. The matrix $S \in \mathbb{R}^{m\times n}$ is a stoichiometric matrix, where m is the number of metabolites, and n is the number of reactions. The constraint $Sv = 0$ represents the assumption of steady-state, that is production $=$ consumption for metabolites not exchanged across the boundary of the system and production $+$ uptake $=$ consumption $+$ secretion for metabolites exchanged across the boundary of the system. The inequalities $lb \leq v \leq ub$ denote box constraints from the lower ($lb \in \mathbb{R}^n$) and upper bounds ($ub \in \mathbb{R}^n$) 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 $\Omega \coloneqq \{Sv = 0,\ lb \leq v \leq ub\}$. 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 [29].

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 represents internal reactions, which do conserve mass, and the matrix B represents external reactions, which do not conserve mass, e.g., reaction equation $A \rightarrow \varnothing$. 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.

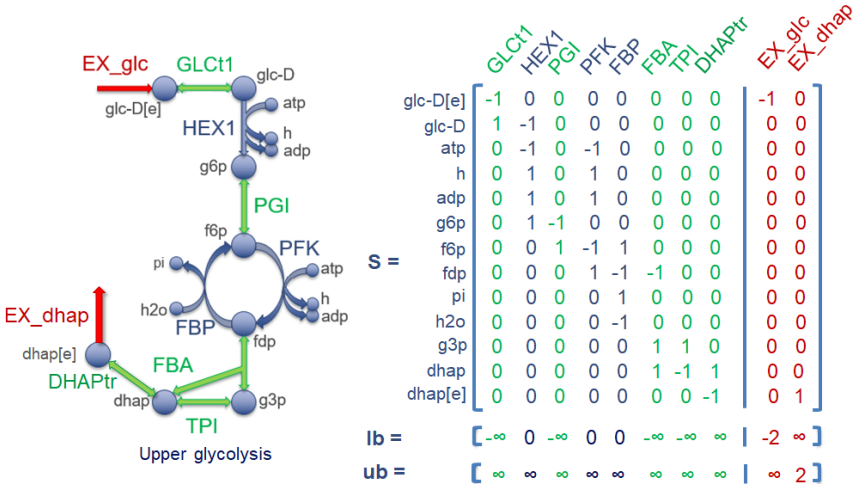


Figure 1: Network diagram of upper glycolysis (left) with reactions (arrows, upper case labels) and metabolites (nodes, lower case labels) with corresponding labels. The corresponding Stoichiometric matrix (right, $S \in \mathbb{R}^{13\times10}$, $N \in \mathbb{R}^{13\times8}$, $B \in \mathbb{R}^{13\times2}$), 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 $A \rightarrow B \rightarrow C \rightarrow A$ 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 μ_A denote the chemical potential of metabolite A. Given net flux $A \rightarrow B$ 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 [36], MBA [16], mCADRE [35], FastCore [33], and CarveMe [17], and also gap filling algorithms [29], 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 multi-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 multi-omic 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 multi-omic 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 [30] 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 [33]. 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[33], 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 multi-omics data, for model construction and validation, as well as careful selection of gene expression cut-offs, lead to higher model accuracy. 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]	[36]	[16]	[1]	[35]	[33]	N/A
Year	2008	2010	2010	2012	2012	2014	2021
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 tutorial. 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 the activity of reactions associated with the detected genes. 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 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 $umol/gDW/h$. Metabolites detected experimentally can be assigned to be present in the metabolic model. Measured uptakes and secretions in growth media or biofluids can be used to constrain the uptakes and secretions of the model quantitatively. Furthermore, growth conditions information such as growth media composition can also be provided to constrain available uptakes in the 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.

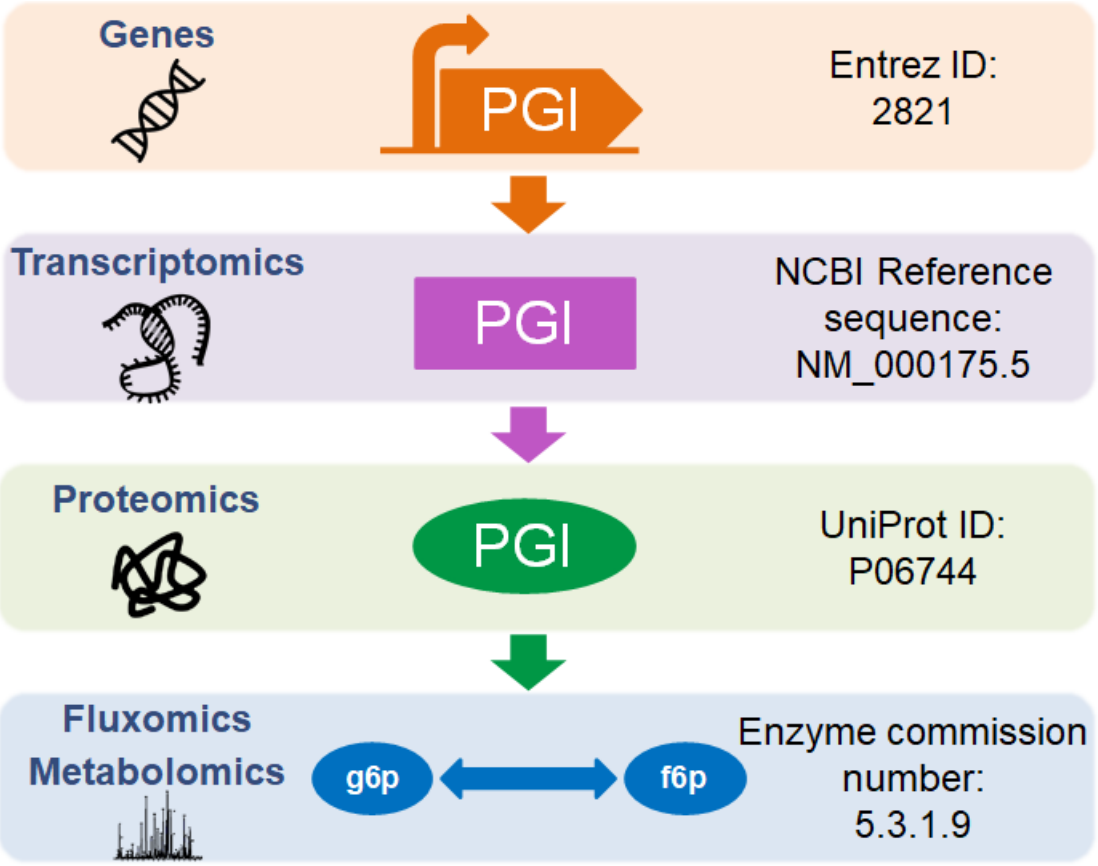


Figure 2: The gene-protein-reaction association or GPRs are boolean operators that describe the interactions of genes, transcripts, proteins, and reactions. For the reaction glucose-6-phosphate isomerase (PGI) to occur, the gene PGI must be activated (Entrez ID: 2821) leading to the transcription of the PGI mRNA (NCBI Reference Sequence: NM 000175.5). The mRNA is then translated into the glycolytic enzyme PGI (UniProt ID: P06744). Finally, the PGI enzyme (Enzyme Commission Number: 5.3.1.9) catalyse a reaction that interconverts glucose-6-phosphate (VMH ID: g6p) and fructose-6-phosphate (VMH ID: 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 understands basic MATLAB programming 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 multi-omic 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 [28], or the analysis of existing models [14]. Additionally, the integration of multi-omics data presents several challenges since they are generated using a variety of platforms, affecting storage and data formats significantly. The integration of multi-omic 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 setupCOBRA 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 cobratoolbox 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.

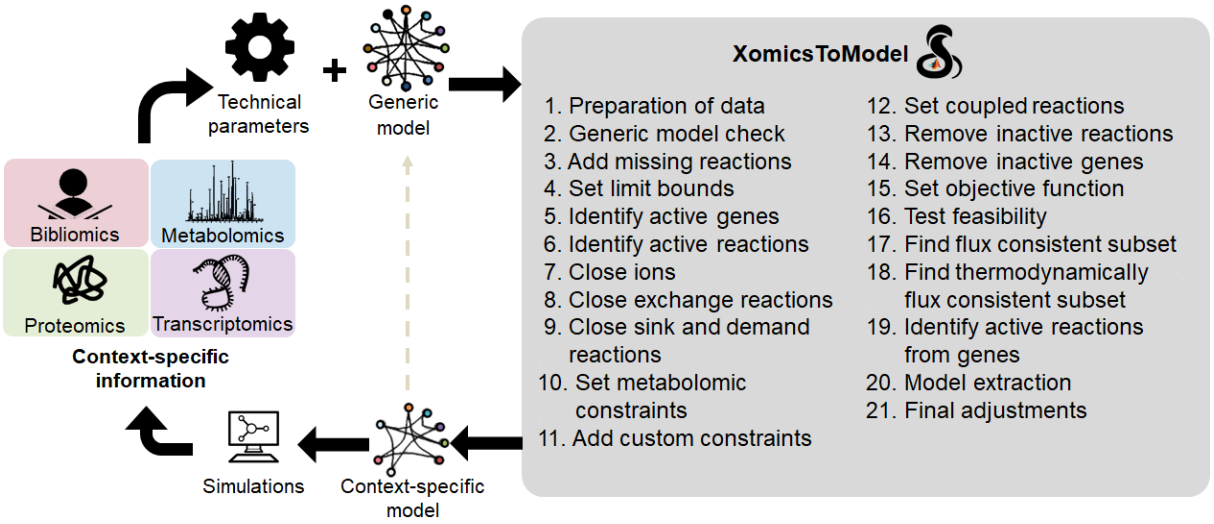


Figure 3: It is possible to create genome-scale models iteratively and systematically using the XomicsToModel pipeline. The generated models can be used to design or interpret the information based on bibliomic data from manual literature curation (red), as well as metabolomic (blue), proteomic (green) 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 multi-omic 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. For user's convenience, 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 ∼ 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 ∼ 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 $lb \leq v \leq ub$, 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 ∼ 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 (->): $lb = 0$; ub =param.TolMaxBoundary.
- Reverse (<-): lb =param.TolMinBoundary; $ub = 0$.
- Reversible (<=>) lb =param.TolMinBoundary; ub =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 ID 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; 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]. 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). If the param.debug is active (true), all results generated up to this point are saved in the file 3.debug_prior_to_identification_of_active_genes.mat.

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).

rxnID	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 | Set limit bounds ● TIMING ∼ 1 s

Description:

Considering the constraints $lb \leq v \leq ub$ shown in Equation 1, where lb is the lower bound and ub the upper bound, each lower bound in the draft model such that $lb_i = min(lb)$ and $lb_i \leq TolMinBoundary$ is set to param.TolMinBoundary and each upper bound such that $ub_i = max(ub)$ and $ub_i \geq TolMaxBoundary$ 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).

5 | 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 IDs.
- 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 $expressionLevel \geq$param.thresholdT, the gene is considered active.

Furthermore, 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 proteomics or transcriptiomics, preference will be given based on the option specified in the param.curationOverOmics.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
entrezGene1	220
entrezGene2	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
entrezGene1	24
entrezGene2	78
:	:

- param.thresholdP - The proteomic cutoff threshold for determining whether or not a gene is active (Default: $log2(1)$).
- specificData.activeGenes - List of Entrez ID 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 ID of genes known to be inactive based on the bibliomics data (Default: empty).

6 | Identify active reactions ● TIMING ∼ 10 s

Description:

The XomicsToModel pipeline uses several fields from the specificData variable to identify the set of metabolic 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 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 exo-metabolomic 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.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).

rxnID	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).

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

- 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	dsence	reference
cRxn1	crxn1, crxn2	1 1	2.0250	G	PMID: ****
cRxn2	crxn1, crxn3	1 -1	0.2265	G	PMID: ****
:	:	:	:	:	

- specificData.mediaData - Table containing the fresh media concentrations. Contains the reaction identifier, the maximum uptake ($umol/gDW/h$) 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 ($umol/l$; Default: empty).

rxnID	mediumMaxUptake	mediumConcentrations
EX_met1	-220	1345
EX_met2	-14916	191021
:	:	:

- 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.

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

- specificData.inactiveReactions - List of reactions known to be inactive based on bibliomic data (Default: empty).

7 | Close ions ● TIMING ∼ 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. If the debugging option is set, all results generated up to this point are saved in the file 7.debug_prior_to_exchange_constraints.mat.

Usable variables:

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

8 | Close exchange reactions ● TIMING ∼ 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).

rxnID	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 ∼ 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 relaxFBA, which is an optimisation problem that approximately minimises the number of reaction bounds to relax 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. relaxFBA is a flexible function for relaxing. If the debugging option is set, all results generated up to this point are saved in the file 9.debug_prior_to_metabolomicsBeforeReactionRemoval.mat.

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 metabolic constraints ● TIMING ∼ 10 s

Description:

The metabolic constraints can be obtained from two different sources: cell culture information and quantitative metabolomic measurements. Based on the overall goal, each of these datasets can be added either before or after the extraction algorithm step (Step 20) by using the param.growthMediaBeforeModelExtraction and param.metabolomicsBeforeModelExtraction parameters respectively.

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 mediumConcentrations in specificData.mediaData can be measured or directly obtained from the product specifications. The function 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 targeted metabolomics 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.) [26]. 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 (g dry weight) by the following formula:

$$v_{met} = \frac{metConcentrationEnd(umol/L) - metConcentrationStart(umol/L)}{interval(hr) * proteinConcentration(g/L) * proteinFractionInDW}$$

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 (tissues). Furthermore, the error of the flux estimation should also be calculated by propagating the uncertainty of all the non-exact values:

$$\sigma\Delta metConcentration = \sqrt{(\sigma metConcentrationEnd)^2 + (\sigma metConcentrationStart)^2}$$

$$\sigma\nu = \frac{\sqrt{\left(\frac{\sigma\Delta metConcentration}{\Delta metConcentration}\right)^2 + \left(\frac{\sigma proteinConcentration}{proteinConcentration}\right)^2} \cdot |vmet|}{interval \cdot proteinFractionInDW}$$

Last but not least, measured metabolites should be mapped into the model namespace where metabolite IDs and/or exchange reaction IDs for each metabolite should be found (ideally by cross-mapping standard metabolite IDs 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],

$$\begin{aligned} \min_{v,p,q\in\mathbb{R}^n} \quad & (v_{exp}-v)^T \mathrm{diag}(w_{exp})(v_{exp}-v) + p^T \mathrm{diag}(w_l) p + q^T \mathrm{diag}(w_u) q \\ \text{s.t.} \quad & Sv = 0, \\ & Cv \leq d, \\ & lb - p \leq v \leq ub + q, \\ & 0 \leq p, \\ & 0 \leq q, \end{aligned} \tag{2}$$

where $p \in \mathbb{R}^n_{\geq 0}$ and $q \in \mathbb{R}^n_{\geq 0}$ are non-negative variables that allow the lower (lb) and upper bound (ub) 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

and are derived from the mean ('mean', default), standard deviation ('SD'), or relative standard deviation ('RSD') of the experimental reaction flux.

To avoid numerical errors in the 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 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 metabolomics-based constraints 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 relaxFBA. If the debugging option is set, all results generated up to this point are saved in the file 10.debug_prior_to_custom_constraints.mat.

Usable variables:

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

rxnID	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.

rxnID	rxnNames	mean	SD	units	platform
EX_mMet1	Exchange of mMet1	100	8x10^{-4}	$umol/gDW/h$	LC-MS
EX_mMet2	Exchange of mMet2	4x10^{-3}	3x10^{-4}	$umol/gDW/h$	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 metabolomics 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 relaxFBA. If the debugging option is set, all results generated up to this point are saved in the file 11.debug_prior_to_setting_coupled_reactions.mat.

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).

rxnID	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 [30]. 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 distribution 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 IDs, 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. If the debugging option is set, all results generated up to this point are saved in the file 12.debug_prior_to_removing_inactive_reactions.mat.

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 Step 6, 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 relaxFBA. 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. If the debugging option is set, all results generated up to this point are saved in 13.debug_prior_to_removing_inactive_genes.mat.

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 the Step 6 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 relaxFBA , 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 5), 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 ID 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 ~ 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 relaxFBA. If the debugging option is set, all results generated up to this point are saved in the file 16.debug_prior_to_flux_consistency_check.mat.

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 6, 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. If the debugging option is set, all results generated up to this point are saved in the file 17.debug_prior_to_thermo_flux_consistency_check.mat.

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 ~ 10 s

Description:

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.

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 $\exists \ell \in \mathbb{R}^m_{\geq 0}$ such that $N^T \ell = 0$, and external reactions are not stoichiometrically consistent, that is $\nexists \ell \in \mathbb{R}^m_{\geq 0}$ such that $B^T \ell = 0$, as they represent net exchange of mass across the boundary of the system. Previously, Desouki *et al.* [8] introduced the CycleFreeFlux algorithm, which reduces a given flux vector to its thermodynamically feasible part, using a linear optimisation post-processing step. Inspired by this incisive result, we observed that a thermodynamically feasible flux may be computed as a single linear optimisation problem

$$\begin{aligned} \min_{z,w} \quad & \|z\|_1 + c^T \cdot w \\ \text{s.t.} \quad & Nz + Bw = 0 \ : r \\ & l \leq z \leq u \ : s \\ & y \leq w \leq x \ : t \end{aligned} \tag{3}$$

where $\|z\|_1$denotes the one-norm of internal reaction fluxes, $c^T \cdot w$ is a linear objective of external reaction fluxes, while $l \in \{0, -\infty\}^m$ and $u \in \{0, \infty\}^m$ denote lower and upper bounds on internal

reaction fluxes, $y, x \in \mathbb{R}^k$ denote lower and upper bounds on external reaction fluxes and the rest are dual variables. The optimality conditions of Problem 3 are

$$\begin{aligned} Nz + Bw &= 0 \\ \nabla \|z\|_1 = \text{sign}(z) &= N^T y + s \\ c &= B^T y + t \end{aligned}$$

Since $l \in \{0, -\infty\}^m$ and $u \in \{0, \infty\}^m$ then s_j, the dual variable to the inequality constraints on internal reaction j, is non-zero if and only if $z_j^\star$ is zero, that is $s_j^\star \neq 0 \iff z_j^\star = 0$. Therefore $z_j^\star \neq 0 \iff \text{sign}(z^\star) = N^T y^\star$, which enforces energy conservation and the second law of thermodynamics on the optimal vector of nonzero internal reaction fluxes [10]. It is important to reiterate that, for internal reactions only, positive lower bounds and negative upper bounds are eschewed to prevent a set of bounds from enforcing thermodynamically infeasible flux around a stoichiometrically balanced cycle [8]. Therefore, internal reactions forced to be active, i.e., the reactions for which $lb > 0$ or $ub < 0$ are assumed to be external reactions.

Problem 3 may be used to compute a single flux vector that is thermodynamically feasible. Specifically, we define thermodynamically feasibility as the requirement that any net flux to be driven by a change in chemical potential for the corresponding reaction, that is

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

where y is a vector of chemical potentials. 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 $\text{sign}(v) = -\text{sign}(N^T y)$, since $N_j^T y \neq 0 \nRightarrow v \neq 0$. However, only reactions with non-zero thermodynamically feasible net flux are assigned to be thermodynamically consistent.

One must compute multiple thermodynamically feasible net flux vectors in order to approximate the largest thermodynamically consistent subset of a given flux consistent model, because different combinations of reactions can be active in thermodynamically feasible fluxes with disjoint sparsity patterns. Inspired by the use of randomly weighted objectives to efficiently explore the set of flux consistent reactions [27], by default, we employ a similar strategy to bias toward thermodynamically feasible flux vectors that have non-zero flux in random subsets of reactions. A greedy sequence of optimisation problems is then used to iteratively increase the number of reactions in the thermodynamically consistent subset.

At each iteration the following cardinality optimisation problem is solved

$$\begin{aligned} \min_{z,w,p,q} \quad & g^T \|z\|_0 + \beta 1^T (p+q) \\ \text{s.t.} \quad & Nz + Bw = 0 \\ & z - p + q = 0 \\ & l \leq z \leq u \\ & y \leq w \leq x \\ & 0 \leq p \\ & 0 \leq q \end{aligned} \tag{4}$$

where $g^T \|z\|_0$ denotes the weighted zero-norm of internal reaction fluxes, where the zero-norm for each reaction is weighted individually, that is

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

Here $g \in \mathbb{R}^n$ is standard random vector generated from the zero-mean normal distribution in the open interval $(-0.5, 0.5)$ (the random part), except $g_j = 0$ if the j^{th} reaction has already been identified

as part of the thermodynamically consistent set (the greedy part). Minimisation of the one-norm of internal net reaction flux, to promote thermodynamic feasibility [8], is achieved by constraining net flux to be equal to the difference between two non-negative vectors, that is $z = p - q$ and minimising the sum of forward net reaction flux $p \in \mathbb{R}^n_{\geq 0}$ and reverse net reaction flux $q \in \mathbb{R}^n_{\geq 0}$. Bounds on z and w are the same as in Problem 3. Each iteration of Problem 4 is efficiently solved using a sequence of linear optimisation problems, as described in detail elsewhere.

Given a solution Problem 4, thermodynamic consistency is checked using a linear optimisation step [8]. The scalar weight $\beta > 0$ is chosen to trade off between cardinality optimisation of internal reaction fluxes, to explore the thermodynamically consistent subset of the generic model, and minimising the one norm of the sum of forward and reverse reaction rates, to promote thermodynamic consistency. By default, $\beta = 0.1$, which computational experiments determined to be approximately optimal for genome-scale models tested. If β is too small, too few of the reaction fluxes are thermodynamically feasible, while if β is too large, exploration of the thermodynamically consistent set is impeded.

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 6, 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. If the debugging option is set, all results generated up to this point are saved in the file 18.debug_prior_to_create_dummy_model.mat.

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.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 ∼ 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 [31] and extraction of a neuronal model [24] from Recon3D supports the use of 'oneRxnPerActiveGene' as the default. If the debugging option is set, all results generated up to this point are saved in the file 19.debug_prior_to_create_tissue_specific_model.mat.

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 [33], and 'thermoKernel', the default algorithm, which extracts a stoichiometrically, thermodynamically and flux consistent context-specific model. The 'FastCore' algorithm has been previously described [33]. 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 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 weights 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 net reaction flux and reverse net reaction flux, the rate of consumption of each metabolite is $Fp + Rq$ and the rate of production of each metabolite is then $Fq + Rp$. Therefore, letting $\bar{N} := F + R$, we observe that $s := (F + R)p + (F + R)q = \bar{N}p + \bar{N}q \geq 0$ represents the sum of the rate of production and consumption of each metabolite. If metabolite i is not present in a network, then $s_i = 0$.

To compute 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 following optimisation problem

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

which is an extension of Problem 4. When $h_i > 0$, the objective term $h_i^T \|s_i\|_0$ represents minimisation of the zero-norm of the rate of production and consumption of metabolite i. However, 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 $p = q \neq 0$ implies $s > 0$. Therefore, we introduce the variable $r := \bar{N}z$ as an approximation to the sum of production and consumption due to net reaction flux, which is $\bar{N}|z|$, but is non-trivial to implement. If $r_j := \bar{N}_j z > 0$ then $\bar{N}_j |z| > 0$ 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 $f = -\min(h, 0)$ so that optimisation of the cardinality of s and r is concordant.

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.

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 [27] whereby we choose a random vector from a uniform rectangular set, $d \in \{\mathbb{R}^n | \text{ unif}(0,1)\}$, 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 4 is efficiently solved using a sequence of linear optimisation problems, as described in detail elsewhere.

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. If the debugging option is set, all results generated up to this point are saved in the file 20.debug_after_create_tissue_specific_model.mat.

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 * 100;
```

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.

Set limit bounds

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

Identify active genes

5 | 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.

Identify active reactions

6 | 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 cell culture and/or exometabolomic constraints added. 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 IDs 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 thermoKernel, the optimizeCardinality procedure, an extraction summary, and statistics for the new model.

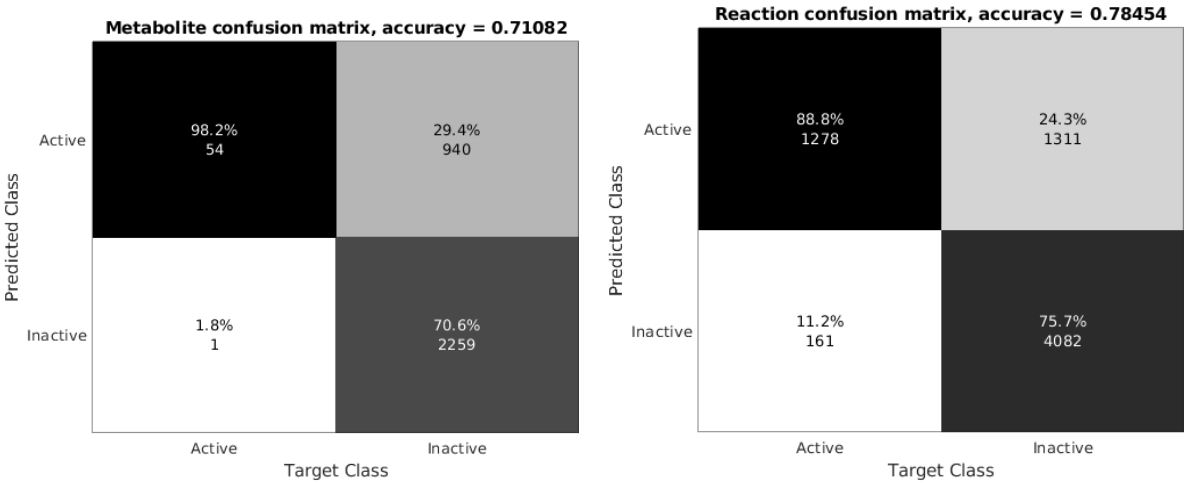


Figure 4: Comparison of specified and actual metabolites and reactions in an extracted model. Output from the 'thermoKernel' algorithm for an example where there is a trade off between presence and absence of metabolites and reactions in an extracted model.The predicted class is the specification of metabolites by weights to be present (active) or absent (inactive) while the target class refers to the presence (active) or absent (inactive) metabolites in the extracted model. The predicted class is the specification of reactions by weights to be present (active) or absent (inactive) while the target class refers to the presence (active) or absent (inactive) reactions in the extracted model.

Final

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

To test the predictive capacity and size of alternative versions of the *iDopaNeuro1* model, four models were generated with the function XomicsToMultipleModels (See supplementary information) while excluding context-specific information such as bibliomics, metabolomics, transcriptomics, and fresh medium concentrations (Figure 5). On average, the *iDopaNeuro1* model with all constraints correctly predicts uptakes and secretions with the highest frequency based on the result performed by

the function modelPredictiveCapacity (See supplementary information). It can also be seen how different context-specific information influences the number of independent variables in the stoichiometric matrixFor example, culture medium information has the biggest influence on the rank(S) because it determines which nutrients are available to the neuron, requiring activity of corresponding specific metabolic pathways. This showcases the importance of the inclusion of high-quality, highly context-specific data from multiple origins for an accurate description of the biological system of interest [21, 25]. More extensive validation of the *iDopaNeuro1* model and its application to study neuronal energy metabolism has been described in Preciat *et al.* [24].

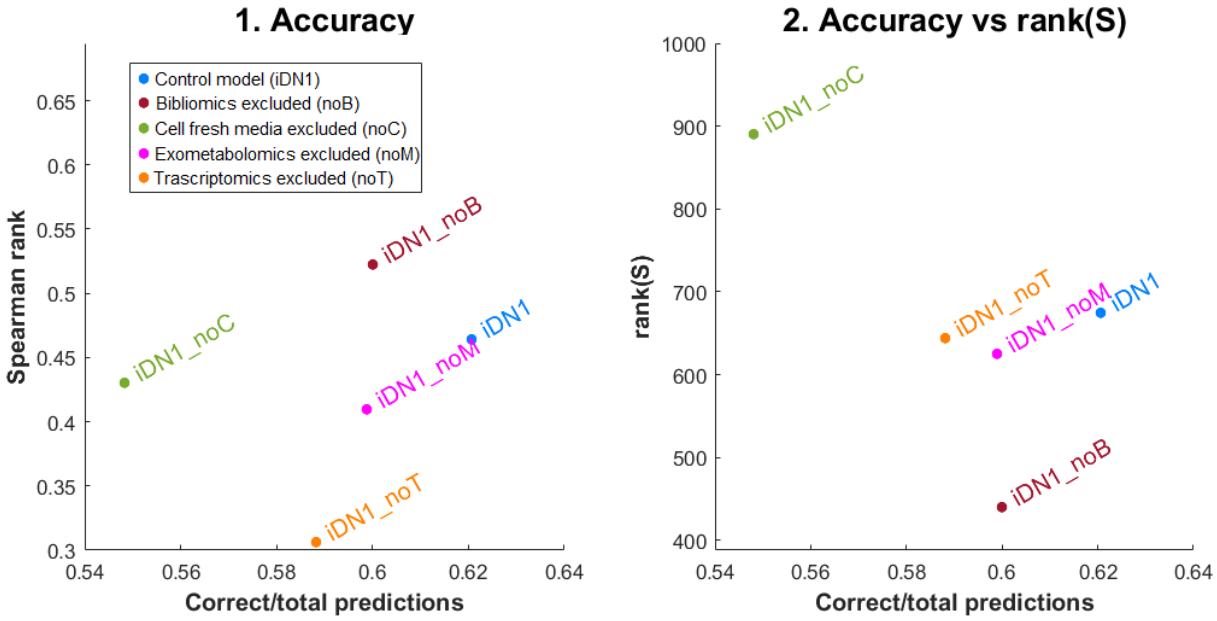


Figure 5: Predictive capacity, with all omics data (blue), and selected types of omics data omitted in the *iDopaNeuro1* model. Predictions based on eight cellular objectives as detailed in Preciat *et al.* [24]. 1) Inclusion of all omics data leads to the highest qualitative (ratio of correct/total prediction of uptake/secretion/neither) and quantitative predictive accuracy (Spearman rank of predicted versus measured exchange reaction rates). 2) Qualitative accuracy (ratio of correct/total prediction of uptake/secretion/neither) is not simply a function of model flexibility, as measured by the rank of the stoichiometric matrix, rank(S).

? TROUBLESHOOTING

Step	Problem	Possible reason	Solution
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. 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. If it is over-constrained, relaxFBA relaxes the minimum number of reaction bounds in the draft model to make it feasible.
3 |	RXN1 in specificData.rxns2add.rxnID 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: 'met1 + 2 met2 -> met3 + met4' 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: specificData.rxns2add, param.TolMinBoundary, param.TolMaxBoundary, specificData.rxns2constrain and specificData.exoMet described in Sections 3, 4, 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.

● **TIMING**

Step 1, Preparation of data: ~ 5 s
Step 2, Generic model check: ~ 1 s
Step 3, Add missing reactions: ~ 30 s
Step 4, Set limit bounds: ~ 1 s
Step 5, Identify active genes: $\sim 10^2$ s
Step 6, Identify active reactions: ~ 10 s
Step 7, Close ions: ~ 1 s
Step 8, Close exchange reactions: ~ 1 s
Step 9, Close sink and demand reactions: ~ 10 s
Step 10, Set metabolic constraints: ~ 10 s
Step 11, Add custom constraints: ~ 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: ~ 10 s
Step 17, Find flux consistent subset: $\sim 10^2$ s
Step 18, Find thermodynamically consistent subset: ~ 10 s
Step 19, Identify active reactions from genes: ~ 10 s
Step 20, Model extraction: $\sim 10^2$ s
Step 21, Final: ~ 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	Conceptualisation; 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] Jonathan L. Robinson and Jens Nielsen. "Integrative Analysis of Human Omics Data Using Biomolecular Networks". In: *Molecular bioSystems* 12.10 (Oct. 2016), pp. 2953–2964. DOI: 10.1039/c6mb00476h.

[26] Vinod P. Shah et al. "Bioanalytical method validation–a revisit with a decade of progress". In: *Pharmaceutical Research* 17.12 (2000), pp. 1551–1557. ISSN: 07248741. DOI: 10.1023/A:1007669411738. URL: http://link.springer.com/10.1023/A:1007669411738 (visited on 08/11/2021).

[27] 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.

[28] 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.

[29] 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.

[30] 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.

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

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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 MATLAB live-script tutorial on how to use the XomicsToModel pipeline.

Supplementary Information 1 - Additional tools

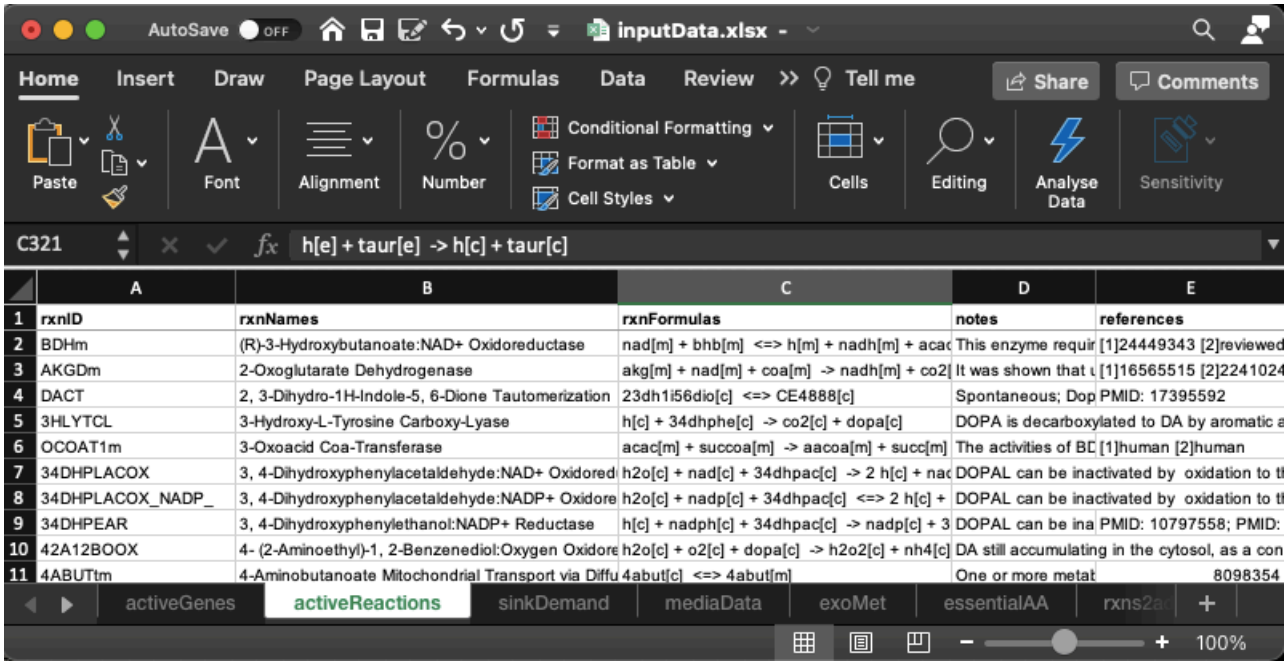
Preprocessing Omics Data

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 th	
8	34DHPLACOX_NADP_	3, 4-Dihydroxyphenylacetaldehyde:NADP+ Oxidore	h2o[c] + nadp[c] + 34dhpac[c] <=> 2 h[c] +	DOPAL can be inactivated by oxidation to th	
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:

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

Lastly, it is also possible to include an older version of the specificData variable in order to update the context-specific data.

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).

rxnID	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'[33] 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 struct 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 5 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').

rxnID	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').

Supplementary Information 2 - XomicsToModel pipeline tutorial

XomicsToModel pipeline tutorial

This tutorial illustrates how to prepare the data that will be used by the `XomicsToModel` function [1] of the COBRA Toolbox V3.0 [2]. This function facilitates the generation of a thermodynamic-flux-consistent, context-specific, genome-scale metabolic model in a single command by combining a generic model with bibliomic, transcriptomic, proteomic, and metabolomic data. To ensure the network's quality, several thermodynamic consistency checks are implemented within the function. To generate a thermodynamic-flux-consistent, context-specific, genome-scale metabolic model, the function requires two inputs: a generic COBRA model a variable containing the context-specific data and a variable with technical information defined by the user.

This tutorial shows how to extract a context-specific genome-scale model of a dopaminergic neuron (*iDopaNeuro1* [3]) from the human generic model Recon3D [4] using the `XomicsToModel` function. The *iDopaNeuro1* [3] model is extracted using data from manual curation of a dopaminergic neuron to identify active and inactive genes, reactions, and metabolites, as well as information from in vitro experiments such as exometabolomic quantification and transcriptomic sequencing of cell culture of pluripotent stem cell-derived dopaminergic neurons.

Generic COBRA model

The COBRA model can be found in a file with the extension ".mat". Recon3D [1], which is found in the VMH database [2], can be used as a generic model for human metabolism.

```
modelPath = uigetdir;
genericModelName = 'model.mat';
load([modelPath filesep genericModelName])
```

Context-specific information

This type of information represents the biological system's phenotype and can be obtained through a review of the literature or experimental data derived from the biological system. The following context-specific information can be used by the function:

Automated data integration

Tables or multiple data sets can be inserted in an external worksheet document so that the `preprocessingOmicsModel` function can include them in the `options` variable. The name of the sheet corresponding to the options field must be the same as those specified above and in the manuscript, or they will be omitted.

Bibliomic data. It is derived from manual reconstruction following a review of the literature. This includes data on the activation or inactivation of genes, reactions, or metabolites. Another example is the addition of coupled reactions or the constraints of different reactions based on phenotypic observations.

- **`specificData.activeGenes:`** List of Entrez ID of genes that are known to be active based on the bibliomic data (Default: empty).
- **`specificData.addCoupledRxns:`** Logical, should the coupled constraints be added (Default: `true`).

- **`specificData.coupledRxns:`** Logical, indicates whether curated data should take priority over omics data (Default: false).
- **`specificData.essentialAA:`** List exchange reactions of essential amino acid (Default: empty).
- **`specificData.inactiveGenes:`** List of Entrez ID of genes known to be inactive based on the bibliomics data (Default: empty).
- **`specificData.presentMetabolites:`** List of metabolites known to be active based on the bibliomics 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).
- **`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).

```
dataFolder = uigetdir;
bibliomicData = 'bibliomicData.xlsx';
specificData = preprocessingOmicsModel([dataFolder bibliomicData], 1, 1);
```

Metabolomic data. Differences in measured concentrations of metabolites within cells, biofluids, tissues, or organisms are translated into flux units of flux ($\mu\mathrm{mol}/\mathrm{gDW}/h$).

- **`options.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, and the sign for uptakes (Default: empty).
- **`options.exoMet:`** Table with the fluxes obtained from exometabolomics 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.
- **`options.mediaData:`** Table containing the initial media concentrations. Contains the reaction identifier, the maximum uptake ($\mu\mathrm{mol}/\mathrm{gDW}/h$) based on the concentration of the metabolite and the concentration ($\mu\mathrm{mol}$; Default: empty).

```
specificData.exoMet = readtable([dataFolder 'exoMet']);
```

Proteomic data. This information indicates the level of expression of the proteome.

- **`options.proteomics:`** Table with a column with Entrez ID's and a column for the corresponding protein levels (Default: empty).

Transcriptomic data. Indicates the level of transcriptome expression and can also be used to calculate reaction expression. Transcriptomic data can be analysed in a variety of formats, including RPM, RPKM, FPKM, TPM, TMM, DESeq, SCnorm, GeTMM, ComBat-Seq, and raw read counts.

- **`options.transcriptomicData:`** Table with a column with Entrez ID's and a column for the corresponding transcriptomics expression value (Default: empty).

```
specificData.transcriptomicData = readtable([dataFolder 'transcriptomicData']);
specificData.transcriptomicData.genes = string(specificData.transcriptomicData.genes);
```

Technical data

With these options, technical constraints can be added to the model, as well as setting the parameters for model extraction or debugging.

Bounds. They are the instructions that will be set in the boundaries.

- **`options.boundPrecisionLimit:`** Precision of flux estimate, if the absolute value of the lower bound (`model.lb`) or the upper bound (`model.ub`) are lower than `options.boundPrecisionLimit` but higher than 0 the value will be set to the boundPrecisionLimit (Default: primal feasibility tolerance).
- **`options.TolMaxBoundary:`** The reaction boundary's maximum value (Default: $1e3$).
- **`options.TolMinBoundary:`** The reaction boundary's minimum value (Default: $-1e3$).
- **`options.relaxOptions:`** A structure array with the relaxation options (Default: `options.relaxOptions.steadyStateRelax = 0`).

```
param.TolMinBoundary = -1e5;
param.TolMaxBoundary = 1e5;
feasTol = getCobraSolverParams('LP', 'feasTol');
param.boundPrecisionLimit = feasTol * 10;
```

Debugging options. The user can specify the function's verbosity level as well as save the results of the various blocks of the function for debugging.

- **`options.debug:`** Logical, should the function save its progress for debugging (Default: false).
- **`options.diaryFilename:`** Location where the output be printed in a diary file (Default: 0).
- **`options.printLevel:`** Level of verbose that should be printed (Default: 0).

```
param.printLevel = 1;
param.debug = true;
if isunix()
    name = getenv('USER');
else
    name = getenv('username');
end
param.diaryFilename = [pwd filesep datestr(now,30) '_' name '_diary.txt'];
```

Exchange reactions. They are the instructions for the exchange, demand, and sink reactions.

- **`options.addSinksexoMet:`** Logical, should sink reactions be added for metabolites measured in the media but without existing exchange reaction (Default: false).
- **`options.closeIons:`** Logical, it determines whether or not ion exchange reactions are closed. (Default: false).
- **`options.closeUptakes:`** Logical, decide whether or not all of the uptakes in the generic model will be closed (Default: false).

- **`options.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.closeIons = false;
param.sinkDMinactive = 1;
param.nonCoreSinksDemands = 'closeAll';
```

Extraction options. The solver and parameters for extracting the context-specific model.

- **`options.activeGenesApproach:`** String with the name of the active genes approach will be used (Possible options: `'oneRxnsPerActiveGene'` or `'deletModelGenes'`; Default: `'oneRxnsPerActiveGene'`).
- **`options.fluxCCmethod:`** String with thee name of the algorithm to be used for the flux consistency check (Possible options: `'swiftcc'`, `'fastcc'` or `'dc'`, Default: `'fastcc'`).
- **`options.fluxEpsilon:`** Minimum non-zero flux value accepted for tolerance (Default: Primal feasibility tolerance).
- **`options.thermoFluxEpsilon:`** Flux epsilon used in `'thermoKernel'` (Default: feasibility tolerance).
- **`options.tissueSpecificSolver:`** The name of the solver to be used to extract the context-specific model (Possible options: `'thermoKernel'` and `'fastcore'`; Default: `'thermoKernel'`).

```
param.activeGenesApproach = 'oneRxnPerActiveGene';
param.tissueSpecificSolver = 'thermoKernel';
param.fluxEpsilon = feasTol * 10;
param.fluxCCmethod = 'fastcc';
```

Data-specific parameters. Parameters that define the minimum level of transcript/protein to be considered as present in the network (threshold) and whether the transcripts below the set threshold should be removed from the model.

- **`options.metabolomicsBeforeModelExtraction:`** Logical, should the metabolomics data be added before the model extraction (Default: true).
- **`options.metabolomicWeights:`** String indicating the type of weights to be applied for metabolomics fitting (Possible options: `'SD'`, `'mean'` and `'RSD'`; Default: `'mean'`).
- **`options.weightsFromOmics:`** Should gene weights be assigned based on the omics data (Default: 0).
- **`options.thresholdP:`** The proteomic cutoff threshold for determining whether or not a gene is active (Default: $\log2(1)$).
- **`options.inactiveGenesTranscriptomics:`** Logical, indicate if inactive genes in the transcriptomic analysis should be added to the list of inactive genes (Default: true).
- **`options.thresholdT:`** The transcriptomic cutoff threshold for determining whether or not a gene is active (Default: $\log2(1)$)

- **`param.curationOverOmics:`** Logical, indicates whether curated data should take priority over omics data (Default: false).
- **`param.setObjective:`** Linear objective function to optimise (Default: empty).

```
param.boundsToRelaxExoMet = {'b'};
param.thresholdT = 2;
param.inactiveGenesTranscriptomics = true;
param.setObjective = '';
param.curationOverOmics = true;
param.weightsFromOmics = 1;
param.metabolomicWeights='mean';
param.addCoupledRxns = 1;
param.closeUptakes = true;
```

Function

```
[solverOK, solverInstalled] = changeCobraSolver('gurobi','all', 0);
[contextSpecificModel, modelGenerationReport] = XomicsToModel(model, specificData, par
```

```
------------------------------------------------------------------

XomicsToModel input specificData:

             inputData: '~/work/sbgCloud/programReconstruction/projects/exoMetDN/data/xomics/bibliomicDat
           activeGenes: {239×1 cell}
       activeReactions: {329×1 cell}
       cellCultureData: [1×6 table]
           coupledRxns: [11×5 table]
           essentialAA: [9×1 table]
         inactiveGenes: {61×1 cell}
             mediaData: [56×3 table]
    presentMetabolites: [45×4 table]
              rxns2add: [9×9 table]
        rxns2constrain: [81×5 table]
           rxns2remove: [228×6 table]
            sinkDemand: [49×8 table]
               exoMet: [89×6 table]
    transcriptomicData: [18530×2 table]

XomicsToModel input param:

                      TolMinBoundary: -100000
                      TolMaxBoundary: 100000
                 boundPrecisionLimit: 1e-05
                          printLevel: 1
                               debug: 1
                       diaryFilename: '/Users/gpreciat/20210907T223113_gpreciat_diary.txt'
                           closeIons: 0
                      sinkDMinactive: 1
                 nonCoreSinksDemands: 'closeAll'
                 activeGenesApproach: 'oneRxnPerActiveGene'
                tissueSpecificSolver: 'thermoKernel'
                         fluxEpsilon: 1e-05
                        fluxCCmethod: 'fastcc'
                 boundsToRelaxExoMet: {'b'}
                           thresholdT: 2
         inactiveGenesTranscriptomics: 1
                        setObjective: ''
```

```
                    curationOverOmics: 1
                     weightsFromOmics: 1
                   metabolomicWeights: 'mean'
                       addCoupledRxns: 1
                         closeUptakes: 1
                    inactiveReactions: []
                           thresholdP: 0
                           uptakeSign: -1
                    thermoFluxEpsilon: 1e-06
    metabolomicsBeforeReactionRemoval: 1
                     workingDirectory: '/Users/gpreciat'

Replacing reaction name DM_atp_c_ with ATPM, because it is not strictly a demand reaction.

Old reaction formulas
ATPS4mi    adp[m] + pi[m] + 4 h[i]     ->    h2o[m] + 3 h[m] + atp[m]
CYOOm2i    o2[m] + 8 h[m] + 4 focytC[m]     ->    2 h2o[m] + 4 ficytC[m] + 4 h[i]
CYOOm3i    o2[m] + 7.92 h[m] + 4 focytC[m]     ->    1.96 h2o[m] + 4 ficytC[m] + 0.02 o2s[m] + 4 h[i]
CYOR_u10mi    2 h[m] + 2 ficytC[m] + q10h2[m]     ->    q10[m] + 2 focytC[m] + 4 h[i]
NADH2_u10mi    5 h[m] + nadh[m] + q10[m]     ->    nad[m] + q10h2[m] + 4 h[i]
  0×0 empty char array

New reaction formulas
ATPS4m    4 h[c] + adp[m] + pi[m]     ->    h2o[m] + 3 h[m] + atp[m]
CYOOm2    o2[m] + 8 h[m] + 4 focytC[m]     ->    2 h2o[m] + 4 h[c] + 4 ficytC[m]
CYOOm3    o2[m] + 7.92 h[m] + 4 focytC[m]     ->    1.96 h2o[m] + 4 h[c] + 4 ficytC[m] + 0.02 o2s[m]
CYOR_u10m    2 h[m] + 2 ficytC[m] + q10h2[m]     ->    4 h[c] + q10[m] + 2 focytC[m]
NADH2_u10m    5 h[m] + nadh[m] + q10[m]     ->    4 h[c] + nad[m] + q10h2[m]

Feasible generic input model.


------------------------------------------------------------------
Adding 9 reactions ...

Reaction boundaries not provided. Default (min and max) values will be used based on the reaction formula.
acleua    h2o[c] + acleu_L[c]     ->    ac[c] + leu_L[c]
acthra    h2o[c] + acthr_L[c]     ->    ac[c] + thr_L[c]
acileua    h2o[c] + acile_L[c]     ->    ac[c] + ile_L[c]
acglua    h2o[c] + acglu[c]     ->    glu_L[c] + ac[c]
CE1554tm    CE1554[c]     <=>    CE1554[m]
RE2031M    accoa[m] + ala_L[m]     <=>    h[m] + coa[m] + CE1554[m]
RE2642C    h2o[c] + CE1554[c]     <=>    ac[c] + ala_L[c]
CE1554t    CE1554[c]     <=>    CE1554[e]
EX_CE1554[e]    CE1554[e]     <=>
------------------------------------------------------------------

Identifying the stoichiometrically consistent subset...

--- findStoichConsistentSubset START ----
--- Summary of stoichiometric consistency ----
  5837     10608     totals.
     0      1810     heuristically external.
  5837      8798     heuristically internal:
  5837      8798     ... of which are stoichiometrically consistent.
     0         0     ... of which are stoichiometrically inconsistent.
     0         0     ... of which are of unknown consistency.
  5837      8798     Confirmed stoichiometrically consistent by leak/siphon testing.
--- findStoichConsistentSubset END ----

Feasible stoichiometrically consistent model with new reactions.

Feasible model with default bounds.
```

```
------------------------------------------------------------------

Assuming gene expression is NaN for 161 genes where no transcriptomic data is provided.

Model statistics:

5837 x 10608 stoichiometric matrix.
1563 exchange reactions.
138 exchange reactions in the core reaction set.
40 exchange reactions in the rxns2Constrain set.

1523 exchange reactions with uptake closed

247 closed non-core sink/demand reactions via param.nonCoreSinksDemands = closeAll
0 core sink/demand reactions.
0 open core sink/demand reactions.

Feasible after closing non-core sink/demand reactions.

------------------------------------------------------------------

Adding growth media information...
The following reactions could not be constrained since they are not present in the model:
    {'EX_HC01944[e]'}
    {'EX_adpcbl[e]' }
    {'EX_ca2[e]'    }
    {'EX_cl[e]'     }
    {'EX_mg2[e]'    }
    {'EX_selni[c]'  }
    {'EX_zn2[e]'    }

Adding constraints on 49 reactions


Feasible after application of media constraints.

------------------------------------------------------------------

Adding quantitative metabolomics constraints ...

Warning: There are duplicate rxnID entries in the metabolomic data. Only using data corresponding
to first occurance
Fit experimental flux method: zeroOne

79 measured exchange reaction rates.
79 perfectly fit measured exchange reaction rates.
0 imperfectly fit measured exchange reaction rates.
4 lower bounds relaxed.
4 external lower bounds relaxed.
4 measured external metabolite lower bounds relaxed.
0 unmeasured external metabolite lower bounds relaxed.
0 upper bounds relaxed.
0 external upper bounds relaxed.
0 measured external metabolite upper bounds relaxed.
0 unmeasured external metabolite upper bounds relaxed.

Fit reactions:
rxns{n}                     lb          wl          -p      lb_old           v        vexp        wexp
EX_2hb[e]                0.06609         Inf          -0           0     0.06724     0.06724       1.995
EX_34dhphe[e]             0.5363         Inf          -0           0      0.6734      0.6734       1.688
EX_adrnl[e]                0.364         Inf          -0           0      0.4786      0.4786       1.814
EX_ala_B[e]               0.8067         Inf          -0           0       0.845       0.845       1.583
EX_arach[e]              -0.1506           1          -0      -1e+05    0.006315    0.006315           2
EX_arachd[e]           0.0004675         Inf          -0      -1e+05   0.0007782   0.0007782           2
```

EX_bhb[e]	0.119	Inf	−0	0	0.1216	0.1216	1.985
EX_dopa[e]	0.6242	Inf	−0	0	0.7599	0.7599	1.634
EX_hdca[e]	3.282	Inf	−0	−1e+05	3.708	3.708	1.068
EX_nrpphr[e]	0.4056	Inf	−0	0	0.4753	0.4753	1.816
EX_ocdca[e]	−0.2742	1	−0	−1e+05	−0.08502	−0.08502	1.993
EX_ocdcea[e]	0.1272	Inf	−0	0	0.1614	0.1614	1.975
EX_octa[e]	0.02357	Inf	−0	−1e+05	0.02605	0.02605	1.999
EX_srtn[e]	0.2202	Inf	−0	0	0.263	0.263	1.935
EX_citr_L[e]	0.009373	Inf	−0	0	0.01139	0.01139	2
EX_c10crn[e]	0.001768	Inf	−0	−1e+05	0.002068	0.002068	2
EX_doco13ac[e]	0.006101	Inf	−0	−1e+05	0.01259	0.01259	2
EX_dca[e]	0.04424	Inf	−0	−1e+05	0.07249	0.07249	1.995
EX_4hpro[e]	0.2851	Inf	−0	−1e+05	0.3398	0.3398	1.896
EX_crtn[e]	0.09424	Inf	−0	0	0.1112	0.1112	1.988
EX_icit[e]	0.02212	Inf	−0	−1e+05	0.0233	0.0233	1.999
EX_Lkynr[e]	0.001837	Inf	−0	0	0.002147	0.002147	2
EX_acrn[e]	0.04061	Inf	−0	−1e+05	0.04225	0.04225	1.998
EX_pcrn[e]	0.0007726	Inf	−0	0	0.0008125	0.0008125	2
EX_pmtcrn[e]	2.123e−05	Inf	−0	−1e+05	0.000195	0.000195	2
EX_3hpp[e]	0.07464	Inf	−0	0	0.08577	0.08577	1.993
EX_acgly[e]	0.01632	Inf	−0	0	0.02334	0.02334	1.999
EX_acthr_L[e]	−0.09061	1	−0.08579	0	−0.08579	−0.08579	1.993
EX_C02712[e]	0.01301	Inf	−0	0	0.01351	0.01351	2
EX_CE1557[e]	−2.275	1	−0	−1e+05	−2.071	−2.071	1.189
EX_CE2028[e]	2.31	Inf	−0	0	2.348	2.348	1.154
EX_HC00900[e]	0.2203	Inf	−0	0	0.2218	0.2218	1.953
EX_Nacasp[e]	0.3114	Inf	−0	0	0.3174	0.3174	1.908
EX_acile_L[e]	−0.003047	1	−0.002207	0	−0.002207	−0.002207	2
EX_acleu_L[e]	−0.002727	1	−0.002383	0	−0.002383	−0.002383	2
EX_acglu[e]	−0.1001	1	−0.09751	0	−0.09751	−0.09751	1.991
EX_ddca[e]	0.008962	Inf	−0	−1e+05	0.01002	0.01002	2
EX_glyc_R[e]	0.1911	Inf	−0	−1e+05	0.1978	0.1978	1.962
EX_glyclt[e]	0.1666	Inf	−0	−1e+05	0.1694	0.1694	1.972
EX_oaa[e]	0.167	Inf	−0	0	0.1838	0.1838	1.967
EX_HC00342[e]	0.007249	Inf	−0	0	0.008629	0.008629	2
EX_akg[e]	0.003671	Inf	−0	−1e+05	0.003861	0.003861	2
EX_asn_L[e]	0.782	Inf	−0	−4.293	0.8749	0.8749	1.566
EX_asp_L[e]	−1.584	1	−0	−3.865	−1.549	−1.549	1.294
EX_fum[e]	8.257	Inf	−0	0	8.964	8.964	1.012
EX_glu_L[e]	−1.764	1	−0	−3.865	−1.593	−1.593	1.283
EX_ile_L[e]	−11.32	1	−0	−94.11	−9.79	−9.79	1.01
EX_lac_L[e]	53.49	Inf	−0	0	54.87	54.87	1
EX_leu_L[e]	−19.79	1	−0	−96.81	−16.85	−16.85	1.004
EX_lys_L[e]	−16.61	1	−0	−100.2	−12.6	−12.6	1.006
EX_mal_L[e]	0.1072	Inf	−0	0	0.1113	0.1113	1.988
EX_met_L[e]	−2.825	1	−0	−24.51	−0.3147	−0.3147	1.91
EX_orn[e]	21.54	Inf	−0	0	22.11	22.11	1.002
EX_pro_L[e]	15	Inf	−0	−16.81	16.57	16.57	1.004
EX_ser_L[e]	22.95	Inf	−0	−50.25	25.66	25.66	1.002
EX_succ[e]	0.1158	Inf	−0	0	0.1354	0.1354	1.982
EX_val_L[e]	−9.331	1	−0	−97.03	−7.525	−7.525	1.017
EX_gly[e]	−16.05	1	−0	−50.25	−15.95	−15.95	1.004
EX_cys_L[e]	−5.145	1	−0	−27.84	−4.838	−4.838	1.041
EX_ala_L[e]	19.93	Inf	−0	−5.603	20.6	20.6	1.002
EX_his_L[e]	−1.174	1	−0	−27.05	−0.843	−0.843	1.585
EX_thr_L[e]	−12.37	1	−0	−96.44	−9.241	−9.241	1.012
EX_gln_L[e]	−13.08	1	−0	−322.1	−11.41	−11.41	1.008
EX_phe_L[e]	−0.7164	1	−0	−47.55	−0.5569	−0.5569	1.763
EX_tyr_L[e]	−2.63	1	−0	−47.28	−2.291	−2.291	1.16
EX_arg_L[e]	−17.79	1	−0	−84.82	−15.85	−15.85	1.004
EX_cit[e]	0.1352	Inf	−0	−1e+05	0.1415	0.1415	1.98
EX_etha[e]	−0.2915	1	−0	−1.506	−0.29	−0.29	1.922
EX_ptrc[e]	0.143	Inf	−0	−8.097	0.2593	0.2593	1.937
EX_trp_L[e]	−0.4732	1	−0	−9.482	−0.4081	−0.4081	1.857

```
EX_ura[e]                 0.0204         Inf          -0      -1e+05     0.02212     0.02212           2
EX_pyr[e]                 0.4858         Inf          -0      -94.88      0.5475      0.5475       1.769
EX_4abut[e]              0.01043         Inf          -0           0     0.01083     0.01083           2
EX_taur[e]               0.005998        Inf          -0           0    0.006414    0.006414           2
EX_actyr[e]              0.002463        Inf          -0           0    0.002711    0.002711           2
EX_CE1310[e]            -0.01455           1          -0           0    0.001975    0.001975           2
EX_glc_D[e]               -279.9           1          -0       -3286      -275.8      -275.8           1
EX_M03117[e]             0.001699        Inf          -0      -1e+05    0.004818    0.004818           2
EX_CE1554[e]               0.0231        Inf          -0           0     0.02564     0.02564       1.999

Relaxation of lower bounds:
rxns{n}                        lb          wl          -p      lb_old           v        vexp        wexp
EX_acthr_L[e]            -0.09061           1    -0.08579           0    -0.08579    -0.08579       1.993
EX_acile_L[e]           -0.003047           1    -0.002207          0    -0.002207   -0.002207          2
EX_acleu_L[e]           -0.002727           1    -0.002383          0    -0.002383   -0.002383          2
EX_acglu[e]              -0.1001            1     -0.09751          0     -0.09751    -0.09751       1.991
No relaxation of upper bounds.
...done.


Feasible after application of metabolomic constraints

------------------------------------------------------------

Checking for mismatches ...

                rxn                                  rxnNames           vExp           sdExp           vI
         EX_met_L[e]              Exchange of L-Methionine        -0.314695          2.5099       -0.314(
         EX_ocdca[e]     Exchange of Octadecanoate (N-C18:0)      -0.0850241        0.189166      -0.0850
        EX_CE1310[e]         Exchange of N-Acetyl-L-Cysteine       0.0019755       0.0165277       0.0019
         EX_arach[e]                  Exchange of Arachidate      0.00631452        0.156892      0.00631

Analysis of the reasons for 14 mismatches between sign of experimental and fit metabolite exchange:
... of whom 10 experimental metabolite not part of model:
    {0×0 char      }    {'Anserine'             }
    {0×0 char      }    {'N.acetylproline'      }
    {0×0 char      }    {'N.acetyltryptophan'   }
    {0×0 char      }    {'N.acetylphenylalanine'}
    {0×0 char      }    {'N.acetylvaline'       }
    {0×0 char      }    {'N.acetylarginine'     }
    {0×0 char      }    {'N.acetylglutamine'    }
    {0×0 char      }    {'N.acetylserine'       }
    {0×0 char      }    {'N.acetylhistidine'    }
    {'EX_glutar[e]'}    {0×0 char               }

... of whom 4 fit perfectly but experimental mean +/- SD includes zero:
    {'EX_met_L[e]' }    {'Methionine'      }
    {'EX_ocdca[e]' }    {'Stearic.acid'    }
    {'EX_CE1310[e]'}    {'N.acetylcysteine'}
    {'EX_arach[e]' }    {'Arachidic.acid'  }

... of whom 0 fit sign but not magnitude and experimental mean +/- SD includes zero:
... of whom 0 mean measured to be taken up but cannot be taken up:
... of whom 0 mean measured to be taken up but can only be secreted:
... of whom 0 mean measured to be taken up but secreted, even if it can be uptaken:
... of whom 0 mean measured to be secreted but cannot be secreted:
... of whom 0 mean measured to be secreted but must be uptaken:
... of whom 0 mean measured to be secreted but uptaken, even if it can be secreted:
------------------------------------------------------------

Adding custom constraints ...

tissueSpecificSolver = thermoKernel. Ignoring specificData.rxns2constrain for demand reactions, i.e. with
The following reactions could not be constrained since they are not present in the model:
```

```
    {'DM_ca2[c]'    }
    {'DM_clpn_hs[c]'}
    {'Pcm'          }

Adding constraints on 78 reactions

Feasible after application of custom constraints.


--------------------------------------------------------------

Adding 11 sets of coupled reactions ...

           coupledRxnId                                         constraints
    ___________________________    _______________________________________________________________________

    {'Phosphatidylcholine'    }    {'PCHOLP_hs + PLA2_2 + SMS >= 2.025'
    {'Adenosine Monophosphate'}    {'AMPDA + NTD7 >= 0.2265'
    {'Glutamate'              }    {'- ALATA_L + GLUCYS + GLUDxm + GLUDym - ASPTA - ILETA - LEUTA - VALTA
    {'Aspartate'              }    {'ARGSS + ASPTA >= 1.1925'
    {'Serine'                 }    {'GHMT2r + r0060 >= 0.8625'
    {'Arginine'               }    {'GLYAMDTRc + r0145 + ARGN >= 0.7245'
    {'Tyrosine'               }    {'TYR3MO2 + TYRTA + HMR_6728 + HMR_6874 >= 0.55875'
    {'Histidine'              }    {'HISDC + HISD >= 1.095'
    {'Leucine + Isoleucine'   }    {'ILETA + LEUTA >= 1.305'
    {'Valine + Methionine'    }    {'METAT + VALTA >= 0.705'
    {'Glycine'                }    {'GTHS - GHMT2r >= 0.7725'


Feasible model after adding coupling constraints.

--------------------------------------------------------------

Removing 228 reactions ...

The following reaction(s) to be removed is(are) not in the model:
    {'HMR_biomass_Renalcancer'   }
    {'DM_HMR_biomass_renalcancer'}
    {'biomass_components'        }
    {'EX_ser_D[e]'               }
    {'EX_pro_D[e]'               }
    {'HMR_1708'                  }
    {'HMR_1934'                  }
    {'r0947'                     }
    {'r1431'                     }
    {'r1432'                     }
    {'RE1096R'                   }
    {'RE1134R'                   }
    {'RE2117M'                   }
    {'RE2768R'                   }
    {'RE2782C'                   }
    {'RE3111M'                   }
    {'RE3338C'                   }
    {'RE3340C'                   }
    {'RE3448C'                   }
    {'RE3564C'                   }
    {'RE3627C'                   }
    {'DM_adchac[c]'              }
    {'DM_alchac[c]'              }

205 reactions were deemed inactive by reaction bibliomics and removed

Feasible model after removing inactive reactions.
```

```
------------------------------------------------------------

Removing inactive genes

57 active genes from the omics data have been manually assigned as inactive genes and will be discarded fr
    {'100137049'}
    {'100526794'}
    {'10060'    }
    {'10165'    }
    {'10858'    }
    {'10873'    }
    {'1160'     }
    {'125965'   }
    {'130752'   }
    {'1346'     }
    {'1468'     }
    {'1583'     }
    {'1588'     }
    {'1607'     }
    {'170712'   }
    {'206358'   }
    {'2110'     }
    {'240'      }
    {'246213'   }
    {'2571'     }
    {'26227'    }
    {'2645'     }
    {'27165'    }
    {'2747'     }
    {'2752'     }
    {'2820'     }
    {'3099'     }
    {'3101'     }
    {'341947'   }
    {'349565'   }
    {'366'      }
    {'374291'   }
    {'3767'     }
    {'412'      }
    {'43'       }
    {'5053'     }
    {'5106'     }
    {'548596'   }
    {'57084'    }
    {'5834'     }
    {'622'      }
    {'64802'    }
    {'6505'     }
    {'6529'     }
    {'6531'     }
    {'6538'     }
    {'6571'     }
    {'6581'     }
    {'6582'     }
    {'6818'     }
    {'6833'     }
    {'7054'     }
    {'79751'    }
    {'83733'    }
    {'84889'    }
    {'8659'     }
    {'9481'     }
```

```
11 inactive genes are not in the model to be removed.
Infeasible model after temporarily closing reactions corresponding to inactive genes, relaxing.

8 reaction(s) were not deleted based on inactive genes as their removal would cause the model to be infeas
1753 reactions were deleted or constrained based on the inactive genes (that do not affect core reactions)
480 genes were specified as inactive but not removed as they are involved in reactions that may be cataly

Feasible model after removing inactive genes (that do not affect core reactions).

------------------------------------------------------------------

Generating model without an objective function.

------------------------------------------------------------------

Identifying flux consistent reactions ...

12 core reactions are not in the model:
    {'DM_ca2[c]'    }
    {'DM_clpn_hs[c]'}
    {'EX_HC01944[e]'}
    {'EX_adpcbl[e]' }
    {'EX_ca2[e]'    }
    {'EX_cl[e]'     }
    {'EX_mg2[e]'    }
    {'EX_selni[c]'  }
    {'EX_zn2[e]'    }
    {'Htmi'         }
    {'Pcm'          }
    {'RE1917C'      }

5709 x 8650 stoichiometric matrix, before flux consistency.
3356 flux inconsistent reactions that are not core reactions,  prior to createTissueSpecificModel.
40 core reactions that are flux inconsistent prior to createTissueSpecificModel.
    Closed_reaction              Name              lb    ub        equation
    _______________    ____________________________    __    __    _______________

    {'EX_icit[e]'}     {'Exchange of Isocitrate'}    0     0     {'icit[e]  -> '}

     Forward_Reaction                                  Name                                                  lb      ub
    ___________________    ____________________________________________________________________________    ___    _______

    {'ACHEe'          }    {'Acetylcholinesterase'                                                    }    0      100000
    {'APOC_LYS_BTNPm'}    {'Proteolysis of ApoC-Lys-Biotin, Mitochondrial'                           }    0      100000
    {'ARGNm'          }    {'Arginase, Mitochondrial'                                                 }    0      100000
    {'CLS_hs'         }    {'Cardiolipin Synthase (Homo Sapiens)'                                     }    0      100000
    {'DURIK1m'        }    {'Deoxyuridine Kinase (ATP:Deoxyuridine), Mitochondrial'                   }    0      100000
    {'EX_co[e]'       }    {'Exchange of Carbon Monoxide '                                            }    0       10000
    {'H2CO3Dm'        }    {'Carboxylic Acid Dissociation'                                            }    0      100000
    {'OCOAT1m'        }    {'3-Oxoacid Coa-Transferase'                                               }    0      100000
    {'P45011A1m'      }    {'Cytochrome P450 11A1, Mitochondrial [Precursor]'                         }    0      100000
    {'P45027A11m'     }    {'5-Beta-Cholestane-3-Alpha, 7-Alpha, 12-Alpha-Triol 27-Hydrox'}           0      100000
    {'P45027A14m'     }    {'5-Beta-Cytochrome P450, Family 27, Subfamily A, Polypeptide '}           0      100000
    {'RBK_D'          }    {'D-Ribulokinase'                                                          }    0      100000
    {'SARDHm'         }    {'Sarcosine Dehydrogenase, Mitochondrial'                                  }    0      100000
    {'STS1'           }    {'Steryl-Sulfatase'                                                        }    0      100000
    {'r0321'          }    {'Acetoacetate:Coa Ligase (AMP-Forming)'                                   }    0      100000
    {'ARGN'           }    {'Arginase'                                                                }    0      100000
    {'RBK'            }    {'Ribokinase'                                                              }    0      100000
    {'DOPAOQNOX'      }    {'Dopamine-O-Quinone Oxidase'                                              }    0      100000
    {'EX_CE2172[e]'   }    {'Exchange of 6, 7-Dihydroxy-1, 2, 3, 4-Tetrahydroisoquinoline'}           0      100000
    {'HMR_9726'       }    {'5-Formyltetrahydrofolate:L-Glutamate N-Formiminotransferase' }           0      100000

    Reverse_Reaction                    Name                              lb      ub
```

{'EX_dlnlcg[e]'}	{'Exchange of Dihomo-Gamma-Linolenic Acid (N-6) '}	-100	0	{'dlnlcg[e]
{'EX_lnlnca[e]'}	{'Exchange of Alpha-Linolenic Acid '}	-100	0	{'lnlnca[e]
{'EX_lnlncg[e]'}	{'Exchange of Gamma-Linolenic Acid '}	-100	0	{'lnlncg[e]
{'r0245' }	{'Glycerol:NADP+ Oxidoreductase'}	-10000	0	{'nadp[c] +

Reversible_Reaction	Name	ll
{'EX_i[e]' }	{'Exchange of Iodide '}	
{'RE1530M' }	{'Thymidine Kinase'}	
{'RE2130C' }	{'RE2130C'}	
{'C02712tm' }	{'Transport of N-Acetylmethionine, Intracellular'}	
{'ACGLUtm' }	{'Transport of N-Acetyl-L-Glutamate, Mitochondrial'}	
{'r2535m' }	{'Transport of L-Homoserine, Mitochondrial'}	
{'EX_k[e]' }	{'Exchange of Kalium'}	-733.70
{'EX_na1[e]' }	{'Exchange of Sodium'}	-17903.
{'EX_pnto_R[e]'}	{'Exchange of (R)-Pantothenate '}	-1.0113
{'EX_hxan[e]' }	{'Exchange of Hypoxanthine '}	-1.1620
{'EX_thm[e]' }	{'Exchange of Thiamin'}	-1.4153
{'EX_pydxn[e]' }	{'Exchange of Pyridoxine'}	-2.2664
{'EX_btn[e]' }	{'Exchange of Biotin '}	-0.0011089
{'EX_CE1310[e]'}	{'Exchange of N-Acetyl-L-Cysteine'}	-0.
{'CE2172t' }	{'Transport of 6, 7-Dihydroxy-1, 2, 3, 4-Tetrahydroisoquinolin'}	

2614 flux inconsistent metabolites that are not core metabolites, prior to createTissueSpecificModel.
10 core metabolites that are flux inconsistent prior to createTissueSpecificModel.

mets	metNames
{'Tyr_ggn[c]' }	{'Tyr-194 Of Apo-Glycogenin Protein (Primer For Glycogen Synthesis)'}
{'pre_prot[r]' }	{'Glycophosphatidylinositol (Gpi)-Anchored Protein Precursor'}
{'retfa[c]' }	{'Fatty Acid Retinol'}
{'thm[m]' }	{'Thiamin'}
{'no2[c]' }	{'Nitrite'}
{'CE1273[c]' }	{'5Beta-Cholestane-3Alpha,7Alpha,12Alpha,24S,25-Pentol'}
{'pail35p_hs[n]'}	{'1-Phosphatidyl-1D-Myo-Inositol 3,5-Bisphosphate'}
{'c101coa[c]' }	{'Decenoyl Coenzyme A'}
{'6hddopaqn[c]' }	{'6-Hydroxydopamine-Quinone'}
{'gm1_hs[n]' }	{'Ganglioside Gm1'}

3085 x 5254 stoichiometric matrix, after flux consistency.

Identifying thermodynamically flux consistent subset ...

22 forced internal reactions, assumed to be external reactions while testing for thermodynamic feasibility
--- findThermoFluxConsistentSubset START ----

```
               formulation: 'pqzw'
                   epsilon: 1e-06
                printLevel: 1
                      nMax: 20
               relaxBounds: 0
        acceptRepairedFlux: 1
           iterationMethod: 'random'
           warmStartMethod: 'random'
           thetaMultiplier: 1.5
                     theta: 0.5
           regularizeOuter: 0
    thermoConsistencyMethod: 'cycleFreeFlux'
                  bigNum: 10000
```

```
                    debug: 0

optCardThermo objective data:
         0.1 = beta, the global weight on one-norm of internal reaction rate.
          -5 = min(g0), the local weight on zero-norm of internal reaction rate.
          -0 = max(g0), the local weight on zero-norm of internal reaction rate.
           0 = min(h0), the local weight on zero-norm of metabolite production rate.
           0 = max(h0), the local weight on zero-norm of metabolite production rate.

optimizeCardinality objective data:

0 min cardinality variables:
         NaN mean(c(p))            NaN min(c(p))            NaN max(c(p))
           1 lambda0             NaN min(k)             NaN max(k)
           1 lambda1             NaN min(o(p))             NaN max(o(p))

2630 max cardinality variables:
          -0 mean(c(q))              -0 min(c(q))              -0 max(c(q))
           1 delta0                5 min(d)                5 max(d)
           0 delta1                0 min(o(q))                0 max(o(q))

11650 cardinality free variables:
       0.077 mean(c(r))              -0 min(c(r))             0.1 max(c(r))
           0 alpha1                0 min(o(r))                0 max(o(r))

 itn     theta    ||dx||     del_obj         obj      linear      ||x||0        a(x)      ||x||1      ||y
   1      0.50  4.2669e+06    -4.5e+06    -4.7e+03       1e+03           0           0           0       -6
   2      0.75       829.04      -7e+02    -5.4e+03     7.4e+02           0           0           0       -6
   3      1.12       274.42    -2.6e+02    -5.7e+03     5.9e+02           0           0           0       -6
   4      1.69       157.04    -1.9e+02    -5.8e+03     4.9e+02           0           0           0       -6
   5      2.53       142.13    -1.3e+02      -6e+03     4.2e+02           0           0           0       -6
   6      3.80       123.52         -98    -6.1e+03     3.8e+02           0           0           0       -6
   7      5.70       47.226         -77    -6.2e+03     3.5e+02           0           0           0       -6
   8      8.54       57.981         -72    -6.2e+03     3.3e+02           0           0           0       -6
   9     12.81       21.274         -38    -6.3e+03     3.2e+02           0           0           0       -6
  10     19.22       155.61         -36    -6.3e+03     3.1e+02           0           0           0       -6
  11     28.83       9.4587         -22    -6.3e+03     3.1e+02           0           0           0       -6
  12     43.25       6.6252         -16    -6.3e+03     3.1e+02           0           0           0       -6
  13     64.87       3.8855         -16    -6.4e+03       3e+02           0           0           0       -6
  14     97.31       2.5721         -23    -6.4e+03       3e+02           0           0           0       -6
  15    145.96       53.976         -24    -6.4e+03       3e+02           0           0           0       -6
  16    218.95       1.2543        -7.5    -6.4e+03       3e+02           0           0           0       -6
  17    328.42      0.79786         -11    -6.4e+03       3e+02           0           0           0       -6
 itn     theta    ||dx||     del_obj         obj      linear      ||x||0        a(x)      ||x||1      ||y
    Optimise cardinality reached the stopping criterion. Finished.
100.00% thermodynamically feasible internal fluxes (checked by cycleFreeFlux method).
               iter            card(y)              nz      %feas int.nz.   tot %feas int.nz.    tot
                  1               2630            1619               1.00                0.31
                  2               1795            2014               1.00                0.58
                  3               1166            1442               1.00                0.70
                  4                835             988               1.00                0.75
                  5                697             840               1.00                0.78
                  6                618             797               1.00                0.81
                  7                534             753               1.00                0.83
                  8                487             649               1.00                0.84
                  9                480             672               1.00                0.85
                 10                452             613               1.00                0.86
                 11                425             615               1.00                0.86
                 12                385             588               1.00                0.87
                 13                363             613               1.00                0.87
                 14                393             611               1.00                0.88
                 15                365             588               1.00                0.88
                 16                393             591               1.00                0.88
                 17                371             583               1.00                0.88
```

```
                 18                350                583               1.00               0.89
                 19                321                580               1.00               0.89
                 20                339                578               1.00               0.89
               iter            card(y)                 nz       %feas int.nz.   tot %feas int.nz.   tot
findThermoConsistentFluxSubset terminating early: n = nMax = 20
--- findThermoFluxConsistentSubset END ----
36 core reactions that are thermodynamically flux inconsistent prior to createTissueSpecificModel.
371 thermo flux inconsistent metabolites that are not core metabolites,  prior to createTissueSpecificMode
    {'13_cis_retn[r]'      }
    {'13dampp[e]'          }
    {'2h3mv[c]'            }
    {'2h3mv[e]'            }
    {'2hiv[c]'             }
    {'2hiv[e]'             }
    {'34dhpha[e]'          }
    {'34hpl[e]'            }
    {'34hpp[m]'            }
    {'35cgmp[e]'           }
    {'4hbz[m]'             }
    {'4hbzcoa[m]'          }
    {'4hphac[e]'           }
    {'5cysdopa[e]'         }
    {'5mta[e]'             }
    {'C05767[c]'           }
    {'C05767[e]'           }
    {'C05957[r]'           }
    {'C09642[c]'           }
    {'C09642[e]'           }
    {'CE0713[m]'           }
    {'CE0849[m]'           }
    {'CE1261[e]'           }
    {'CE1617[r]'           }
    {'CE1935[e]'           }
    {'CE1936[e]'           }
    {'CE1940[e]'           }
    {'CE2176[e]'           }
    {'CE2245[c]'           }
    {'CE2249[c]'           }
    {'CE2253[c]'           }
    {'CE2421[m]'           }
    {'CE2434[m]'           }
    {'CE2705[n]'           }
    {'CE2866[c]'           }
    {'CE2870[c]'           }
    {'CE2872[c]'           }
    {'CE2873[c]'           }
    {'CE2874[c]'           }
    {'CE2875[c]'           }
    {'CE4810[c]'           }
    {'CE4811[c]'           }
    {'CE4812[c]'           }
    {'CE4834[c]'           }
    {'CE4844[c]'           }
    {'CE4845[c]'           }
    {'CE4846[c]'           }
    {'CE4847[m]'           }
    {'CE4848[c]'           }
    {'CE4849[c]'           }
    {'CE4850[c]'           }
    {'CE4851[c]'           }
    {'CE4852[c]'           }
    {'CE4853[c]'           }
    {'CE4854[m]'           }
    {'CE4876[c]'           }
```

```
{'CE4876[r]'           }
{'CE4877[c]'           }
{'CE4890[c]'           }
{'CE4890[e]'           }
{'CE4987[c]'           }
{'CE5025[e]'           }
{'CE5026[e]'           }
{'CE5156[c]'           }
{'CE5157[c]'           }
{'CE5158[c]'           }
{'CE5160[c]'           }
{'CE5161[c]'           }
{'CE5162[c]'           }
{'CE5236[c]'           }
{'CE5236[n]'           }
{'CE5626[c]'           }
{'CE5629[c]'           }
{'CE5629[e]'           }
{'CE7079[c]'           }
{'CE7083[c]'           }
{'CE7083[e]'           }
{'HC01118[c]'          }
{'HC01118[r]'          }
{'HC01361[c]'          }
{'HC01361[e]'          }
{'HC01496[m]'          }
{'HC02121[c]'          }
{'HC02136[c]'          }
{'HC02205[c]'          }
{'HC02207[c]'          }
{'M00261[c]'           }
{'M00261[m]'           }
{'M00263[c]'           }
{'M00263[m]'           }
{'N1aspmd[c]'          }
{'N1aspmd[e]'          }
{'Rtotal3coa[m]'       }
{'Rtotal3crn[c]'       }
{'Rtotal3crn[m]'       }
{'Ser_Gly_Ala_X_Gly[c]'}
{'Ser_Gly_Ala_X_Gly[g]'}
{'Ser_Gly_Ala_X_Gly[l]'}
{'Sfglutth[c]'         }
{'Ssq23epx[c]'         }
{'T4hcinnm[e]'         }
{'adprbp[c]'           }
{'adprbp[e]'           }
{'adrncoa[m]'          }
{'aicar[e]'            }
{'alpam[m]'            }
{'alpro[m]'            }
{'ametam[e]'           }
{'amp[g]'              }
{'amp[x]'              }
{'arachd[l]'           }
{'argalaphe[c]'        }
{'argalaphe[e]'        }
{'arghisthr[c]'        }
{'arghisthr[e]'        }
{'argleuphe[c]'        }
{'argleuphe[e]'        }
{'argphearg[c]'        }
{'argphearg[e]'        }
{'argtyrval[c]'        }
```

```
{'argtyrval[e]'        }
{'argvaltrp[c]'        }
{'argvaltrp[e]'        }
{'asnphecys[c]'        }
{'asnphecys[e]'        }
{'asntyrphe[c]'        }
{'asntyrphe[e]'        }
{'asphispro[c]'        }
{'asphispro[e]'        }
{'c226coa[m]'          }
{'camp[e]'             }
{'camp[g]'             }
{'chsterol[l]'         }
{'clpndcoa[m]'         }
{'core5[c]'            }
{'core5[e]'            }
{'core5[g]'            }
{'core7[c]'            }
{'core7[e]'            }
{'core7[g]'            }
{'core8[c]'            }
{'core8[e]'            }
{'core8[g]'            }
{'coucoa[m]'           }
{'cysglutrp[c]'        }
{'cysglutrp[e]'        }
{'dedol_L[c]'          }
{'dedoldp_L[c]'        }
{'dedolp_L[c]'         }
{'dhbpt[e]'            }
{'dhlpro[m]'           }
{'dolichol_L[c]'       }
{'dolichol_L[e]'       }
{'dopa3glcur[c]'       }
{'dopa3glcur[e]'       }
{'dopa4glcur[c]'       }
{'dopa4glcur[e]'       }
{'dopa4sf[c]'          }
{'dopa4sf[e]'          }
{'dopasf[c]'           }
{'dopasf[e]'           }
{'dxtrn[c]'            }
{'eicostetcoa[m]'      }
{'estrone[r]'          }
{'estrones[r]'         }
{'f1a[e]'              }
{'f1a[g]'              }
{'f1a[l]'              }
{'fad[e]'              }
{'fdp[e]'              }
{'galgluside_hs[g]'    }
{'glnhishis[c]'        }
{'glnhishis[e]'        }
{'glumethis[c]'        }
{'glumethis[e]'        }
{'glutrpala[c]'        }
{'glutrpala[e]'        }
{'glygn3[c]'           }
{'glyphe[c]'           }
{'hisargcys[c]'        }
{'hisargcys[e]'        }
{'hisargser[c]'        }
{'hisargser[e]'        }
{'hisglnala[c]'        }
```

```
{'hisglnala[e]'        }
{'hisglugln[c]'        }
{'hisglugln[e]'        }
{'hisglylys[c]'        }
{'hisglylys[e]'        }
{'hishislys[c]'        }
{'hishislys[e]'        }
{'hismet[c]'           }
{'hismet[e]'           }
{'hisphearg[c]'        }
{'hisphearg[e]'        }
{'hisprolys[c]'        }
{'hisprolys[e]'        }
{'histrphis[c]'        }
{'histrphis[e]'        }
{'hmcr[c]'             }
{'hmcr[e]'             }
{'ibup_R[c]'           }
{'ibup_R[e]'           }
{'id3acald[m]'         }
{'iletrptyr[c]'        }
{'iletrptyr[e]'        }
{'imp[e]'              }
{'ind3ac[m]'           }
{'ins[e]'              }
{'lald_L[m]'           }
{'leutrp[c]'           }
{'leutrp[e]'           }
{'leutrparg[c]'        }
{'leutrparg[e]'        }
{'leutyrtyr[c]'        }
{'leutyrtyr[e]'        }
{'lpro[m]'             }
{'lystrparg[c]'        }
{'lystrparg[e]'        }
{'metglntyr[c]'        }
{'metglntyr[e]'        }
{'metphearg[c]'        }
{'metphearg[e]'        }
{'mettrpphe[c]'        }
{'mettrpphe[e]'        }
{'n8aspmd[c]'          }
{'n8aspmd[e]'          }
{'normete_L[c]'        }
{'normete_L[e]'        }
{'nrvnccoa[c]'         }
{'nrvnccoa[x]'         }
{'odecoa[x]'           }
{'pac[e]'              }
{'pd3[c]'              }
{'phaccoa[c]'          }
{'phacgly[c]'          }
{'phacgly[e]'          }
{'pheacgln[c]'         }
{'pheacgln[e]'         }
{'pheasnmet[c]'        }
{'pheasnmet[e]'        }
{'pheglnphe[c]'        }
{'pheglnphe[e]'        }
{'pheleuasp[c]'        }
{'pheleuasp[e]'        }
{'pheleuhis[c]'        }
{'pheleuhis[e]'        }
{'phelysala[c]'        }
```

```
{'phelysala[e]'        }
{'phephe[c]'           }
{'phephe[e]'           }
{'phepheasn[c]'        }
{'phepheasn[e]'        }
{'phephethr[c]'        }
{'phephethr[e]'        }
{'pheproarg[c]'        }
{'pheproarg[e]'        }
{'phesertrp[c]'        }
{'phesertrp[e]'        }
{'phetrpleu[c]'        }
{'phetrpleu[e]'        }
{'phetyr[c]'           }
{'phetyr[e]'           }
{'phetyrgln[c]'        }
{'phetyrgln[e]'        }
{'phetyrlys[c]'        }
{'phetyrlys[e]'        }
{'phlac[c]'            }
{'phlac[e]'            }
{'prgnlone[c]'         }
{'prgnlone[r]'         }
{'prgnlones[c]'        }
{'prgnlones[r]'        }
{'prohis[c]'           }
{'prohis[e]'           }
{'prohistyr[c]'        }
{'prohistyr[e]'        }
{'prophe[c]'           }
{'prophe[e]'           }
{'prostgd2[r]'         }
{'prostgf2[c]'         }
{'protrpthr[c]'        }
{'protrpthr[e]'        }
{'retinal[r]'          }
{'retinal_cis_13[r]'   }
{'retinal_cis_9[r]'    }
{'retn[r]'             }
{'rsv[r]'              }
{'rsvgluc[r]'          }
{'serargtrp[c]'        }
{'serargtrp[e]'        }
{'sertrphis[c]'        }
{'sertrphis[e]'        }
{'sql[c]'              }
{'sql[e]'              }
{'stcrn[r]'            }
{'thbpt4acam[n]'       }
{'thbpt[e]'            }
{'thrhishis[c]'        }
{'thrhishis[e]'        }
{'thrphearg[c]'        }
{'thrphearg[e]'        }
{'trpalapro[c]'        }
{'trpalapro[e]'        }
{'trpargala[c]'        }
{'trpargala[e]'        }
{'trpglngln[c]'        }
{'trpglngln[e]'        }
{'trpglupro[c]'        }
{'trpglupro[e]'        }
{'trpglutyr[c]'        }
{'trpglutyr[e]'        }
```

```
    {'trpglyphe[c]'        }
    {'trpglyphe[e]'        }
    {'trphismet[c]'        }
    {'trphismet[e]'        }
    {'trpiletrp[c]'        }
    {'trpiletrp[e]'        }
    {'trpmetarg[c]'        }
    {'trpmetarg[e]'        }
    {'trpphe[c]'           }
    {'trpphe[e]'           }
    {'trpprogly[c]'        }
    {'trpprogly[e]'        }
    {'trpproleu[c]'        }
    {'trpproleu[e]'        }
    {'trpproval[c]'        }
    {'trpproval[e]'        }
    {'trpsertyr[c]'        }
    {'trpsertyr[e]'        }
    {'trpthrtyr[c]'        }
    {'trpthrtyr[e]'        }
    {'trptyrgln[c]'        }
    {'trptyrgln[e]'        }
    {'trptyrtyr[c]'        }
    {'trptyrtyr[e]'        }
    {'ttccoa[c]'           }
    {'ttccoa[x]'           }
    {'tym[e]'              }
    {'tyrala[c]'           }
    {'tyrala[e]'           }
    {'tyralaphe[c]'        }
    {'tyralaphe[e]'        }
    {'tyrargglu[c]'        }
    {'tyrargglu[e]'        }
    {'tyrargser[c]'        }
    {'tyrargser[e]'        }
    {'tyrglu[c]'           }
    {'tyrglu[e]'           }
    {'tyrphetyr[c]'        }
    {'tyrphetyr[e]'        }
    {'tyrtrpphe[c]'        }
    {'tyrtrpphe[e]'        }
    {'tyrtyr[c]'           }
    {'tyrtyr[e]'           }
    {'tyrvalmet[c]'        }
    {'tyrvalmet[e]'        }
    {'urate[c]'            }
    {'urate[e]'            }
    {'valphearg[c]'        }
    {'valphearg[e]'        }
    {'valprotrp[c]'        }
    {'valprotrp[e]'        }
    {'valtrpphe[c]'        }
    {'valtrpphe[e]'        }
    {'vanillac[c]'         }
    {'vanillac[e]'         }
    {'vanilpyr[c]'         }
    {'vitd3[c]'            }
    {'xmp[e]'              }
    {'xtsn[e]'             }

No core metabolites that are thermo flux inconsistent prior to createTissueSpecificModel.
0 thermodynamically flux inconsistent forward reactions with  DrGtMean > 0 and DrGtError < abs(DrGtMean).
2714 x 4604 stoichiometric matrix, after thermodynamic flux consistency.
211 active genes not present in model.genes, so they are ignored.
```

```
------------------------------------------------------------------

Creating tissue specific model ...

--- thermoKernel START ----
thermoKernel parameters:
                     solver: 'thermoKernel'
                    epsilon: 1e-06
                  metWeights: [3412×1 double]
                  rxnWeights: [5302×1 double]
                  printLevel: 1
                 formulation: 'pqzwrs'
                        nMax: 20
                 relaxBounds: 0
          acceptRepairedFlux: 1
             iterationMethod: 'greedyRandom'
    normalizeZeroNormWeights: 0
           activeInactiveRxn: []
            presentAbsentMet: []
           removeOrphanGenes: 1
              nbMaxIteration: 30

            warmStartMethod: 'random'
                formulation: 'pqzwrs'
            thetaMultiplier: 1.5
                      theta: 0.5
            regularizeOuter: 1
                    epsilon: 1e-06
                printLevel: 1
                relaxBounds: 0
         acceptRepairedFlux: 1
    thermoConsistencyMethod: 'cycleFreeFlux'
                     bigNum: 10000
                      debug: 0

optCardThermo objective data:
           1 = beta, the global weight on one-norm of internal reaction rate.
          -1 = min(g0), the local weight on zero-norm of internal reaction rate.
         2.7 = max(g0), the local weight on zero-norm of internal reaction rate.
          -1 = min(h0), the local weight on zero-norm of metabolite production rate.
         2.7 = max(h0), the local weight on zero-norm of metabolite production rate.

optimizeCardinality objective data:

6792 min cardinality variables:
           0 mean(c(p))              -0 min(c(p))              -0 max(c(p)...
--- thermoKernel END ----
4182 reactions removed by createTissueSpecificModel.
50 core reactions removed by createTissueSpecificModel.
2662 metabolites removed by createTissueSpecificModel.
1 core metabolites removed by createTissueSpecificModel.
    {'ahdt[c]'}

Feasible tissue specific model. Done.

------------------------------------------------------------------
750 x 1120 stoichiometric matrix

Feasible at end of XomicsToModel.


------------------------------------------------------------------
Diary written to: /Users/gpreciat/20210907T223113_gpreciat_diary.txt
XomicsToModel run is complete.
```

References

1. German Preciat, Agnieszka B. Wegrzyn, Ines Thiele, et al., "XomicsToModel: a COBRA Toolbox extension for generation of thermodynamic-flux-consistent, context-specific, genome-scale metabolic models", *bioRxiv* (**2021**)
2. Laurent Heirendt, Sylvain Arreckx, Thomas Pfau, et al., "Creation and analysis of biochemical constraint-based models using the COBRA Toolbox v. 3.0", *Nature protocols* (**2019**).
3. German Preciat, Edinson Lucumi Moreno, Agnieszka B. Wegrzyn, et al., "Mechanistic model-driven exometabolomic characterisation of human dopaminergic neuronal metabolism", *bioRxiv* (**2021**)
4. Elizabeth Brunk, Swagatika Sahoo, Daniel C. Zielinski, et al., "Recon3D enables a three-dimensional view of gene variation in human metabolism", *Nature biotechnology* (**2018**)
5. Noronha et al., "The Virtual Metabolic Human database: integrating human and gut microbiome metabolism with nutrition and disease", *Nucleic Acids Research* (**2018**).