Published online 6 April 2015

Nucleic Acids Research, 2015, Vol. 43, No. 8 ***3899–3910***
doi: 10.1093/nar/gkv294

Reconstructing genome-scale metabolic models with *merlin*

Oscar Dias*, Miguel Rocha, Eugénio C. Ferreira and Isabel Rocha*

Centre of Biological Engineering, University of Minho, Campus de Gualtar, 4710–057 Braga, Portugal

Received August 13, 2014; Revised March 14, 2015; Accepted March 18, 2015

ABSTRACT

The Metabolic Models Reconstruction Using Genome-Scale Information (*merlin*) tool is a user-friendly Java application that aids the reconstruction of genome-scale metabolic models for any organism that has its genome sequenced. It performs the major steps of the reconstruction process, including the functional genomic annotation of the whole genome and subsequent construction of the portfolio of reactions. Moreover, *merlin* includes tools for the identification and annotation of genes encoding transport proteins, generating the transport reactions for those carriers. It also performs the compartmentalisation of the model, predicting the organelle localisation of the proteins encoded in the genome and thus the localisation of the metabolites involved in the reactions promoted by such enzymes. The gene-proteins-reactions (GPR) associations are automatically generated and included in the model. Finally, *merlin* expedites the transition from genomic data to draft metabolic models reconstructions exported in the SBML standard format, allowing the user to have a preliminary view of the biochemical network, which can be manually curated within the environment provided by *merlin*.

INTRODUCTION

Genome-scale metabolic models (GSMMs) are used to predict, *in silico*, microorganisms' responses to different genetic or environmental stressors (1–3). The reconstruction and use of these biochemical models is, nowadays, a common alternative to the more expensive and time-consuming wet-lab experiments as the output provided by the *in silico* simulations permits focusing on experiments with promising results. A GSMM allows predicting a given organism's phenotype from its genome sequence and biochemical information. To achieve this purpose, a set of biochemical reactions that can take place within the target organism should be assembled (4,5). Besides the reactions catalysed by enzymes associated with metabolic genes, the crossing of cellular membranes by metabolites is often promoted by transporter proteins also encoded in the genome.

The collection of these reactions is a laborious and iterative process, which was previously described by several authors. The most comprehensive of these studies describes a protocol with about 100 steps (3) that can be summarised in six stages (1), according to Figure 1. In the first stage, information on the genome annotation is retrieved from several data sources. Here, data collected include Enzyme Commission (EC) (6) and Transporter Classification (TC) numbers (7), as well as the associated genes and gene product names, if available. Other data, such as genes associated to signal transduction or gene expression regulation are not considered. Although the information on the genome annotation can be found in public databases for a wide variety of organisms, it should be remarked that, often, these annotations have been performed at the time of genome sequencing, being usually out-dated. Also, the information collected during this process does not always comply with the requirements of a GSMM (e.g. often EC numbers are missing for genes identified as metabolic). Frequently, a re-annotation has to be performed as a first step of the reconstruction process (8–10). The next stage is the identification of the metabolic reactions associated with the organism. Initially, only reactions associated to the EC numbers previously identified should be retrieved. Afterwards, reactions catalysed by enzymes without EC numbers assigned, namely transport reactions and reactions known to exist in a given organism (from experimental evidence described in the literature), are used to complement the GSMM, together with spontaneous reactions.

After the assembly of the reaction set, its stoichiometry should be checked, namely using information available in online databases such as BRENDA (11), BKM-react (12) or MetaCyc (13).

For compartmentalisation, information about the reactions (and corresponding enzymes) localisation should be sought. In prokaryotes, compartments are limited to the cytosol and (often) the periplasmic space. In eukaryotes, reactions can take place in several different compartments,

*To whom correspondence should be addressed. Tel: +351 253 604 421; Fax: +351 253 604 429; Email: odias@deb.uminho.pt
Correspondence may also be addressed to Isabel Rocha. Tel: +351 253 604 414; Fax: +351 253 604 429; Email: irocha@deb.uminho.pt

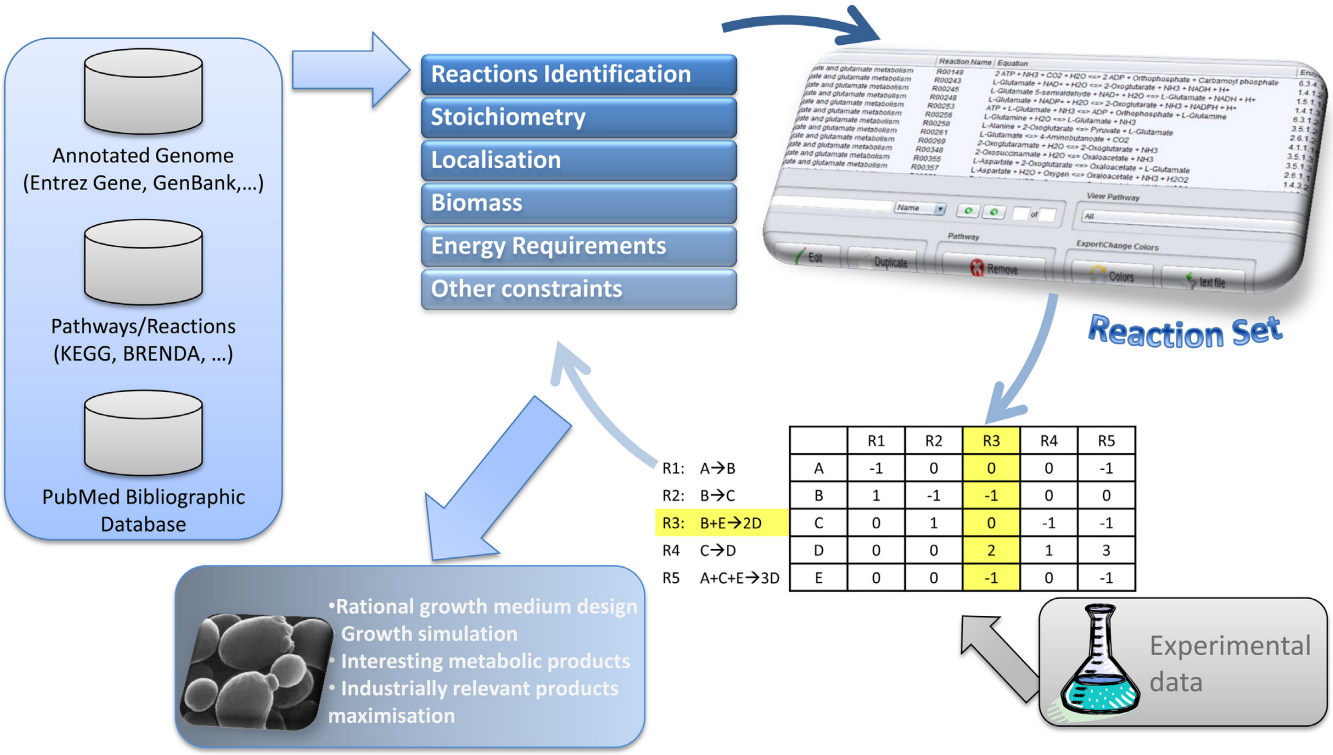

Figure 1. Illustration of the GSMM's reconstruction process. Adapted from Rocha *et al.* (1).

including the mitochondrion, endoplasmic reticulum, or Golgi apparatus.

The reconstruction of a GSMM is not complete without the addition of an equation representing biomass formation, which should denote a drain of building blocks (e.g. amino acids) into the biomass (5). Growth-associated energy requirements (ATP molecules needed per gram of biomass synthesised) are also necessary for inclusion in the biomass equation. The addition of other GSMM constraints includes determining the reversibility of the reactions, defining numeric values for the bounds of uptake reaction fluxes and the non-growth ATP requirements. All of these constraints are important and should be sought in online databases and the literature. After debugging the reaction set, the GSMM simulation results should be validated using appropriate experimental data. This will allow further debugging, improving the simulation results of the GSMM in an iterative loop.

Using this, or a similar approach, several GSMMs were reconstructed since the publication of the *Haemophilus influenza* GSMM (14), which was the first microorganism to have its metabolic model reconstructed. An updated list of these reconstructions can be found at www.optflux.org/models.

The complete reconstruction of a GSMM can take from weeks to over a year (3). However, this process can be greatly accelerated if some steps are automated. The sequence of steps described above involves the utilisation of a disparate number of bioinformatics tools available to the public, usually in different services. These, typically, require the definition of several parameters that have to be optimised and validated for this specific purpose, such as the homology search metrics (expected value, normalised scores or alignment coverage) or protein localisation score thresholds. Also, other specific tools may need to be developed, such as data integration tools.

Many of these steps require subsequent and substantial manual curation and validation and a significant amount of data still needs to be extracted from the literature and manually inserted into the model. Finally, it is important for many model developers to privately manage and control the reconstruction process, a feature only possible with applications running in their own computers.

In summary, in our view, a tool that could greatly accelerate the reconstruction process would need to perform the workflow of the bioinformatics related tasks, including genome (re-)annotation, in an optimised way, while simultaneously allowing to perform manual curation locally without the need for commercial software.

In this context, to face these challenges, we have developed Metabolic Models Reconstruction Using Genome-Scale Information (merlin). This tool can annotate a genome with both enzymatic and transport functions, and build a compartmentalised draft GSMM, with minimum user interaction, in less than a week, depending on genome size. It also provides a user-friendly interface to perform manual curation of the draft model at any stage.

Various software tools have been developed and databases have been assembled to help on the reconstruction process. Some features of *merlin* can also be found in repositories and web applications such as FAME (15), MEMOSys (16), MicrobesFlux (17), the Pathway Tools

(18), CoReCo (19), RAVEN (20), or Model SEED (21). Nevertheless, none complies with the full set of requirements described above. Table 1 shows the main capabilities of each of these tools, highlighting the differences between these applications.

Currently, to the best of our knowledge, *merlin* is the only tool that provides an integrated framework for the reconstruction of GSMM for both prokaryotes and eukaryotes that retrieves enzymatic, transport and localisation information from the genome.

The first four frameworks described in Table 1 use previously annotated genomes, thus not allowing metabolic (re-) annotations, which can be important to unambiguously define the reactions that will be added to the GSMM. The RAVEN toolbox performs genome-wide functional annotations, using template models or KEGG as source for homology alignments. However, this toolbox does not perform the annotation of transporter genes and the corresponding reactions. Also, RAVEN requires the commercial MatLab® software to run. Although not having a proprietary database for metabolic data, FAME does not provide any operation to verify if the stoichiometric reactions are balanced. Model SEED and Pathways Tool have internal databases for metabolic data; thus it is assumed that their reactions are balanced. Most of these applications also have a tool that identifies reactions with dead-end metabolites. CoReCo does not perform compartmentation of the model nor identifies transporters and gene-protein-reaction (GPR) associations; yet, it allows performing comparative reconstructions of GSMMs, exporting these to SBML. The interaction with this platform is performed via Python™ scripts.

Although Model SEED offers nearly the same features that *merlin* provides, there are some significant differences. For one, the curation of the annotation in Model SEED is performed by expert curators within the SEED project. This is clearly an advantage for users that just need the draft model or that use it as the starting point for further developments. However, the submission of a genome for annotation involves sharing the user's data with the SEED's web server, which might not be desirable to some researchers. Contrarily, *merlin* provides a semi-automatic annotation, with confidence scores, which can be privately curated by the user. Also, both applications have structural differences, such as the origin of the metabolic information used to develop the GSMM, which is the KEGG database (23) in *merlin* and an internally developed database in Model SEED. Lastly, one of the major differences between both tools is that *merlin* performs the reconstruction of eukaryotic GSMMs, which is not currently supported by the Model SEED.

One of *merlin*'s important features is the '*Reactions Viewer*', which allows visualising all reactions present in the model, as well as reactions not yet associated, grouped by pathway. The 'Draw in Browser' button, within this panel, aids the user in the gap filling process by showing enzymes and reactions annotated by *merlin* directly in the selected KEGG pathway browser.

MATERIALS AND METHODS

Specifications and architecture

merlin is an open-source application, currently available for Linux and Windows. It is distributed under the GNU General Public License at the website http://www.merlin-sysbio.org. The application is fully implemented in Java™, which was chosen since it is a widely used platform-independent programming language. *merlin* was built on top of the AIBench (http://www.aibench.org) software development framework (24). The design principles and architecture of this framework allow reusing and combining components. Applications developed on top of this framework incorporate three types of well-defined objects: operations, datatypes and datatype views, following the MVC (model-view-controller) software architecture pattern.

merlin uses several Java libraries to access web services, namely BioJava (25), NCBI Utilities Web Service Java Application Programming Interface (API), UniProtJAPI (26), the ChEBI Java API, the KEGG Representational State Transfer (REST) API and jSBML (27), among others.

The open source MySQL® relational database is used for the local data repository. MySQL schemas (given in Supplementary Data) were prepared to allow the further development of the framework, already including table structures still unused by the current version (e.g. tables prepared to store regulatory and experimental data).

merlin databases

Several public databases are used by *merlin* to collect data for the development of GSMMs. A brief description of each database is available in Supplementary Table S1.1 (supplemental file 1 of the supplementary data).

Methods and algorithms

merlin features four main independent modules: the *Load internal database*, the *Enzymes annotation*, the *Transporters annotation* and the *Compartments prediction* modules (Figure 2). A brief description of each module is performed next.

Initially, the genome file(s) for the target organism, in the FASTA format, is uploaded to *merlin*. Genomes retrieved from NCBI's FTP website are automatically processed. The input of other genomes requires the introduction of the taxonomy ID, retrieved from the NCBI Taxonomy website (http://www.ncbi.nlm.nih.gov/taxonomy). Usually, the organism species identifier is used, but when the species is unknown, an identifier for another taxonomic branch can be used, for instance the genre identifier. Both amino acid files (*.faa) and nucleotide files (*.fna) can be loaded.

Load internal database. This module is used to retrieve and load the initial set of metabolic data (metabolites, enzymes and reactions) into *merlin*'s internal model. Several types of data are retrieved from KEGG by this module, including Compounds, Glycans, Drugs, Reactions, Modules, Pathways and Enzymes. Afterwards, *merlin* saves this information and builds a local database, according to the schema given by Supplementary Figure S2.1 (supplemental file 2 of the supplementary data). Optionally, genomic information

Table 1. Comparison of the features of software tools developed for aiding the reconstruction of genome-scale metabolic models

Software	FAME	MEMOSys	MicrobesFlux	Pathway Tools	CoReCo	RAVEN	Model SEED	*merlin*
Enzymes annotation					•	•	•	•
Transporters annotation				•			•	•
Compartments prediction	*i*	*i*				•	•	•
Biomass reaction	*ii*		*ii*		*ii*	*ii*	•	*ii*
Export to SBML	•	•	•		•	•	•	•
Runs locally		•		•	•	•		•
Requires commercial software						•		
Graphical interface for manual curation				•				•
Pathways visualisation	•		•	•		•	•	•
Gene-Protein-Reaction rules							•	•
Highlight metabolic dead-ends	•		•	•		•		•
Reactions stoichiometry validation		*iii*	•	*iii*	•		*iii*	•
Prokaryotic models	•	•	•	•	•	•	•	•
Eukaryotic models				•	•	•		•

[*i*] Allow to manually assign compartments to reactions (*merlin,* RAVEN and Model SEED automatically predict reactions localisation).
[*ii*] Biomass reaction inserted manually (Model SEED - Biomass reaction automatically generated).
[*iii*] Model SEED and Pathways tools use their own metabolic databases. MicrobesFlux checks for new reactions.
* SBML - Systems Biology Markup Language (22).

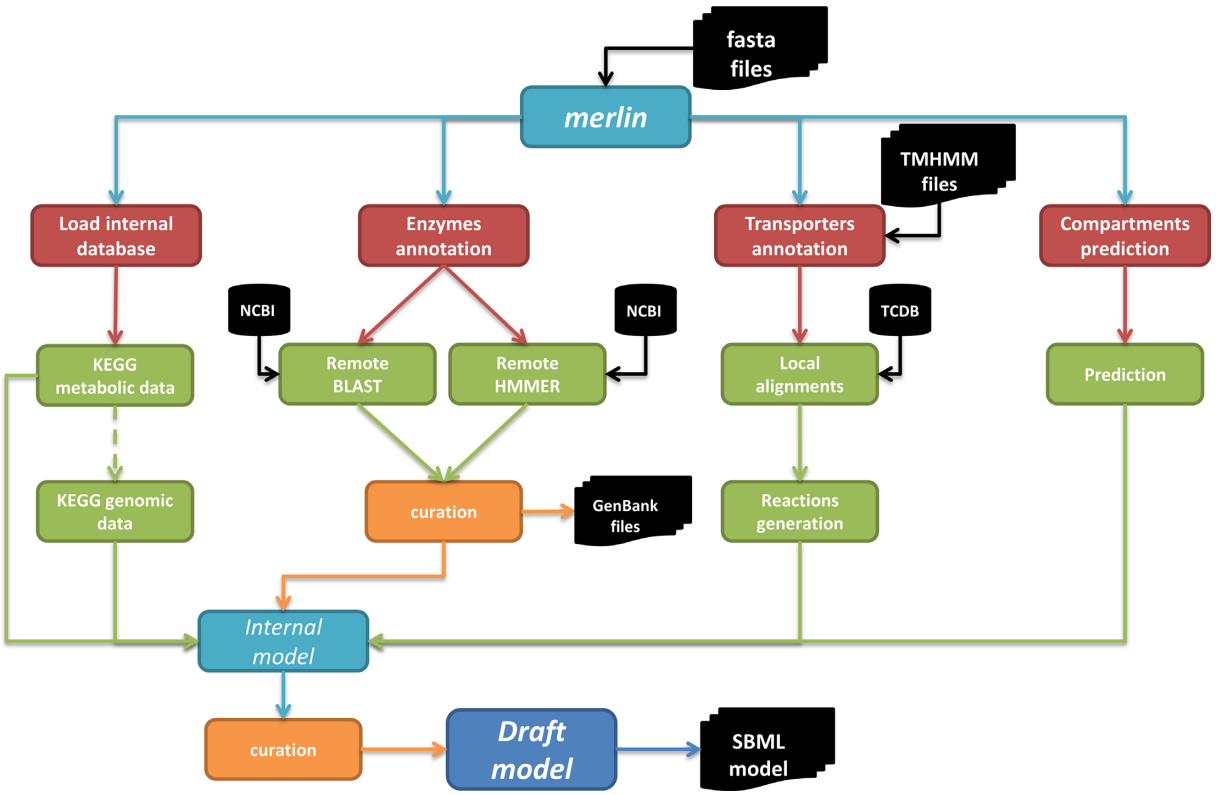

Figure 2. Schematic representation of *merlin*'s architecture.

for organisms annotated in KEGG Genes may also be retrieved by this module. The GSMM draft can be built using these data only or it can be integrated with the information from other modules, namely the re-annotation results.

Enzymes annotation. The purpose of this module is the assignment of enzymatic functions to proteins encoded in the genome using homology search tools: the Basic Local Alignment Search Tool (BLAST) (28), the profile Hidden Markov Models (HMMER) (29) tool or both.

merlin is able to retrieve data, including species' name and full lineage, for each homologous gene identified in either the BLAST or the HMMER similarity searches from EBI or NCBI. Also, the locus tag gene identifiers for genomes downloaded from the NCBI's FTP website are retrieved.

To that end, each gene is processed individually, and for every homologue identified by the similarity searches (no matter which program and database are used) the retrieved homology data are the following: locus identifier, expected value, score and organism. Afterwards, *merlin* remotely retrieves and collects information from the *NCBI Protein* or the UniProt databases for each of the homologous genes. The downloaded information is the following: taxonomy, organelle (if available), chromosome (if available), locus tag, product (protein name), EC number (if available) and molecular weight. The downloaded information is kept in *merlin*'s MySQL relational local database, assembled according to the data schema in Supplementary Figure S2.2 (supplemental file 2 of the supplementary data).

merlin uses a specific algorithm to assign EC numbers and product names to each gene *g*. The assignments are performed by weighting the number of times each EC number is found within the homologous gene records (frequency) and the taxonomy of the organisms to which such records belong. In the following, we will assume the use of EC numbers in the calculations, while the same is valid for product names. Equation (1) describes how, for each gene *g* the score for a given EC number (ec) is calculated. The weights of the frequency (Score_f) and taxonomy scores (Score_t) are controlled by parameter α:

$$\text{Score}_{\text{g}}^{\text{ec}} = \alpha \times \text{Score}_f + (1-\alpha) \times \text{Score}_{\text{t}} \quad (1)$$

The frequency score, on its turn, calculates the number of occurrences of an EC number within all homologues of that gene. Thus, this score is obtained by counting the number of homologous genes encoding an EC number and dividing by the total number of homologous genes (n), as follows:

$$\text{Score}_f = \frac{\sum_{i=1}^{n} (v_i)}{n} \quad (2)$$

where:

$$v_i = \begin{cases} 1, & \text{if ec number exists in record } i \\ 0, & \text{otherwise} \end{cases}$$

The taxonomy score, on the other hand, is used to weight favourably homologies with records of closely related organisms. As shown in Equation (3), the taxonomy frequency (sum of the number of common taxa between the organism being studied and the ones in the first n homology records) is multiplied by a penalty factor. This penalty decreases the score for EC numbers assigned to a small number of genes, as that may be associated to annotation errors or incorrect assignments. The denominator is calculated by multiplying the maximum taxonomy ($\text{Max}_{\text{Taxonomy}}$) value, which is the number of taxa of the target organism, by the minimum between the number of genes encoding the EC number and the user defined minimal number of homologies ($n_{\text{homologies}}$). This classification allows determining if the first n homology records annotated with a given EC number are closely related to the target organism, taxonomically.

$$\text{Score}_t = \frac{\sum_{i=1}^{n} (t_i \times v_i) \times \text{penalty}_{\text{score}}}{\text{Max}_{\text{Taxonomy}} \times \min\left(\sum_{i=1}^{n} (v_i),\ n_{\text{homologies}}\right)} \quad (3)$$

where t_i is the common taxa count for the organism corresponding to the sequence of hit i.

The $\text{penalty}_{\text{score}}$ is 0 if p times β is equal to or higher than 1; β is a penalty parameter initially set to 0.15, as shown below.

$$\text{penalty}_{\text{score}} = \begin{cases} 0, & (1 - p \times \beta) \leq 0 \\ 1 - p \times \beta, & \text{otherwise} \end{cases} \quad (4)$$

The p calculation is given bellow, being obtained subtracting the frequency of the genes encoding ec from $n_{\text{homologies}}$. If positive, the p penalty is multiplied by β and subtracted to 1. Otherwise, the p penalty is zero, as shown in Equation (5).

$$p = \begin{cases} 0, & \sum_{i=1}^{n} (v_i) \geq n_{\text{homologies}} \\ n_{\text{homologies}} - \sum_{i=1}^{n} (v_i), & \text{otherwise} \end{cases} \quad (5)$$

The α, β and the $n_{\text{homologies}}$ parameters can be directly configured in *merlin*'s *'Homology Data Viewer'*.

The confidence score, with a numeric value between 0 and 1, allows easily curating the EC numbers assigned to a given gene. The user can also define a minimum threshold score value for the automatic approval of annotations. Nevertheless, all annotations can be curated and the automatic assignments changed. The output of this tool is the annotated metabolic genome, which can be integrated into *merlin*'s internal model, exported to a file in the Excel format or integrated into a GenBank file (*.gbk).

Transporters annotation and compartments prediction. Transport reactions are often only included in models if there are evidences supported in experimental data or literature. However, this approach usually originates a very small number of transporters and does not allow performing GPR associations, as often the associated gene is unknown.

Therefore, we propose a new methodology to identify and annotate transport systems that is fully explained in a recently submitted article (Dias, O., Gomes, D.G., Vilaça, P., Cardoso, J., Rocha, M., Ferreira, E.C. and Rocha, I. (2015) Genome-wide Semi-automated Annotation of Transporter Systems. *Submitted*). This methodology automatically annotates carriers with TC family numbers and generates transport reactions for all metabolites transported by these carriers. It is based on the identification and classification of genes that encode transmembrane proteins, as it is assumed that transport proteins are located in membranes (30). Hence, the user must, beforehand, submit the genome amino acid FASTA files to the TransMembrane Prediction using Hidden Markov Models (TMHMM) (31) web server (this tool cannot be remotely accessed) to identify protein encoding genes with transmembrane helices.

Afterwards, *merlin* uses an internal implementation of the Smith-Waterman (SW) algorithm (32) for comparing protein sequences with at least *n* transmembrane helices (being *n* a user defined parameter with a default value of 1) with all protein sequences currently available in the TCDB database. The SW algorithm is used for optimally determining similar regions between two sequences (unlike BLAST that does not assure optimal alignments). This algorithm performs local sequence alignments, comparing segments of all lengths and optimising the similarity measure. The results of the SW similarity search are kept in a relational database according to the schema presented in the Supplementary Figure S2.3 (supplemental file 2 of the supplementary data). This database provides associations between the genome of the organism being studied and TCDB records. These often provide direct access to specific information, namely: UniProt Accession Number, organism, Protein Name, Length and others. However, to date, the substrates and direction of the transport are not directly provided, and, thus, these features have to be inferred from the information provided for each record.

merlin is shipped with a growing database having over 4000 TCDB records already annotated with metabolites and directions. Several databases, namely TCDB, KEGG, ChEBI and the semanticSBML (33) tool were used to assign identifiers to the metabolites transported by each carrier annotated in *merlin*. Although our database does not include all TCDB records, if similarities to un-annotated TCDB records are found, such records can be annotated by the user and uploaded to *merlin*, using a specific operation for that purpose.

Finally, the metabolites transported by each carrier identified in the genome are inferred from the annotations of the TCDB records that have similarities with that carrier. *merlin* uses an internal scorer, based in the schema provided in Supplementary Figure S2.4 (supplemental file 2 of the supplementary data) and similar to the presented above for EC numbers (and product names), to assign metabolites and TC family numbers.

The methodology for the prediction of the subcellular localisation of the proteins and metabolites is supported by WoLF PSORT (34) and PSORTb v3.0 (35) (also described in detail in Dias *et al.* 2015). The information provided by these tools is kept in a relational database, according to the schema presented in Supplementary Figure S2.5 (supplemental file 2 of the supplementary data). The determination of the proteins' localisation in eukaryotic organisms is performed by WoLF PSORT, using a simple remote Java API, provided by Paul Horton in a personal communication. PSORTb 3.0 is used to determine the localisation of proteins in prokaryotic organisms. Unfortunately, unlike WoLF PSORT, PSORTb 3.0 does not provide a web API. In this case, the compartmentalisation data may be retrieved in one of two manners. PSORTb 3.0 offers pre-computed genome results, for genomes deposited in GenBank. These data can be retrieved from the PSORTdb database at http://db.psort.org/browse. Otherwise, the target genome sequence files should be submitted to the PSORTb 3.0 web interface.

The genes are automatically assigned with the main compartment predicted by these programs. Moreover, if alternative compartments have scores that differ by less than a user defined percentage (default value of 10%) from the main compartment, the gene will also be assigned to those compartments.

To annotate transport systems, three criteria have to be met. The first two are that the gene sequences have transmembrane domains and similarities to TCDB records. The third is having a localisation prediction within a membrane. However, the WoLF PSORT and PSORTb 3 lump intracellular membranes with the intracellular organelle predictions, allowing the assignment only to the cytoplasmic membrane or outer membrane for prokaryotes and the plasma membrane for eukaryotes. Therefore, if a sequence meets the first two requirements and WoLF PSORT predicts that the sequence will be assigned to an intracellular organelle, it is considered that such sequence encodes an intracellular transport system. On the other hand, if a protein annotated as a regular enzyme by the other tools is predicted to be within a membrane by WoLF PSORT or PSORTb 3, that enzyme is assigned to the compartments on both sides of the membrane.

These modules allow generating compartment-specific transport reactions, providing associations between genes and reactions, thus allowing the reconstruction of more robust and reliable models. All records provided by TCDB have cross-references to UniProt, thus this identifier is also used as an unambiguous identifier in *merlin*. Moreover, taxonomic information from the TCDB records is retrieved for the classification of the metabolites and the TC family numbers.

Modules integration and SBML model assembly. The integration of the output of the previous modules is easily performed by specific operations within *merlin*, resulting in a fully compartmentalised draft model, which can be curated by the user.

The first module (*Load internal database*) provides data for building the internal model. Moreover, some reactions retrieved from KEGG are automatically integrated into the internal model, such as spontaneous and non-enzymatic reactions. If genomic data are retrieved from KEGG, these are taken into consideration when assembling the internal model. The combination of the output of the enzymes annotation module, with the data retrieved from KEGG, generates a draft GSMM with all the reactions.

Nevertheless, the procedure for the identification of reactions from the genome annotation involved laying out some rules. According to KEGG, enzymes may belong to zero, one or several pathways. This assumption is also true for reactions. Any given enzyme encoded in the genome of an organism should be associated to at least one reaction. Thus, for enzymes that, according to KEGG, catalyse a single reaction, that reaction is automatically added to the model. Likewise, if an enzyme is not present in any KEGG pathway, reactions catalysed by this enzyme will be added to the internal model, and the relevance of these reactions in the model should be assessed by the user. However, if an enzyme catalyses several reactions, only reactions having at least one pathway in common with the enzyme will be added to the internal model. This heuristic is applied to prevent adding too many reactions to the internal model that do not con-

nect to any other reaction. For example, the enzyme alcohol dehydrogenase (EC 1.1.1.1) catalyses 18 reactions; however, 5 of such reactions are not present in any pathway, thus not being added to the internal model when that EC number is identified in the annotation. Nevertheless, the user can manually add/edit/remove any reaction to/from the model.

The determination of the GPR rules can be performed using a unique tool developed within the *merlin* project. This operation retrieves information on the structure of the protein complex modules including their subunits and stoichiometry, from the KEGG BRITE (36) database, generating GPRs for each reaction. For that, information on the protein structure, for all EC numbers selected by the enzymes annotation tool for integration in the model, is retrieved from this KEGG resource. GPR rules for the enzymes included in this database, are provided in text strings that are processed using a grammar specially developed for parsing these data. The GPR rules are defined for sets of genes, with similar roles, and horizontally conserved across several species, the KEGG Orthologues (KO). Each KO in the GPR rule is related to other KOs by AND / OR operators. The KOs are sought in the genome of interest by selecting the sequence of the orthologous gene most closely related, taxonomically, to the case study. Initially, only those genes previously annotated with the EC number of the enzyme to which the rule belongs are used. Yet, if no match is found, the whole genome is sought. If a subunit of a protein complex cannot be found in the genome, the rule is discarded and all genes identified as being involved in that rule are made available for manual curation. The genes associated to enzymatic activities unavailable in the KEGG BRITE database are regarded as distinct copies of the enzyme. Still, the user can manually create rules using any gene. The output of this tool is a set of reactions with GPR rules. The GPR rules are set as notes in the '*Reactions Viewer*' so that users can easily view and edit them.

The Transporters Annotation module provides transport reactions to the internal model, as well as the respective GPR associations. From all reactions generated by this module, only the transport reactions in which the participating metabolites are already present in the network will be included in the internal model, to avoid unconnected reactions. After this stage, the internal model contains reactions taking place in the interior of the cell and transport reactions between the outside and the inside of the cell. The integration of the compartments prediction allows generating a fully compartmentalised draft model. The reconstructed internal model is, at all stages, available in *merlin* for manual curation, where the user can add new reactions, (*e.g.* the biomass reaction) or remove reactions that are not relevant for the model.

merlin includes a tool developed to detect reactions with metabolites unconnected from the network. This tool identifies potential gaps in the model and highlights these reactions. Moreover, reactions downloaded from KEGG might be, even if seldom, unbalanced. Thus, *merlin* also includes an operation to identify potentially unbalanced reactions. Unbalanced reactions impair the model, leading to incorrect predictions.

Finally, the SBML (22) GSMM with Minimum Information Required in the Annotation of Models (MIRIAM) annotations (37) can be exported from *merlin*, so it can be used in many other applications, such as OptFlux (38). MIRIAM annotations are unique identifiers, in the form of Uniform Resource Identifiers (URIs), which provide a way to distinctively characterise data in a given model. The representation of models in the machine-readable SBML format with MIRIAM annotations facilitates comparing, combining and reusing biochemical models.

RESULTS AND DISCUSSION

Operating mode

merlin provides an intuitive and user-friendly interface as depicted in Supplementary Figures S2.6 and S2.7 (supplemental file 2 of the supplementary data). Starting a new project (by choosing *Create Project* from the menu *Project)* involves selecting one MySQL database, as shown in the above mentioned figure. The *Project View* (shown in the panel on the right on Supplementary Figure S2.6) displays important information about the status of the project, such as whether the operations of transporters search or model compartmentalisation have already been performed.

The semi-automatic enzymatic (re-)annotation of an organism's genome is performed by accessing the *Enzymes* menu and clicking the *BLAST annotation* or *HMMER annotation* options. The default configuration of these algorithms is adequate for most purposes. However, most parameters available in these services can be altered within *merlin*. The (re-)annotation process can take from hours to several days, depending on the internet connection and the processing power of the computer running *merlin*, as well as the size of the genome and the availability of the NCBI, EBI or HMMER servers.

After performing the (re-)annotation, *merlin* provides a dedicated view (shown in Figure 3A) for the curation of the enzymes' homology data, the *Homology Data Viewer*. The final annotation reached by *merlin* can be used to update the current GenBank files, by replacing the previous assignments by the new curated annotation.

This visualisation panel depicted in Figure 3A was developed to optimise the manual curation experience. Thus, several features were implemented to facilitate and expedite this process, namely:

- The user can easily select the values for several parameters of the scoring algorithm. The scores are automatically re-calculated and updated, as well as the boxes for selection of the product and EC number, if an EC number has been assigned. Nevertheless, the user can change a pre-selected item, or manually insert a new item.
- The *Info* column allows to access all the information provided by the homology searches, thus providing the user with more information to make a decision.
- The *Status* column is an easy way to determine whether such gene exists (if a star is placed inside the button) and is reviewed (if the star is golden) in UniProt, providing at the same time a direct link to the UniProt entry by clicking the button. Moreover, if the button is green coloured, it means that UniProt's annotation is in agreement with the current EC number selection in *merlin*. Light green means that *merlin* assigns more EC numbers

than UniProt's annotation, but the UniProt assignments are included in *merlin's* annotation. Orange was selected for the cases in which UniProt's annotation includes *merlin* assignments together with other EC numbers. Finally, a red button represents different annotations on *merlin* and UniProt.
- The *Notes* column is useful, for instance, to track changes performed in the annotation during the debugging of the model.
- The *Products* and *EC Number(s)* columns present cross-links to BRENDA and UniProt, accessible clicking the mouse's right button.

The *Transporters* menu can be used to identify transport proteins and generate transport reactions, as well as for loading new transporter annotations and integrating transporters data into the model. The first operation *Transport Proteins Identification* compares genes having transmembrane domains to the protein sequences remotely retrieved from TCDB. The *Transport Reactions Generation* creates transport reactions for metabolites carried by transporters. The third operation can be used to load annotations for TCDB proteins not yet available in *merlin*. The last operation in this menu (*Transporters Integration*) integrates the transporters GPR information into *merlin*'s internal model.

The compartments prediction is handled differently for eukaryotes and prokaryotes in *merlin*. For prokaryotes, the HTML files retrieved from the PSORTb web interface should be loaded using the operation *Load PSORTb v3.0 Results* from the *Compartments* menu. Then, the operation *Perform Compartments Prediction* should be performed. For eukaryotes, the first step is skipped because the second operation retrieves the results from WoLF PSORT remotely. After the compartments prediction, the results should be integrated in the internal model, generating a fully compartmentalised draft model.

Performing the enzymatic (re-)annotation of a genome allows exporting the annotation or integrating it into *merlin*'s internal model. Similarly, the generation of transport reactions, and the compartmentalisation of the model imply the presence of metabolic data previously loaded. Retrieving metabolic data from KEGG involves accessing the *Database* menu and selecting *Load KEGG Data*. If KEGG has its own annotation for the target genome, such annotation can also be retrieved. Being so, the enzymatic (re-)annotation performed within *merlin* is integrated with KEGG's annotation and the internal model is assembled using both annotations.

Several panels were developed for the visualisation and editing of the KEGG data associated with a given metabolic model, namely, the *Genes Viewer*, the *Proteins Viewer*, the *Metabolites Viewer*, the *Reactions Viewer* and the *Pathways Viewer*. The *Proteins Viewer* includes a sub-viewer for the visualisation of information for enzymes, the *Enzymes Viewer*. Likewise, the *Metabolites Viewer* comprehends a couple of sub-viewers: the *Reactants/Products Viewer* (Supplementary Figure S2.6) and the *Compounds/Reactions Viewer*. The first sub-viewer is a fast and easy way to check if a metabolite is a reactant, a product or if it can have both roles in the network. The second sub-viewer is used to determine in which reactions a given metabolite participates.

One of the most relevant panels in *merlin* is probably the *Reactions viewer*, shown in Figure 3B. This allows the user to perform the curation of the GSMM. The panel shows reactions grouped per pathway (thus the repetition of reactions is not uncommon) with different automatically sorted colours in each pathway. This panel allows visualising reactions in all pathways or to select just a specific pathway. When a KEGG pathway is selected, the '*Draw in Browser'* button becomes active (as shown in Figure 3B). This button opens the homepage of the selected KEGG Pathway map, in the default internet browser, and 'paints' all enzymes and reactions, included in the internal model, which belong to that pathway. This feature, together with the *Find unconnected reactions in the network* operation, which paints in red these reactions names and descriptors, allows easily finding gaps in the model. The '*Find unbalanced reactions in the network'* operation is also very useful for finding and labelling stoichiometrically unbalanced reactions within this view. In this case, the reaction name is bolded and italicised.

When the integration of the transporters annotation is performed, a surrogate pathway is created by *merlin*, the *Transporters Pathway* including all transport reactions that met the integration criteria for being inserted in the model. After performing the integration of the compartments data, spontaneous and other reactions not associated to genes are automatically assigned to the internal compartment (cytosol for Eukaryotes and cytoplasm for Prokaryotes).

Finally, the operation *Model >Export to SBML* allows exporting the internal model in the SBML format with MIRIAM annotations.

Validation

merlin has already been used to perform re-annotations and to reconstruct GSMMs for several organisms. For instance, merlin was used to perform the genome-scale metabolic re-annotations of *K. lactis* (8), *Ashbya gossypii* (39) and *Helicobacter pylori* (40), as well as to develop the GSMMs of *K. lactis* (41) and *H. pylori* (Resende, T., Correia, D.M.M. and Rocha, I. (2015) Reconstruction and validation of a genome-scale metabolic model for *Helicobacter pylori* 26695. *In preparation*).

The genome-wide functional re-annotation of the metabolic proteins encoded in the *K. lactis* genome led to the identification of 1759 genes with metabolic functions, including transporter proteins. The genes annotated with metabolic functions were exclusively enzymatic (1410 genes), transporter proteins encoding genes (301 genes) or had both metabolic activities (48 genes). *A. gossypii*'s annotation assigned metabolic functions to 847 genes, including 22 previously unreported enzymatic functions. This re-annotation allowed performing the comparison between *A. gossypii*'s metabolism and the ones of *S. cerevisiae* and *K. lactis*. Some enzymes were found exclusively in *A. gossypii* when compared to *K. lactis* (90) and *S. cerevisiae* (13). Also, 176 and 123 enzymatic functions were absent on *A. gossypii* comparatively to *K. lactis* and *S. cerevisiae*, respectively. On the other hand, the re-annotation of *H. pylori* led to the identification of 1212 genes encoding proteins. Over half of these genes (712) were identified as metabolic, including 191 new metabolic functions.

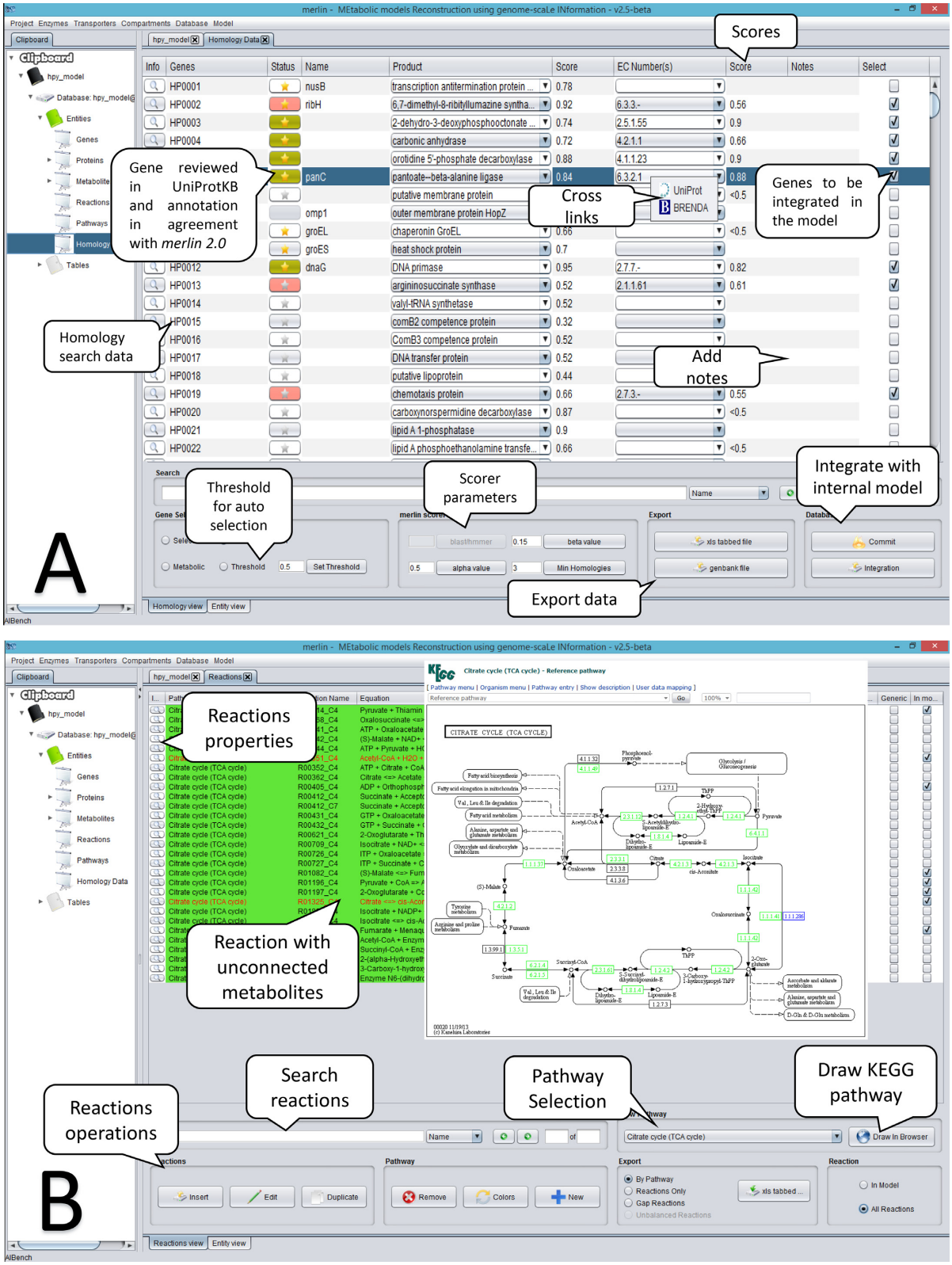

Figure 3. *merlin* annotation and modelling interfaces. (**A**) Homology data curation interface. (**B**) The reactions viewer is used for model curation. This panel allows adding, editing and removing reactions.

Table 2. Specific growth rate of the different models and its assessment to *in vivo* data

	In vivo	*i*TR383	*i*IT341	modelSEED435
μ	0.096	0.0919	0.161	0.017
$q_{glutamate}$	0.9348	0.9348	0 (produced)	0 (produced)

The upper limit of glutamate uptake was kept fixed in the experimental value for all simulations, the same being valid for the essential amino acids. All other medium components were left unbounded, to simulate the growth with glutamate as the carbon source.

All of the above annotations were used as a basis for the reconstruction of GSMMs of the respective organisms. The iOD907 *K. lactis* metabolic model (41) was fully developed using *merlin*. It has four compartments, 1867 reactions and 1476 metabolites. *In silico* growth in several carbon sources was tested and compared to experimental data, with a good agreement. Moreover, the model proved accurate when predicting biomass, oxygen and carbon dioxide yields and the effect of knockouts compared with *in vivo* phenotypes.

A new *H. pylori* metabolic model (Resende *et al.* 2015) was also reconstructed in *merlin*. This model comprises 3 compartments and 640 reactions. Since this organism already had a previously manually reconstructed model and a model developed in the model SEED framework, a comparison of the reaction sets of the models and of the predictive capacity of each model was performed and is presented below. For this, all modelSEED and *i*IT341 metabolite identifiers were converted into KEGG identifiers using an internally developed database.

As shown in Supplementary Figure S3.1 (supplemental file 3 of the supplementary data), only 103 reactions were exactly the same in all three models. However, if protons and compartments are ignored (since *i*TR383 has one more compartment) 120 extra reactions can be found on all three models. A more extensive comparison of the three models is available in supplemental file 3 of the Supplementary Data. This assessment shows that all three models are different, with no closer similarity among any of the pairs.

In the work of Correia (42), *H. pylori* was grown, in batch cultures, using glutamate as carbon source in a semi-defined medium. These experiments were performed after initial tests indicated that *H. pylori* uses amino acids as preferential carbon sources, even in the presence of other substrates such as glucose or organic acids. They also allowed to conclude that glutamine and glutamate are preferential among all amino acids, being *H. pylori* capable of growing using glutamate or glutamine as sole carbon sources. The full medium description is available in Supplementary Table S3.3 (supplemental file 3 of the supplementary data). The *in vivo* specific growth rate (μ) and the specific consumption rate (q_S) for glutamate were calculated according to Sauer *et al.* (43).

As shown in Table 2, the model developed in *merlin* (*i*TR383) complies better with the *in vivo* data. The *i*TR383 model shows the same specific growth rate verified *in vivo,* whilst being the only *in silico* model able to use glutamate as carbon source instead of producing it. In fact, when the conditions used *in vivo* are used to constrain the model, *i*IT341 does not use glutamate as carbon source, using instead L-alanine, an essential amino acid for that model. On its turn, the modelSEED uses fumarate as carbon source, which is also an essential metabolite for this model. Moreover, the *i*TR383 model performs quite well on the essentiality tests, as shown in Supplementary Table S3.4 (supplemental file 3 of the supplementary data).

The reconstruction of fairly accurate models for both eukaryotic and prokaryotic organisms, with clear improvements over existing reconstructions in the latter case, demonstrates that *merlin* provides a reliable framework for developing GSMMs.

CONCLUSIONS

Merlin is a user-friendly Java application that performs the reconstruction of genome-scale metabolic models for every organism that has its genome sequenced. It performs several steps of the reconstruction process, including the functional genomic annotations of the whole genome, using homology tools such as BLAST and HMMER. For every gene, homology information is retrieved and the results are automatically scored, allowing the user to change the automatic selection, and dynamically (re-)annotating the genome.

Moreover, *merlin* includes tools for the identification and annotation of genes encoding transport proteins, as well as the generation of transport reactions for such carriers. Also, tools for the compartmentalisation of the model that predict the localisation of the proteins encoded in the genome, and, thus, the localisation of the metabolites involved in the reactions induced by such proteins, were developed and integrated into *merlin*.

These operations, together with a unique tool for determining GPR associations allow performing the main tasks required to obtain reliable models.

Finally, *merlin* expedites the transition from genome-scale data to SBML metabolic models, also allowing the user to have a preliminary view of the biochemical network.

Therefore, a compartmentalised draft model, with GPRs, can be obtained in less than a week with *merlin* for eukaryotes with 4000–6000 genes, depending on the servers load and the quality of the internet connection, being the genome annotation usually the lengthier step. To overcome this potential bottleneck, currently merlin offers the option to annotate the genome using different databases (NCBI and UniProt) and different alignment algorithms (BLAST, HMMER). These different options are available so that users can select whichever database and algorithm provides better results and faster response times. Also, the queries to the remote web-servers have been parallelised, so that the data could be retrieved more rapidly. *merlin* also provides several tools for the curation of the genome annotation and the draft model, significantly aiding in the model validation and making possible to reach an accurate model.

merlin is freely available at www.merlin-sysbio.org, together with the complete source code.

AVAILABILITY

merlin is an open-source application, currently available for Linux and Windows. It is distributed under the GNU General Public License at the website http://www.merlin-sysbio.org.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

FUNDING

PhD grant to O. D. [SFRH /BD/47307/2008]; ERDF—European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness); National Funds through the FCT within the projects FCOMP-01–0124-FEDER-009707 (HeliSysBio—molecular Systems Biology in *Helicobacter pylori*) and FCOMP-01–0124-FEDER-015079 (ToMEGIM—Computational Tools for Metabolic Engineering using Genome-scale Integrated Models).

FCT Strategic Project PEst-OE/EQB/LA0023/2013 and the Projects 'BioInd—Biotechnology and Bioengineering for improved Industrial and Agro-Food processes', REF. NORTE-07–0124-FEDER-000028 and 'PEM—Metabolic Engineering Platform', project number 23060, both co-funded by the Programa Operacional Regional do Norte (ON.2—O Novo Norte), QREN, FEDER. Funding for open access charge: Centre of Biological Engineering.

Conflict of interest statement. None declared.

REFERENCES

1. Rocha,I., Förster,J. and Nielsen,J. (2008) Design and application of genome-scale reconstructed metabolic models. *Methods Mol. Biol.*, **416**, 409–431.
2. Feist,A.M., Herrgård,M.J., Thiele,I., Reed,J.L. and Palsson,B.Ø. (2009) Reconstruction of biochemical networks in microorganisms. *Nat. Rev. Microbiol.*, **7**, 129–143.
3. Thiele,I. and Palsson,B.Ø. (2010) A protocol for generating a high-quality genome-scale metabolic reconstruction. *Nat. Protoc.*, **5**, 93–121.
4. Francke,C., Siezen,R.J. and Teusink,B. (2005) Reconstructing the metabolic network of a bacterium from its genome. *Trends Microbiol.*, **13**, 550–558.
5. Dias,O. and Rocha,I. (2015) Systems Biology in Fungi. In: Paterson,R (ed). *Molecular Biology of Food and Water Borne Mycotoxigenic and Mycotic Fungi*. CRC Press, Boca Raton, 69–92.
6. Barrett,A.J., Canter,C.R., Liebecq,C., Moss,G.P., Saenger,W., Sharon,N., Tipton,K.F., Vnetianer,P. and Vliegenthart,V.F.G. (1992) In: Webb,EC (ed). *Enzyme Nomenclature NC-ICBMB*. Academic Press, San Diego.
7. Saier,M.H., Tran,C.V. and Barabote,R.D. (2006) TCDB: the Transporter Classification Database for membrane transport protein analyses and information. *Nucleic Acids Res.*, **34**, D181–D186.
8. Dias,O., Gombert,A.K., Ferreira,E.C. and Rocha,I. (2012) Genome-wide metabolic (re-) annotation of *Kluyveromyces lactis*. *BMC Genomics*, **13**, 517.
9. Gundogdu,O., Bentley,S.D., Holden,M.T., Parkhill,J., Dorrell,N. and Wren,B.W. (2007) Re-annotation and re-analysis of the *Campylobacter jejuni* NCTC11168 genome sequence. *BMC Genomics*, **8**, 162.
10. Camus,J.-C., Pryor,M.J., Médigue,C. and Cole,S.T. (2002) Re-annotation of the genome sequence of *Mycobacterium tuberculosis* H37Rv. *Microbiology*, **148**, 2967–2973.
11. Scheer,M., Grote,A., Chang,A., Schomburg,I., Munaretto,C., Rother,M., Söhngen,C., Stelzer,M., Thiele,J. and Schomburg,D. (2011) BRENDA, the enzyme information system in 2011. *Nucleic Acids Res.*, **39**, D670–D676.
12. Lang,M., Stelzer,M. and Schomburg,D. (2011) BKM-react, an integrated biochemical reaction database. *BMC Biochem.*, **12**, 42.
13. Caspi,R., Altman,T., Dreher,K., Fulcher,C.A., Subhraveti,P., Keseler,I.M., Kothari,A., Krummenacker,M., Latendresse,M., Mueller,L.A. *et al.* (2012) The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. *Nucleic Acids Res.*, **40**, D742–D753.
14. Edwards,J.S. and Palsson,B.O. (1999) Systems properties of the Haemophilus influenzae Rd metabolic genotype. *J. Biol. Chem.*, **274**, 17410–17416.
15. Boele,J., Olivier,B.G. and Teusink,B. (2012) FAME, the Flux Analysis and Modeling Environment. *BMC Syst. Biol.*, **6**, 8.
16. Pabinger,S., Rader,R., Agren,R., Nielsen,J. and Trajanoski,Z. (2011) MEMOSys: Bioinformatics platform for genome-scale metabolic models. *BMC Syst. Biol.*, **5**, 20.
17. Feng,X., Xu,Y., Chen,Y. and Tang,Y.J. (2012) MicrobesFlux: a web platform for drafting metabolic models from the KEGG database. *BMC Syst. Biol.*, **6**, 94.
18. Karp,P.D., Paley,S. and Romero,P. (2002) The Pathway Tools software. *Bioinformatics*, **18** (Suppl. 1), S225–S232.
19. Pitkänen,E., Jouhten,P., Hou,J., Syed,M.F., Blomberg,P., Kludas,J., Oja,M., Holm,L., Penttilä,M., Rousu,J. *et al.* (2014) Comparative genome-scale reconstruction of gapless metabolic networks for present and ancestral species. *PLoS Comput. Biol.*, **10**, e1003465.
20. Agren,R., Liu,L., Shoaie,S., Vongsangnak,W., Nookaew,I. and Nielsen,J. (2013) The RAVEN toolbox and its use for generating a genome-scale metabolic model for Penicillium chrysogenum. *PLoS Comput. Biol.*, **9**, e1002980.
21. DeJongh,M., Formsma,K., Boillot,P., Gould,J., Rycenga,M. and Best,A. (2007) Toward the automated generation of genome-scale metabolic networks in the SEED. *BMC Bioinformatics*, **8**, 139.
22. Hucka,M., Finney,A., Sauro,H.M., Bolouri,H., Doyle,J.C., Kitano,H., Arkin,A.P., Bornstein,B.J., Bray,D., Cornish-Bowden,A. *et al.* (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, **19**, 524–531.
23. Ogata,H., Goto,S., Sato,K., Fujibuchi,W., Bono,H. and Kanehisa,M. (1999) KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.*, **27**, 29–34.
24. Glez-Peña,D., Reboiro-Jato,M., Maia,P., Rocha,M., Díaz,F. and Fdez-Riverola,F. (2010) AIBench: a rapid application development framework for translational research in biomedicine. *Comput. Methods Programs Biomed.*, **98**, 191–203.
25. Holland,R.C.G., Down,T.A., Pocock,M., Prlić,A., Huen,D., James,K., Foisy,S., Dräger,A., Yates,A., Heuer,M. *et al.* (2008) BioJava: an open-source framework for bioinformatics. *Bioinformatics*, **24**, 2096–2097.
26. Patient,S., Wieser,D., Kleen,M., Kretschmann,E., Martin,M. and Apweiler,R. (2008) UniProtJAPI: a remote API for accessing UniProt data. *Bioinformatics*, **24**, 1321–1322.
27. Dräger,A., Rodriguez,N., Dumousseau,M., Dörr,A., Wrzodek,C., Le Novère,N., Zell,A. and Hucka,M. (2011) JSBML: a flexible Java library for working with SBML. *Bioinformatics*, **27**, 2167–2168.
28. Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.
29. Eddy,S.R. (1998) Profile hidden Markov models. *Bioinformatics*, **14**, 755–763.
30. Saier,M.H. (2000) A functional-phylogenetic classification system for transmembrane solute transporters. *Microbiol. Mol. Biol. Rev.*, **64**, 354–411.
31. Krogh,A., Larsson,B., von Heijne,G. and Sonnhammer,E.L. (2001) Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. *J. Mol. Biol.*, **305**, 567–580.
32. Smith,T.F. and Waterman,M.S. (1981) Identification of common molecular subsequences. *J. Mol. Biol.*, **147**, 195–197.

33. Krause,F., Uhlendorf,J., Lubitz,T., Schulz,M., Klipp,E. and Liebermeister,W. (2010) Annotation and merging of SBML models with semanticSBML. *Bioinformatics*, **26**, 421–422.
34. Horton,P., Park,K.-J., Obayashi,T., Fujita,N., Harada,H., Adams-Collier,C.J. and Nakai,K. (2007) WoLF PSORT: protein localization predictor. *Nucleic Acids Res.*, **35**, W585–W587.
35. Yu,N.Y., Wagner,J.R., Laird,M.R., Melli,G., Rey,S., Lo,R., Dao,P., Sahinalp,S.C., Ester,M., Foster,L.J. *et al.* (2010) PSORTb 3.0: improved protein subcellular localization prediction with refined localization subcategories and predictive capabilities for all prokaryotes. *Bioinformatics*, **26**, 1608–1615.
36. Tanabe,M. and Kanehisa,M. (2012) Using the KEGG database resource. *Curr. Protoc. Bioinformatics*, Chapter 1, Unit 1.2.
37. Le Novère,N., Finney,A., Hucka,M., Bhalla,U.S., Campagne,F., Collado-Vides,J., Crampin,E.J., Halstead,M., Klipp,E., Mendes,P. *et al.* (2005) Minimum information requested in the annotation of biochemical models (MIRIAM). *Nat. Biotechnol.*, **23**, 1509–1515.
38. Rocha,I., Maia,P., Evangelista,P., Vilaça,P., Soares,S., Pinto,J.P., Nielsen,J., Patil,K.R., Ferreira,E.C. and Rocha,M. (2010) OptFlux: an open-source software platform for *in silico* metabolic engineering. *BMC Syst. Biol.*, **4**, 45.
39. Gomes,D., Aguiar,T.Q., Dias,O., Ferreira,E.C., Domingues,L. and Rocha,I. (2014) Genome-wide metabolic re-annotation of *Ashbya gossypii*: new insights into its metabolism through a comparative analysis with *Saccharomyces cerevisiae* and *Kluyveromyces lactis*. *BMC Genomics*, **15**, 810.
40. Resende,T., Correia,D.M., Rocha,M. and Rocha,I. (2013) Re-annotation of the genome sequence of *Helicobacter pylori* 26695. *J. Integr. Bioinform.*, **10**, 233.
41. Dias,O., Pereira,R., Gombert,A.K., Ferreira,E.C. and Rocha,I. (2014) iOD907, the first genome-scale metabolic model for the milk yeast *Kluyveromyces lactis*. *Biotechnol. J.*, **9**, 776–790.
42. Correia,D.M.M. (2014) Systems analysis of metabolism in *Helicobacter pylori*. PhD thesis University of Minho.
43. Sauer,U., Lasko,D.R., Fiaux,J., Hochuli,M., Glaser,R., Szyperski,T., Wuthrich,K. and Bailey,J.E. (1999) Metabolic flux ratio analysis of genetic and environmental modulations of *Escherichia coli* central carbon metabolism. *J. Bacteriol.*, **181**, 6679–6688.