RBA pipeline

S. Fischer - Biosys - MAIAGE

April 11, 2017

Contents

1 Description of workflow

Figure 1 shows the global workflow of the pipeline. It contains four parts:

1. **PreRba**: Parsing of biological data into RBA compatible XML files. Parts of the process are semi-automated, meaning the user is needed to help solve ambiguous annotations.
2. **RbaXml**: Parsing/modifying XML files.
3. **RbaMatrices**: XML files are transformed into matrices.
4. **RbaSolver**: matrices are used to compute the optimal resource allocation.

As a first step, this workflow should run for any prokaryote.

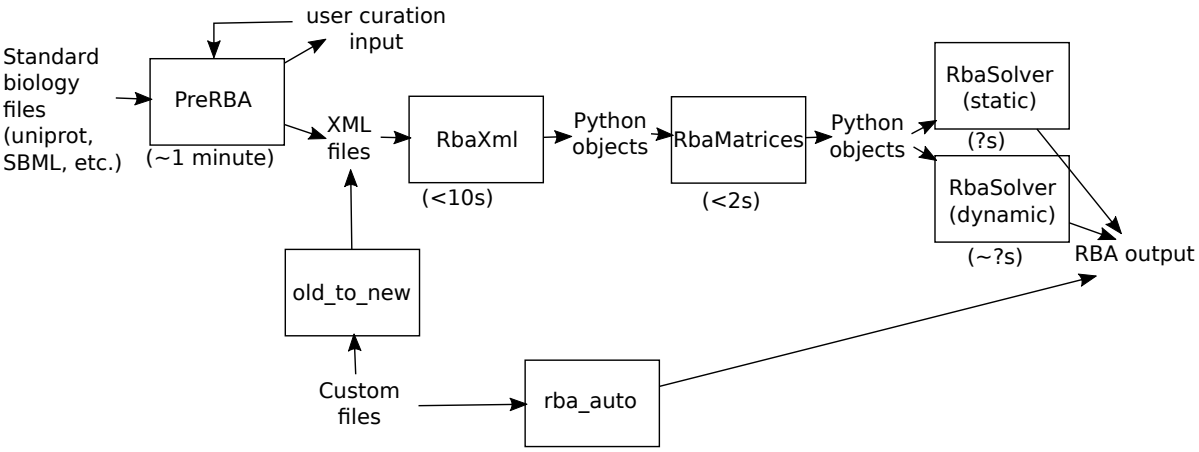

Figure 1: Workflow of pipeline. The lower part shows how compatibility between the workflow and the former version of RBA were maintained (used for consistency checks).

1.1 Philosophy of the pipeline

Everyting should work from first run on We would like the pipeline to run completely on the first run. This means that a user that inputs an approximately standard SBML file should generate a full RBA model and get first growth rate results without having to do anything. Because annotations are often ambiguous, a lot of default parameters will be used for the first run, but the user will be able to parameterize their model progressively.

Helping user to setup main parameters through csv files Conversion of biological data is very heavy and often ambiguous. When an annotation is ambiguous, the user will be asked for help through csv files. Everything that is input by the user is stored so the user does not have to provide the same information twice.

Helping user to setup fine parameters through xml files After the main parameters have been set, the user will have ready-to-use RBA input files. We cannot handle every single parameter through csv files. Because RBA input files are written in xml, an advanced user will still be able to control more subtle parameters in a standardized (but programmatical) way.

1.2 Typical expected usage

1. The user provides SBML and runs the pipeline. They are happy because everything runs and he gets some output value. They are unhappy because this output value is unrealistic.

2. The user spends time going through the helper files and understands why they are here. They provide all the information needed to create a system fully adapted to their organism. They run the whole pipeline to see how growth rate has evolved and generate new and more consistent XML input files.

3. They spend time fine-tuning processes and enzyme catalytic efficiencies by modifying the XML files, finally reaching a biological sound model.

2 preRBA: converting biological data

In order to get the workflow working, the user has to provide an SBML file describing the metabolism of their organism. A uniprot file is also needed, but it can be retrieved automatically. Several helper files will be generated after the first run of the pipeline. The user needs to adapt these files in order to have a biologically relevant model.

This section lists the requirements for the SBML file and the formats used by the helper files.

2.1 SBML: extraction of metabolism and enzyme information

2.1.1 Requirements

1. All cytosolic metabolites should end with the suffix `_c`.
2. Every reaction should contain information about associated enzyme composition. There are two accepted formats:
   - Using the `<fbc:geneProductAssociation>` tag of the Flux Balance Constraints package for sbml.
   - Using the `<notes>` tag containing a GENE_ASSOCIATION field. Gene names should be separated using white spaces `' '` or underscores `'_'`. Association of genes should be described by the keywords `or` and `and`. Parentheses may or may not be used.

2.1.2 Warnings

- The biomass reaction and reactions used to assemble non-metabolites (*e.g.* proteins, rnas, etc.) are not used in the RBA model. The solver will usually assign them zero fluxes. They may safely be removed from the system.
- If a gene listed in the gene association cannot be retrieved in uniprot, it will be replaced by an average protein.
- If a gene association is left empty, the pipeline will assume the reaction is spontaneous.

2.2 Uniprot: extraction of protein information

2.2.1 Requirements

A uniprot file is needed to cross-reference proteins with SBML data. The user needs to provide the Uniprot id of its organism, so a uniprot file can automatically be retrieved. Alternatively, the user can provide a Uniprot file matching following requirements:

- Required fields are: Entry, Gene names, Protein names, Sequence, Cofactor, Subcellular location [CC], Subunit stucture [CC].

2.3 Helper Files

Helper files are tab-separated files generated to handle ambiguous data and parameters for the original input files.

location.tsv We need to match uniprot locations with SBML compartments. Uniprot location can be retrieved automatically. A helper file is generated where the user has to indicate the SBML compartment ids corresponding to the different uniprot locations.

2.4 Automated parsing rules

2.4.1 Enzymatic activity

Default enzyme activity For all enzymes, a constant catalytic activity is applied. In this context, constant means that the catalytic activity does not depend on growth rate.

Transporter detection An enzyme is considered to be a transporter if the following rules apply:

- One of products has the same prefix as an external metabolite (*e.g.* `M_nad_e` has the same prefix as `M_nad_p`).
- One of the reactants is in the cytosol.

The catalytic activity of a transporter is modified in the following way.

- The main catalytic activity is given by the default catalytic activity applied to all enzymes (see above).
- The main catalytic activity is multiplied by a substrate-dependent term ranging from 0 to 1. More precisely, import activity is given by a Michaelis-Mentent function that depends on the concentration of the *external* counterpart of the imported product (*e.g.* if the imported product is `M_nad_p`, the import activity depends on `M_nad_e`).

Note that the latter choice may lead to non-intuitive behaviours. For example, take a bacterium that is able to transform trehalose into glucose in the *periplasm*. Suppose the external concentration of trehalose is set to 0 but the external concentration of glucose is nonzero. The solver might decide to import the (non-existing) trehalose into the periplasm (because import into the periplasm does *not* depend on medium concentrations), transform it into glucose that will automatically be assumed to be at medium concentration and then imported into the cytoplasm.

2.4.2 Enzyme composition and location

Enzyme composition is computed from protein information retrieved in uniprot.

Cofactor stoichiometry From the uniprot `Cofactor` field, we use the following rules to parse protein cofactor information:

- If field is empty, we assume there is no cofactor.
- If there is exactly one occurrence of the keyword `Binds`, we assume stoichiometry is the number that follows `Binds`.
- If there is no stoichiometry information using keyword `Binds`, we assume stoichiometry is 1.
- If we find exactly one name and its associated CHEBI identifier, and stoichiometry could be determined as described above, annotation is considered to be not ambiguous.
- In any other case, annotation is considered ambiguous and written to helper file for user review. Still we use following heuristics:
  - If there were several names and associated CHEBI identifier and stoichiometry could be determined, we assume that only the first cofactor listed is relevant. We give it full stoichiometry and 0 stoichiometry to all other cofactors.

Subunit structure From the uniprot `Subunit stucture [CC]` field, we use the following rules to parse the stoichiometry of proteins within their enzymatic complex:

- If field is empty, we assume there is one subunit in the complex.
- If field contains exactly one occurence of the form "*prefix*mer", we look at the prefix. If prefix is mono or heterodi, we assume stoichiomery is one. If prefix is homodi, homotri, homotera, homopenta, homohexa, hepta, homoocta, homodeca, homododeca, we assume the number of subunits corresponds to the prefix.
- In any other case, field is considered ambiguous and written to helper file for user review.

Location From the uniprot `Subcellular location [CC]` field, we use the following rules to parse location information.

- If field is non-empty, location is usually non ambiguous.
- If field is empty, protein is assumed to be in the cytosol.