Grohar: automated visualisation of genome-scale metabolic models and their pathways

User manual

Miha Moškon, Nikolaj Zimic and Miha Mraz

Faculty of Computer and Information Science

University of Ljubljana

Slovenia

University *of Ljubljana*
Faculty *of Computer and Information Science*
Computational Biology Group

Contents

1. What is Grohar?

Grohar is a computational tool for the visualisation of metabolic networks and their analysis using *matplotlib*, *networkx* and *cobra* packages. Grohar was written in python together with *pyqt* GUI toolkit to provide a user friendly experience.

2. Installation

Grohar is available from the bitbucket repository https://bitbucket.org/mmoskon/grohar. Before you get started you will have to install python 3.5+ with some additional packages. We recommend the installation of Anaconda (https://docs.continuum.io/anaconda/install/) to minimize the number of additional packages that you will need to install. If you will go with Anaconda, you will need to install only the following python packages:

- *cobra 0.5.4 (pip install cobra==0.5.4)*,
- *python-libsbml*,
- *bioservices*,
- *pydotplus,*
- *PyQt5*,
- *cobrababel,*
- *python-Levenshtein.*

You can install them using *pip*.

You also need to install *graphviz*, which can be downloaded from http://www.graphviz.org/.

If you will be using the default Anaconda IDE, i.e. Spyder, you need to configure IPython in Spyder to open the Matplotlib graphs in a separate window. In Spyder go to *Tools* → *Preferences* → *IPython console* → *Graphics*. Here you will find a setting entitled *Graphics Backend*, which should be set to *Automatic* (its default value is *Inline*).

If you want to run advanced functions, such as MOMA, you should also install a solver, which can handle quadratic programming problems, such as *gurobi*. The easiest way to install *gurobi* in Anaconda environment is to type the following commands in the command line:

```
conda config --add channels http://conda.anaconda.org/gurobi
conda install gurobi
```

You should also install the gurobi license for the solvers to work. The license and instructions are available at http://www.gurobi.com/academia/for-universities.

Now you only need to find the main file of the project (*top.py*) and run it by pressing the F5 key.

Grohar was written in python together with *pyqt* GUI toolkit to provide a user friendly experience. Graphical user interface is located within the file *top.ui* and can be modified easily with Qt Designer (included within the package *pyqt-distutils*).

3. Loading the model

Grohar is able to load Matlab (*.mat) or SBML models (*.xml) in COBRA dialect (Schellenberger, et al., 2011). Three models among which you can choose are already available together with the tool, namely three context specific models of iCHO (Selvarasu, et al., 2012). These models can be selected from the *Preloaded models* combo box and loaded with the *Load Model* button (see Figure *1*).

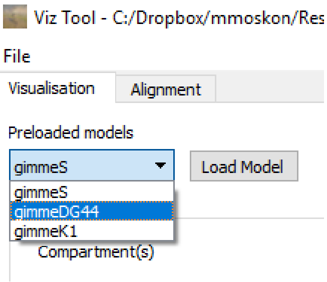
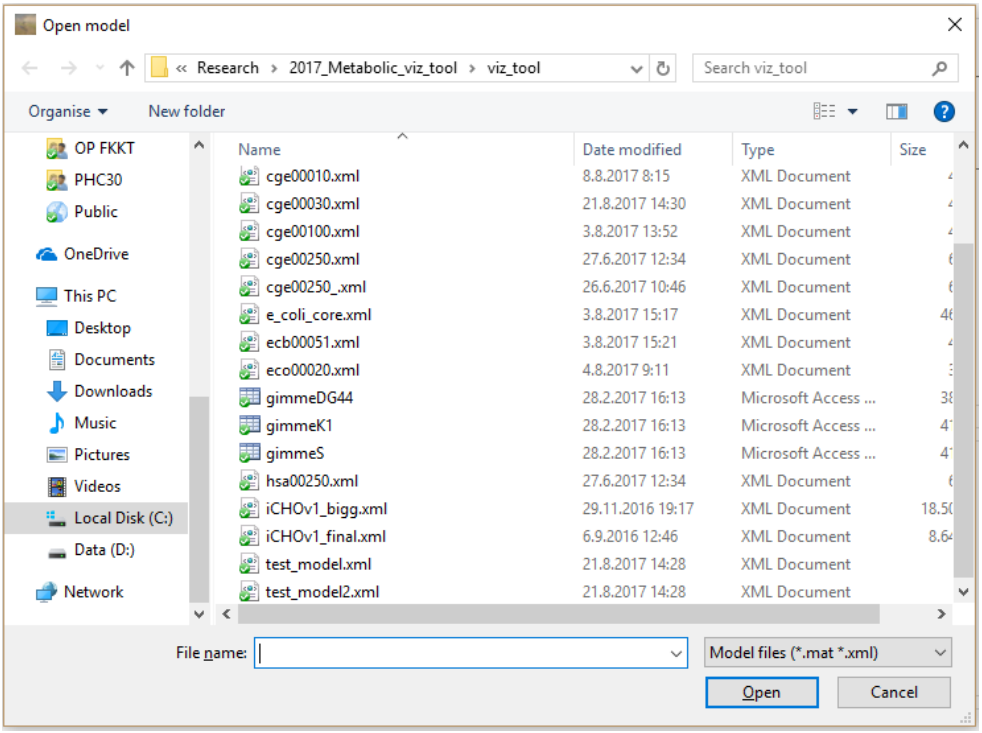


Figure 1: You can select among three preloaded models.

You can select an arbitrary model from your computer by clicking the menu entry *File* → *Open*. Open file dialog appears, which allows you to select the model in **.mat* or **.xml* format (see Figure *2*).



Name	Date modified	Type	Size
cge00010.xml	8.8.2017 8:15	XML Document	4
cge00030.xml	21.8.2017 14:30	XML Document	4
cge00100.xml	3.8.2017 13:52	XML Document	4
cge00250.xml	27.6.2017 12:34	XML Document	6
cge00250_.xml	26.6.2017 10:46	XML Document	6
e_coli_core.xml	3.8.2017 15:17	XML Document	46
ecb00051.xml	3.8.2017 15:21	XML Document	4
eco00020.xml	4.8.2017 9:11	XML Document	3
gimmeDG44	28.2.2017 16:13	Microsoft Access ...	38
gimmeK1	28.2.2017 16:13	Microsoft Access ...	41
gimmeS	28.2.2017 16:13	Microsoft Access ...	41
hsa00250.xml	27.6.2017 12:34	XML Document	6
iCHOv1_bigg.xml	29.11.2016 19:17	XML Document	18.50
iCHOv1_final.xml	6.9.2016 12:46	XML Document	8.64
test_model.xml	21.8.2017 14:28	XML Document	
test_model2.xml	21.8.2017 14:28	XML Document	

File name:

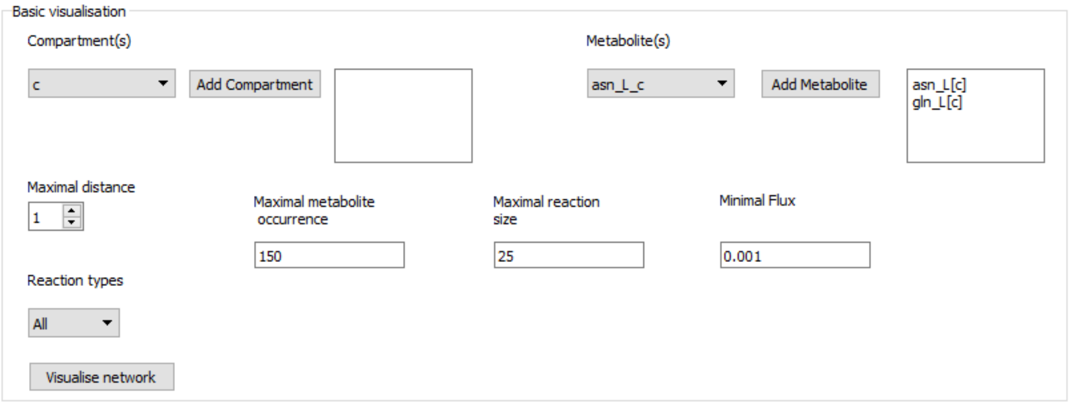
Model files (*.mat *.xml)

Figure 2: You can select an arbitrary SBML or Matlab model in COBRA dialect.

4. Basic visualisation

Setting the parameters

Basic visualisation allows you to visualise the neighbourhood of specific metabolite(s). The controls that are dedicated to this visualisation are shown in Figure *3*.



Compartment(s): c

Metabolite(s): asn_L_c

asn_L[c]
gln_L[c]

Maximal distance: 1

Maximal metabolite occurrence: 150

Maximal reaction size: 25

Minimal Flux: 0.001

Reaction types: All

Figure 3: The group box containing the controls for performing the basic visualisation.

These controls are enabled only after you have successfully loaded the model. Their meaning is as follows:

- *Compartment(s)*: you can specify the model compartment(s) of interest. Options are:
  - If no compartments are specified, the visualisation will ignore the compartmentation of metabolites and reactions.
  - If one compartment is specified, only the metabolites and reactions belonging to this compartment will be visualised.
  - If more compartments are specified, only the transport reactions between these compartments and their corresponding metabolites will be visualised.
- *Metabolite(s)*: you have to specify at least one metabolite of interest. Grohar will visualise the metabolic (sub)network surrounding, i.e. producing or consuming, the given metabolite(s).
- *Maximal distance*: you can define the size of the neighbourhood around the metabolite(s) of interest with this parameter. If *Maximal distance* is set to 1 (default), only adjacent metabolites (metabolites that share the same reactions with the selected metabolite(s)) and adjacent reactions (reactions in which at least one of the selected metabolites is either product or reactant) will be visualised.
- *Reaction types*: you can select among four different options:
  - *All*: reactions producing or consuming at least one of the selected metabolites or its neighbours (if maximal distance is larger than 1) will be visualised.
  - *Producing*: reactions producing at least one of the selected metabolites or its neighbours through other producing reactions (if maximal distance is larger than 1) will be visualised.

    - *Consuming*: reactions consuming at least one of the selected metabolites or its neighbours through other consuming reactions (if maximal distance is larger than 1) will be visualised.
    - *Both*: reaction that are producing and at the same time consuming at least one of the selected metabolites or its neighbours through other reactions with the same property (if maximal distance is larger than 1) will be visualised.
- *Maximal metabolite occurrence*: if set, the metabolites that are present in more than the given number of reactions will be omitted from the visualisation. If not set, set to 0 or a negative value, the parameter will be ignored.
- *Maximal reaction size*: if set, the reactions that have more than the given number of metabolites will be omitted from the visualisation. If not set, set to 0 or a negative value, the parameter will be ignored.
- *Minimal flux*: if set, the reactions with absolute metabolic flux lower than the given value will be omitted from the visualisation. If not set, set to 0 or a negative value, the parameter will be ignored.

After the parameters have been set (the only mandatory parameter is the *Metabolite(s)* parameter), you can visualise the sub(network) with the click on the *Visualise network* button.

Example

We will try to perform a simple visualisation on one of the preloaded models. We will select the *gimmeDG44* model from the *Preloaded* models combo box and click the *Load model* button. The fields in the *Basic visualisation* group box should now be enabled. Now, we will try to visualise the sub(network) of the model surrounding the metabolite with the COBRA ID *arg_L[c]* (*L-argininium(1+)*). First, we will delete the default entries within the *Metabolite*(s) text edit box. We can replace these entries by typing in *arg_L[c]* or by selecting *arg_L[c]* from the *Metabolite(s)* combo box and by clicking the *Add Metabolite* button (see Figure *4*).

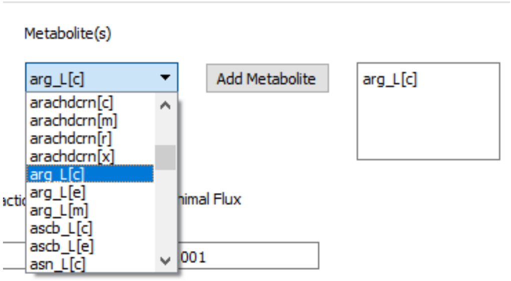


Figure 4: Selecting the *arg_L[c]* entry from the *Metabolite(s)* combo box.

When we click the *Visualise network* button new window with the visualisation of the metabolic sub(network) appears (see Figure *5*). Here, the network is visualised as a bipartite directed graph. Smaller nodes correspond to the metabolites and larger nodes to the metabolic reactions. Metabolite(s) of interest (in our case *arg_L[c]*) are coloured purple. Reactions producing the metabolites(s) of interest are coloured green and reactions consuming metabolites(s) of interest are coloured red. Each metabolite and reaction (each node) has a label with the identifier of the entity that it represents. The labels corresponding to the metabolic reactions also include flux values through the reactions. The directionalities of the edges correspond to the directionalities of the observed metabolic reactions. When you hover the mouse cursor over the node its full name is displayed. The legend (Figure *6*) can be displayed by pressing the L key on the keyboard.

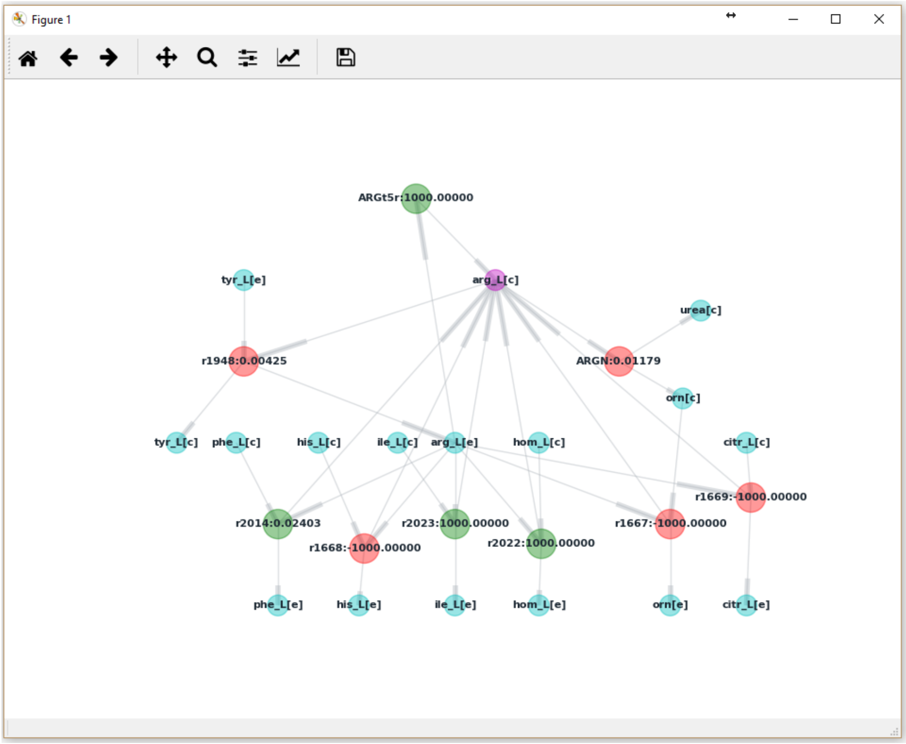
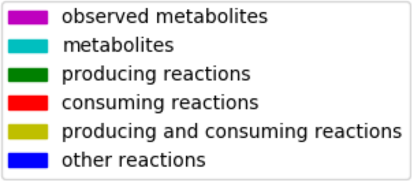


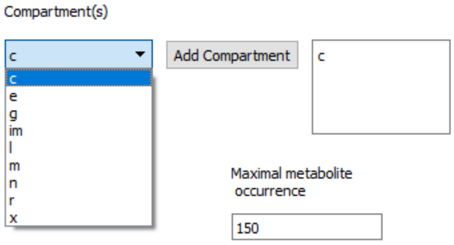
Figure 5: The visualisation of the metabolic sub(network) surrounding the metabolite *arg_L[c]*.



🟪 observed metabolites
🟦 metabolites
🟩 producing reactions
🟥 consuming reactions
🟨 producing and consuming reactions
🟦 other reactions

Figure 6: The legend describing the colours of the nodes. The legend can be displayed by pressing the L key on the keyboard.

We can further reduce the visualised sub(network) by specifying the compartment of interest. We can simply add an entry to the *Compartment(s)* text edit box or select an entry from the *Compartment(s)* combo box and click the *Add Compartment* button (see Figure *7*).



Compartment(s): c

c

Maximal metabolite occurrence: 150

Figure 7: Selecting *c* as *cytosol* from the *Compartment(s)* combo box.

We will select the compartment *c* (as *cytosol*) and click the *Visualise network* button. The visualised sub(network) now includes only the reactions that have all their reactants and products in the cytosol (see Figure *8*).

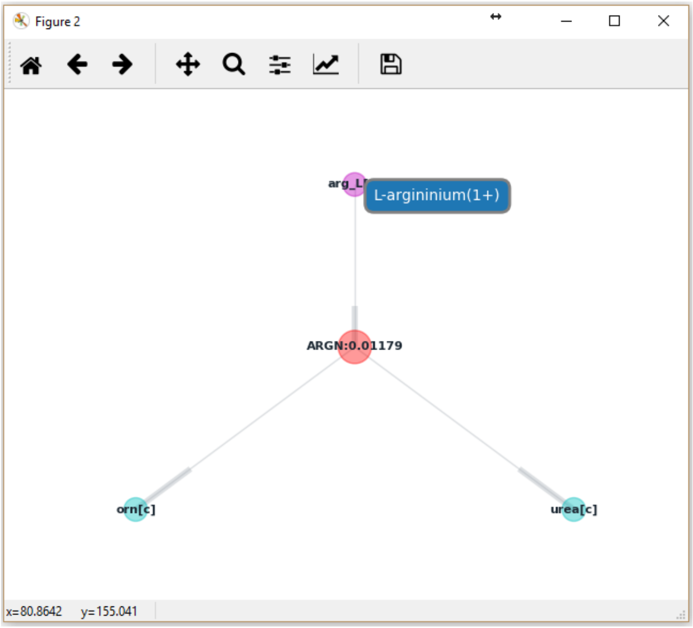
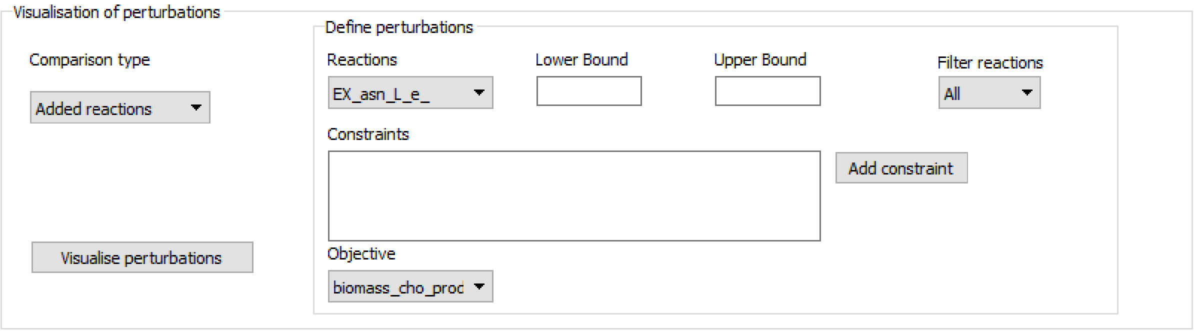


Figure 8: The visualisation of the metabolic sub(network) surrounding the metabolite *arg_L[c]*. The visualisation was limited to include only the cytosolic reactions. Mouse cursor was hovered over the node *arg_L[c]* to display its full name.

The positions of the visualized nodes can be moved around manually by simply drag-and-droping the corresponding nodes with your mouse.

5. Visualisation of perturbations

You can impose additional constraints on the model and observe their consequences on the selected metabolic (sub)network. The controls that are dedicated to the visualisation of perturbations are shown in Figure *9*.



Comparison type: Added reactions

Reactions: EX_asn_L_e_

Filter reactions: All

Objective: biomass_cho_proc

Figure 9: The group box containing the controls for performing the visualisation of perturbations.

Defining additional constraints

You can define additional constraints within the *Define perturbations* group box of the main window. The following controls are dedicated to this purpose:

- *Reactions*: you can select among all the reactions within the model. When a reaction is selected *Lower Bound* and *Upper Bound* fields will be populated with the values obtained from the model.
- *Lower Bound*: field defines the lower bound of the flux through the selected reaction. You may change this value to define a new lower bound for the reaction.
- *Upper Bound*: field defines the upper bound of the flux through the selected reaction. You may change this value to define a new upper bound for the reaction.
- *Add constraint*: when you are finished with the modification of the selected reaction boundaries, the constraints have to be added to the *Constraints* text box with the click on the *Add constraint* button (note: you may also edit the *Constraint* text box manually).
- *Filter reactions*: you can filter the metabolic reaction from the model, so the *Reactions* combo box will include only selected set of the reactions. You can choose among *All*, *Exchange* and *Uptake* reactions.
- *Objective*: you may select a different objective function and observe the modification of reaction fluxes if objective function is altered.

Visualising perturbations

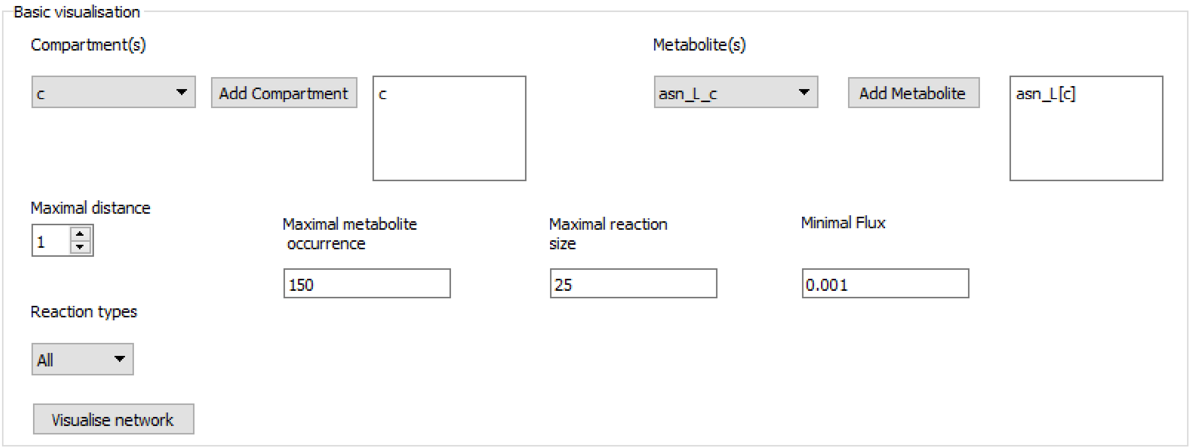
After the constraints have been set you can visualise the differences that occur within the observed sub(network) (defined in the *Basic visualisation* part of the window) after the perturbations are induced on the model. You can choose among three different types of comparison between unperturbed and perturbed model. Type of comparison can be selected with the *Comparison type* combo box, namely:

- *Added reactions*: visualisation will include the reactions that became active after the perturbation.
- *Removed reactions*: visualisation will include the reactions that became inactive after the perturbation.
- *Remained reactions*: visualisation will include the reactions that were active before and after the perturbation, but with different reaction fluxes.

The visualisation of perturbations will be performed in a similar manner as the basic visualisation after the *Visualise perturbations* button is clicked.

Example

We will try to visualise the effects of the removal of *asn_L* (*asparagine*) from the medium. First we need to set the *Basic visualisation* parameters to analyse *asn_L* within the cytosol. We will set the *Compartment(s)* text box to *c* and *Metabolite(s)* text box to *asn_L[c]*.



Basic visualisation

Compartment(s): c ▾ | Add Compartment | c

Metabolite(s): asn_L_c ▾ | Add Metabolite | asn_L[c]

Maximal distance: 1

Maximal metabolite occurrence: 150

Maximal reaction size: 25

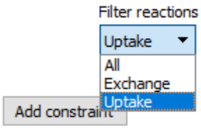
Minimal Flux: 0.001

Reaction types: All ▾

Visualise network

Figure 10: We will analyse the metabolic reactions around the *cytosolic asparagine* (*asn_L[c]*).

Visualisation of this sub(network) yields empty results since all the cytosolic reactions around *asn_L[c]* are inactive. We are interested in what happens if we remove the *asn_L* from the medium. We will find an uptake reaction that transports the metabolite from the medium into the cell. First, we will filter the reactions, so only uptake reactions will be displayed within the *Reactions* combo box. We will set the *Filter reactions* combo box to *Uptake (see* Figure *11).*



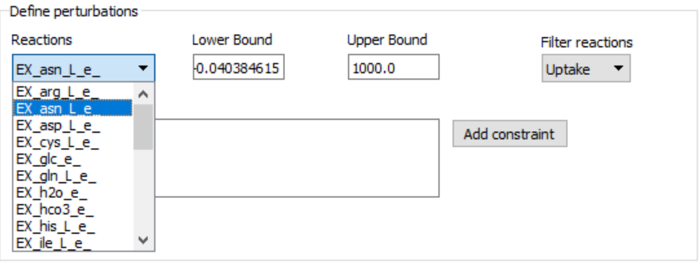
Filter reactions: Uptake ▾

All
Exchange
Uptake

Add constraint

Figure 11: We are only interested in the uptake reactions.

Now, we will select the *asparagine* uptake reaction, i.e. *EX_asn_L_e_*, from the *Reactions* combo box (see Figure *12).*



Define perturbations

Reactions: EX_asn_L_e_

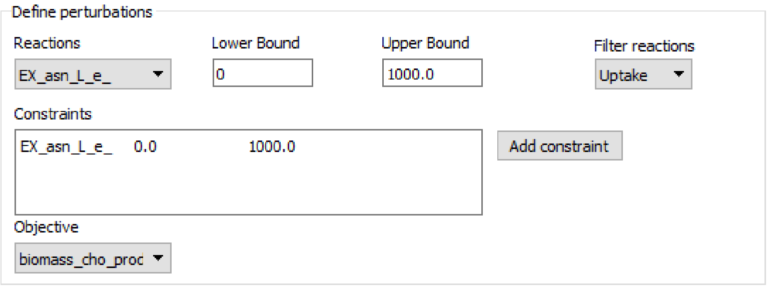
Lower Bound: -0.040384615

Upper Bound: 1000.0

Filter reactions: Uptake

Figure 12: We would like to change the constraints of the *asparagine* uptake reaction.

Values of the selected reaction boundaries defined in the model now populate *Lower Bound* and *Upper Bound* fields. We will set the lower bound to zero, which will simulate the absence of *asparagine* in the medium. Now we just need to click the button *Add constraint* so the constraint is added (see Figure *13).*



Define perturbations

Reactions: EX_asn_L_e_

Lower Bound: 0

Upper Bound: 1000.0

Filter reactions: Uptake

Constraints

```
EX_asn_L_e_    0.0            1000.0
```

Objective: biomass_cho_proc

Figure 13: The constraint was added to the *Constraints* text box.

We can now visualise the reactions that become active after the perturbation by clicking the *Visualise perturbations* button. We can see that *asparagine synthase* reaction was activated, after the *asparagine* was removed from the medium (see Figure *14*).

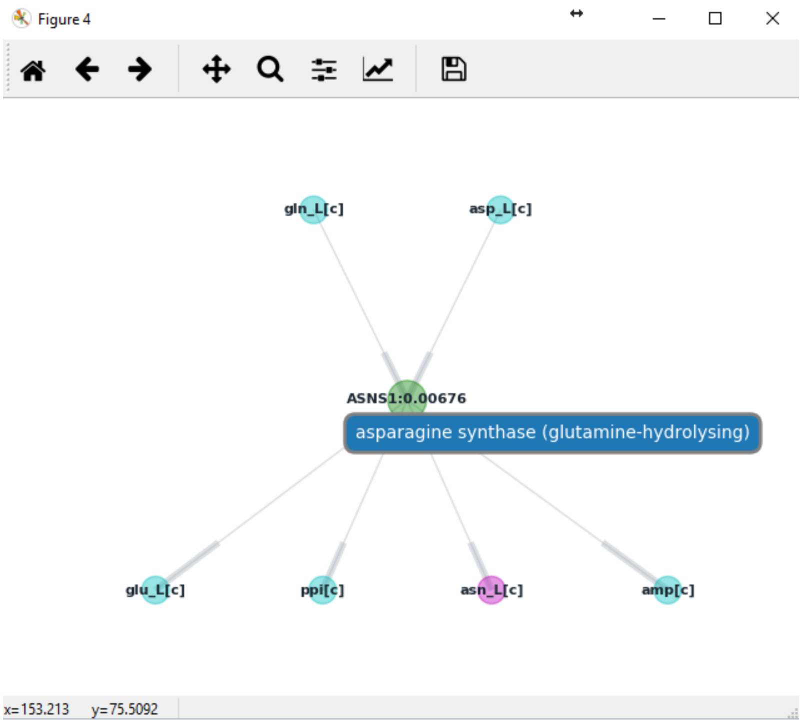


Figure 14: The reaction that becomes active after the removal of *asparagine* from the medium.

6. Pathway alignment and its visualisation

Grohar allows you to automatically identify a specified metabolic pathway within the model. In this case pathway represents a *source*, which is aligned with the *destination* model. Alignment and visualisation of the pathway in the model can be performed within the *Alignment* tab of the main window

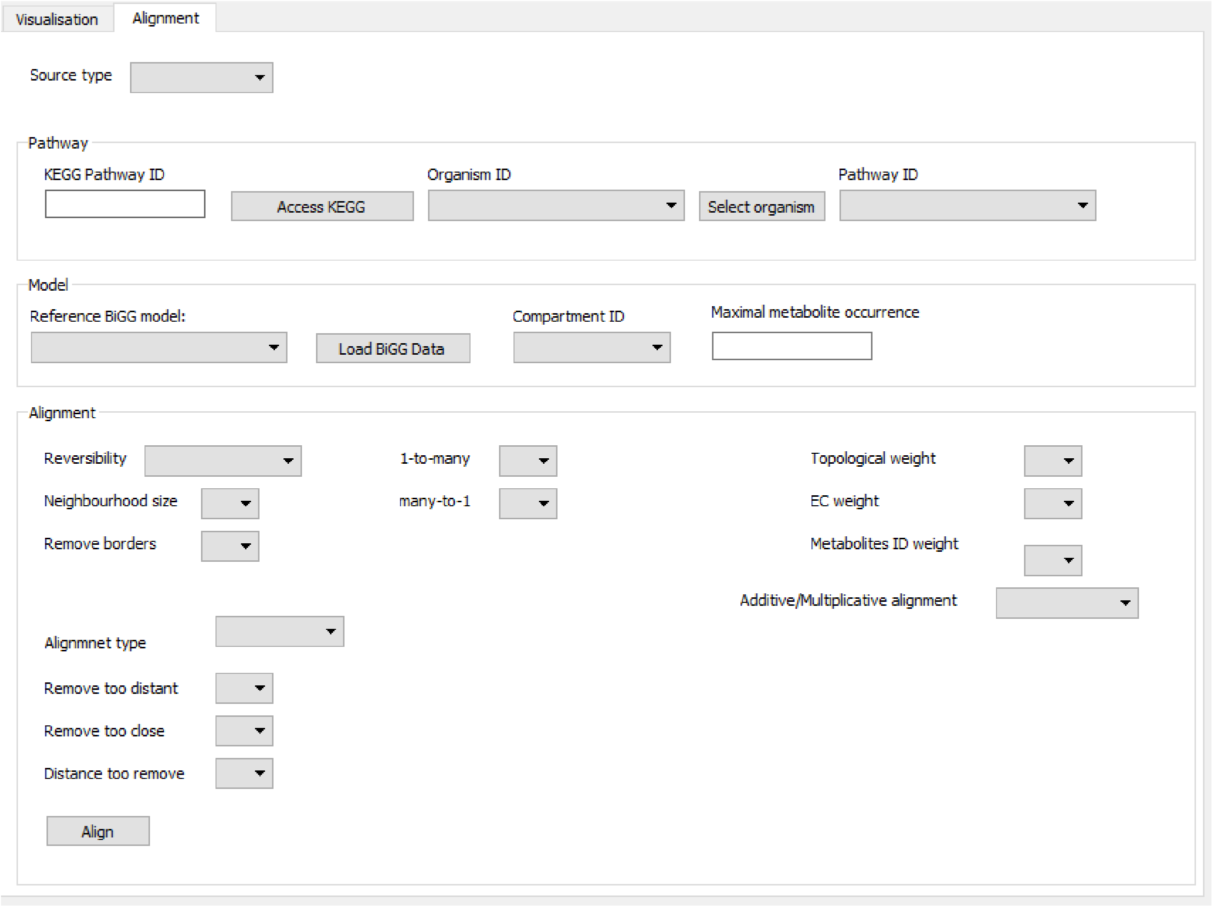


Figure 15: *Alignment* tab provides the features for pathway alignment and its visualisation. Note that all controls are disabled before the model is loaded.

Selecting a pathway

All controls within the *Alignment* tab are disabled before the model is loaded (see Section 1). After the model is loaded you can import a metabolic pathway from Matlab (*.mat) or SBML (*.xml) file in COBRA dialect or directly from KEGG database (Kanehisa, et al., 2006) (requires internet connection). Pathway source can be selected with the *Source type* combo box (see Figure *16*).

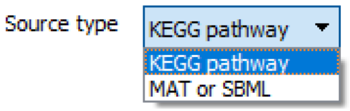


Figure 16: *The* pathway source can be selected with the Source type combo box.

If *KEGG pathway* is selected, controls for loading the pathway from the KEGG database are enabled within the *Pathway* group box (see Figure *17*).

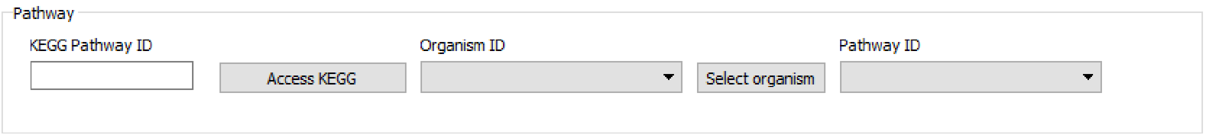


Figure 17: The c*ontrols for selecting a pathway from KEGG database.*

You can type the KEGG pathway ID directly in the *KEGG Pathway ID* field. You can also access the KEGG database and browse through Organisms and Pathways with the *Organism ID* and *Pathway ID* combo boxes. When you click the *Access KEGG* button, the *Organism ID* combo box is populated. After you select an organism and click the button *Select organism* the *Pathway ID* combo box is populated.

If you select *MAT or SBML* from the *Source type* combo box, controls for loading a pathway from a file are enabled within the *Pathway* group box (see Figure *18*).

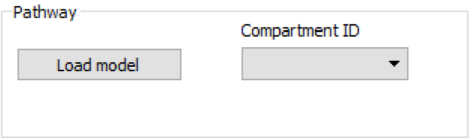
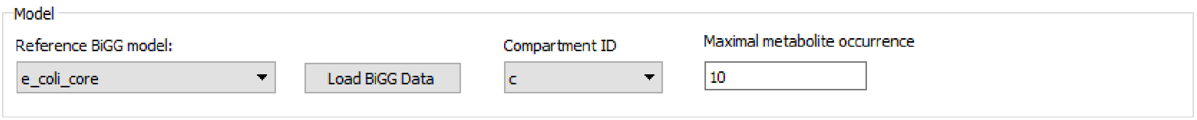


Figure 18: Controls for selecting a pathway from a MAT or SBML file.

These controls allow you to load the model of the pathway from a MAT or SBML file and to also select a compartment of interest (*Compartment ID* combo box).

Preparing the model

There are some parameters that you can set on the main model before the alignment. These parameters can be set by the controls within the *Model* group box (see Figure *19*).



Reference BiGG model: e_coli_core

Compartment ID c

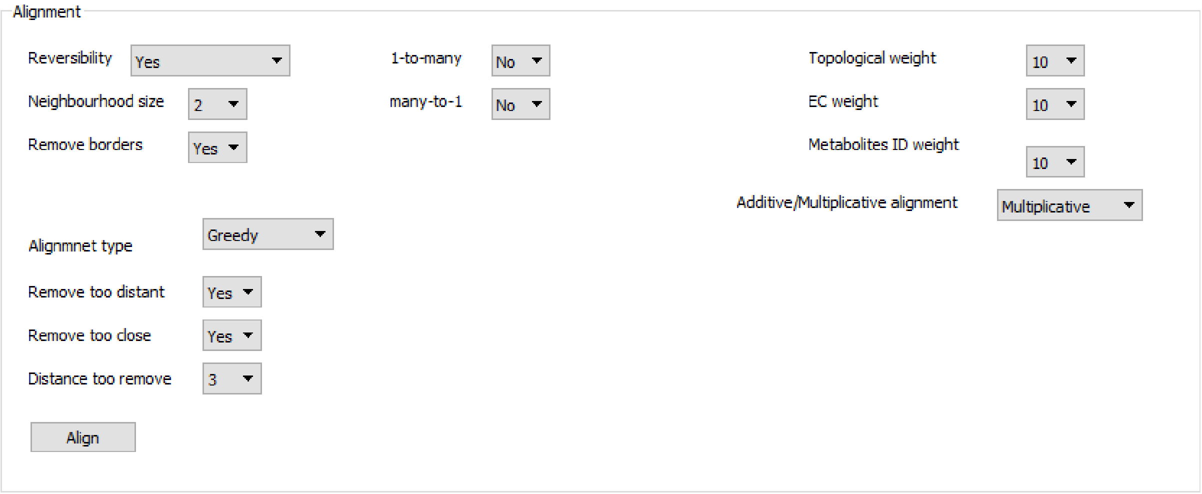
Maximal metabolite occurrence 10

Figure 19: The controls for setting the model parameters before the alignment.

First of all, you can select a reference model from the BiGG database (King, et al., 2016). Data from this model are used later on by the alignment algorithms. When you click the *Load BiGG Data* button the BiGG data, such as EC numbers, will be downloaded from the BiGG database. Data are stored locally so the internet connection is required only for the first time the model is selected. You can also select a compartment ID to limit the reaction set on which the alignment is performed (*Compartment ID* combo box). You can omit the metabolites that are present in too many reactions with the *Maximal metabolite occurrence* field.

Performing the alignment

You can set the alignment properties within the *Alignment* group box (see Figure *20*) after the pathway and model parameters were set.



Alignment

Reversibility Yes

Neighbourhood size 2

Remove borders Yes

1-to-many No

many-to-1 No

Topological weight 10

EC weight 10

Metabolites ID weight 10

Additive/Multiplicative alignment Multiplicative

Alignmnet type Greedy

Remove too distant Yes

Remove too close Yes

Distance too remove 3

Figure 20: The *Alignment* group box contains the controls for setting the alignment properties and performing the alignment.

The following controls are dedicated to this purpose:

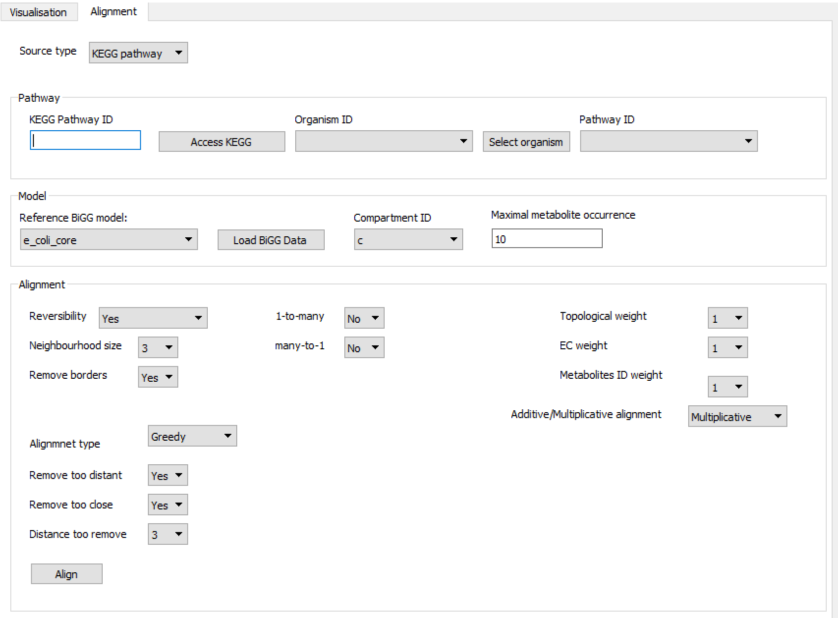
- *Reversibility*: you can choose among *Yes* and *No*. The parameter defines if the reversibility should be accounted for when performing the alignment.
- *Neighbourhood size*: you can define the maximal distance that is accounted for in the alignment process. If distance is set to 1, the neighbourhood will only include the adjacent nodes. If distance is set to 2, the neighbourhood will also include nodes that are adjacent to the adjacent nodes, etc.
- *Remove borders*: you can choose among *Yes* and *No*. The parameter should be set to *No* if aligning model to model, and should be set to *Yes* if aligning pathway to model (borderline properties of the reactions are omitted in this case).
- *1-to-many*: you can choose among *Yes* and *No*. If set to *Yes,* single reaction from the pathway, can be aligned with multiple reactions from the model.
- *many-to-1*: you can choose among *Yes* and *No*. If set to *Yes,* multiple reactions from the pathway can be aligned with a single reaction from the model.
- *Topological weight*: you can select a weight that is assigned to the algorithm calculating the topological similarity between the reactions. An extension of the MPBR algorithm is used for the calculation of topological similarity (Huang, Zhong, Lin, & Huang, 2016).
- *EC weight*: you can select a weight that is assigned to the algorithm calculating the similarity between the reactions on the basis of EC numbers. EC numbers are obtained from the BiGG or KEGG database. Data are stored locally so the internet connection is required only for the first time the model/pathway is selected. EC numbers comparison is performed according to (Heymans & Singh, 2003).

- *Metabolites ID weight*: you can select a weight that is assigned to the algorithm calculating the similarity between the reactions on the basis of reactants' and products' similarities. If KEGG pathway is used, KEGG metabolite identifiers are used in the comparison, similarly as in BiKEGG (Jamialahmadi, Motamedian, & Hashemi-Najafabadi, 2016). KEGG metabolite identifiers are obtained from the BiGG database for MAT and SBML models. Data are stored locally so the internet connection is required only for the first time the model is selected. If MAT or SBML pathway is used, metabolite identifiers from the COBRA model are used in the comparison.
- *Additive/Multiplicative alignment*: you can choose either the results of different algorithms should be summed (*Additive*) or multiplied (*Mulitplicative*).
- *Alignment type*: you can choose among three options for performing the alignment:
  - *Greedy*: always select the best maximal score alignment (among the maximal values select the one that has the least maximal alternatives).
  - *Simple*: select the first maximal score alignment.
  - *Probabilistic*: alignment score defines the probability that the alignment is selected. Selection is performed with the roulette rule.
- *Remove too distant*: if set to Yes – when two reactions are aligned the algorithm will remove all candidates for alignment that project a reaction that is adjacent to a source reaction to a reaction that is too far from the destination reaction (distance is larger than parameter *Distance to remove*).
- *Remove too close*: if set to Yes – when two reactions are aligned the algorithm will remove all candidates for alignment that project a reaction that is too distant to a source reaction (distance is larger than parameter *Distance to remove*) to a reaction that is adjacent to the destination reaction.
- *Distance to remove*: see above.

You can perform the alignment and its visualisation with the click on the button *Align* after the model and the pathway are loaded and after all the parameters have been set.

Example

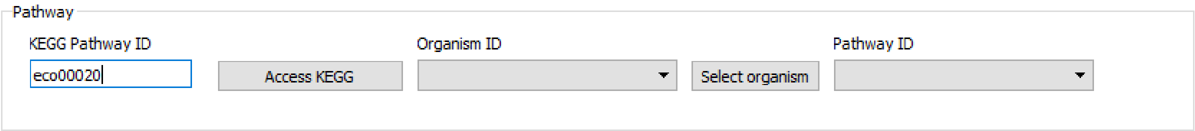
We will perform an alignment of the *E. coli* pathway *eco00020* from KEGG on the model of *E. coli core metabolism* from BiGG (http://bigg.ucsd.edu/models/e_coli_core). First, we will load the model through *File* → *Open* menu item. Next, we have to switch to the *Alignment* tab (see Figure *21*).



Source type: KEGG pathway
KEGG Pathway ID:
Organism ID:
Pathway ID:
Reference BiGG model: e_coli_core
Compartment ID: c
Maximal metabolite occurrence: 10
Reversibility: Yes
Neighbourhood size: 3
Remove borders: Yes
1-to-many: No
many-to-1: No
Topological weight: 1
EC weight: 1
Metabolites ID weight: 1
Additive/Multiplicative alignment: Multiplicative
Alignmnet type: Greedy
Remove too distant: Yes
Remove too close: Yes
Distance too remove: 3

Figure 21: The *Alignment* tab after the model is loaded.

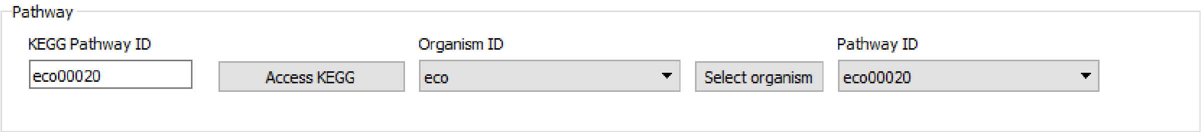
We need to load an appropriate pathway. We can leave the *Source type* combo box on its default selection (*KEGG pathway*). The controls for loading the pathway from the KEGG database are thus enabled within the *Pathway* group box. In the *KEGG Pathway ID* field we can simply type in *eco00020* (see Figure *22*).



KEGG Pathway ID: eco00020
Organism ID:
Pathway ID:

Figure 22: Manual entry of the KEGG pathway ID.

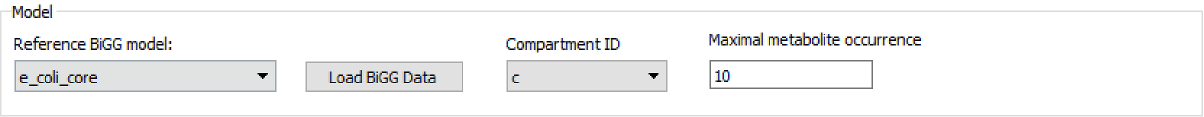
Alternatively, we can Access the KEGG database and find the pathway ID there. We click the *Access KEGG* button, select *eco* as the *Organism ID*, click the *Select organism* button and select the *Pathway ID eco00020* (see Figure *23*). It might take some time to load the data since they are acquired from the KEGG database (internet connection is required).



KEGG Pathway ID: eco00020
Organism ID: eco
Pathway ID: eco00020

Figure 23: The selection of the of KEGG pathway ID from the KEGG database.

Next, we will set the model parameters. Grohar uses COBRA model ID to find the BiGG model with the appropriate identifier within the *Reference BiGG model* combo box so only thing we need to do is to click the *Load Bigg Data* button. These data will be downloaded from the BiGG database. The data will be stored locally so the next time we will use the same model the procedure will be much faster. We will leave the other settings in the *Model* group box as they are (see Figure *24*).



Model

Reference BiGG model: e_coli_core ▾

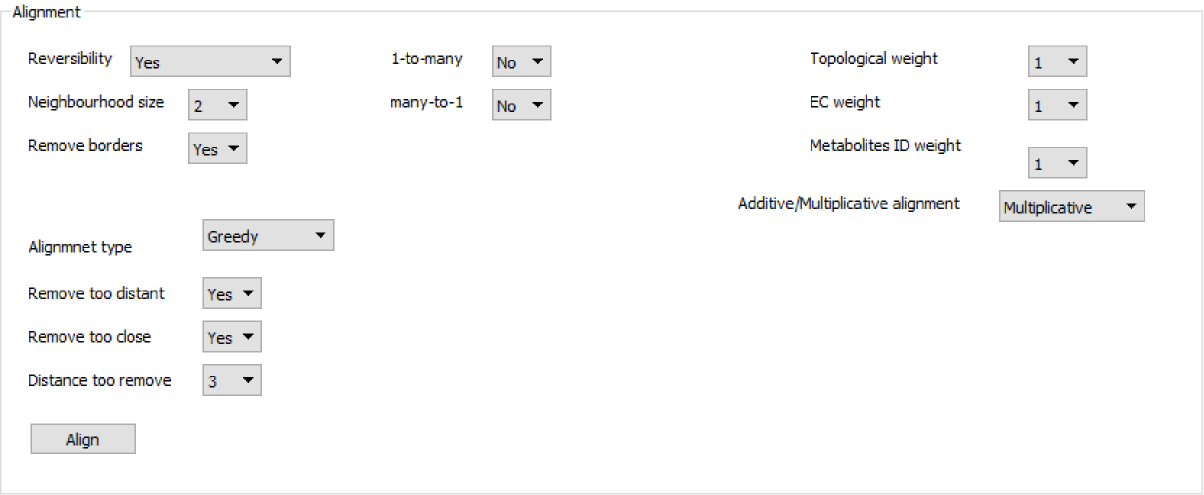
Load BiGG Data

Compartment ID: c ▾

Maximal metabolite occurrence: 10

Figure 24: The model settings.

Last, we need to set the parameters of the alignment procedure. We can leave these settings as they are as well (see Figure *25*).



Alignment

Reversibility: Yes ▾

Neighbourhood size: 2 ▾

Remove borders: Yes ▾

1-to-many: No ▾

many-to-1: No ▾

Topological weight: 1 ▾

EC weight: 1 ▾

Metabolites ID weight: 1 ▾

Additive/Multiplicative alignment: Multiplicative ▾

Alignmnet type: Greedy ▾

Remove too distant: Yes ▾

Remove too close: Yes ▾

Distance too remove: 3 ▾

Align

Figure 25: The alignment settings.

After the *Align* button is pressed KEGG pathway and its properties will be downloaded from the KEGG database. These data will be stored locally so the procedure will be much faster the next time we will be using the same pathway. After the alignment is performed the results will be visualised in a separate window in a similar manner as described in Sections 4 and 5 (see Figure *26*).

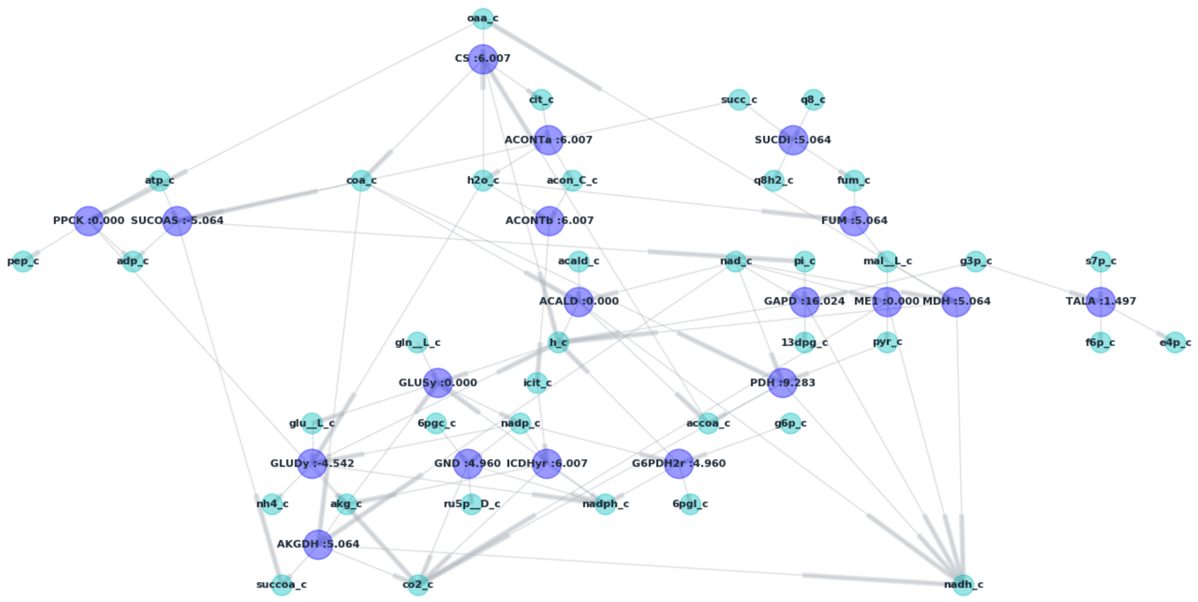


Figure 26: Visualisation of the alignment performed on the E. coli core metabolism model.

7. Bibliography

Heymans, M., & Singh, A. K. (2003). Deriving phylogenetic trees from the similarity analysis of metabolic pathways. *Bioinformatics, 19*, i138-i146. doi:10.1093/bioinformatics/btg1018

Huang, Y., Zhong, C., Lin, H. X., & Huang, J. (2016). Aligning Metabolic Pathways Exploiting Binary Relation of Reactions. *PLOS ONE, 11*, e0168044. doi:10.1371/journal.pone.0168044

Jamialahmadi, O., Motamedian, E., & Hashemi-Najafabadi, S. (2016). BiKEGG: a COBRA toolbox extension for bridging the BiGG and KEGG databases. *Molecular BioSystems, 12*, 3459-3466. doi:10.1039/c6mb00532b

Kanehisa, M., Goto, S., Hattori, M., Aoki-Kinoshita, K. F., Itoh, M., Kawashima, S., . . . Hirakawa, M. (2006). From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Research, 1*, D354--7.

King, Z. A., Lu, J., Dräger, A., Miller, P., Federowicz, S., Lerman, J. A., . . . Lewis, N. E. (2016). BiGG Models: A platform for integrating, standardizing and sharing genome-scale models. *Nucleic acids research, 44*, D515--D522.

Schellenberger, J., Que, R., Fleming, R. M., Thiele, I., Orth, J. D., Feist, A. M., . . . Palsson, B. O. (2011). Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox v2.0. *Nature protocols, 6*, 1290-1307.

Selvarasu, S., Ho, Y. S., Chong, W. P., Wong, N. S., Yusufi, F. N., Lee, Y. Y., . . . Lee, D.-Y. (2012). Combined in silico modeling and metabolomics analysis to characterize fed-batch CHO cell culture. *Biotechnology and Bioengineering, 109*, 1415-1429.