TRN-FBA user manual

TRN-FBA is a constraint-based modelling (CBM) tool able to perform simulation for Transcriptional Regulatory Metabolic Network (TRMN) models, a proposed model that combines transcriptional regulatory network (TRN) and metabolic network, as well as Genome-scale Metabolic Network (GSMN) models. Six types of CBM functions were implemented including flux balance analysis, flux variability analysis, knock-out analysis, transcriptional flux balance analysis, and transcriptional flux variability analysis; both reactions and genes can be interrogated for all types of analyses. TRN-FBA was developed in Python 2.7 as a command-line tool, while you can also run it in Python console using its functions after importing it as a package. This tool can be used in any operating system such as Windows, Linux, and Mac system.

REQUIREMENTS

- Python 2.7+; we suggest installing Anaconda distribution of Python, which allows you easily install the following packages using command conda.
- Pyomo; it is a Python-based, open-source optimization modelling language with a diverse set of optimization capabilities. We applied Pyomo for building linear programming models and access the LP solvers, so far only GLPK and Gurobi solvers have been tested. Here is Pyomo link: https://pyomo.readthedocs.io/en/latest/index.html# .
- GLPK; GLPK solver, which can be installed by following this link: https://pyomo.readthedocs.io/en/latest/installation.html .
- Gurobi; alternatively you can use Gurobi solver which is faster than GLPK solver. To install Gurobi please go to this link: https://www.gurobi.com/documentation/8.1/quickstart_windows/installing_the_anaconda_py.html.

INSTALLATION

You just need to download package source codes to your specified directory, and add this directory to your system PATH, and then you can use it straight away.

OPTIONS

The command line options are explained as follows:

-m	**(--model)**	**Model_file**	Input model file.
-e	**(--expr)**	**Expression_file**	Input expression file.
-p	**(--pfile)**	**Problem_file**	Input problem file, used to facilitate parameter settings such as objective, constraints, target list of genes/reactions, etc., problem file can be overwritten by those specified in command-line.
-a	**(--analysis)**	**obj [fba, fvaR, fvaG, koR, koG, tfbaR, tfbaG, tfvaR, tfvaG]**	Analysis types: obj - find optimized value of a defined objective function; fba - find a flux balance solution; fvaG/fvaR - flux variability analysis for genes/reactions; koG/koR - knockout analysis for genes/reactions;

			tfbaG/tfbaR - transcriptional flux balance analysis for genes/reactions; tfvaG/tfvaR - transcriptional flux variability analysis for genes/reactions.
-o	**(--obj)**	**objective**	Objective function; it will specify expression array ID for analyses of tfbaG/tfbaR and tfvaG/tfvaR.
-l	**(--lst)**	**Test_list**	List of target reactions/genes selected for tests, otherwise all reactions or genes are used.
-s	**(--solver)**	**Solver_name**	Specify a solver name such as glpk, gurobi; default is glpk2 which is glpk with pre-settings of 'dual' and 'nopresol'.
-i	**(--opt)**	**Solver_opts**	Options for the slover, e.g. -i "dual nopresol tmlim:6", it sets 3 options for glpk solver glpsol: "--dual --nopresol --tmlim 6".
-f	**(--outfile)**	**Output_file**	Output file name.
-c	**(--csts)**	**constraints**	Add variable constraint in the format "ID lb ub", e.g. -c "R1 1.5 10" means set reaction R1 lower bound to 1.5 and set upper bound to 10. This option can be used repeatedly to set multiple constraints.
-t	**(--eps)**	**Float_number**	Bounds tolerance for fva, ko and tfva.
-x	**(--extag)**	**External tag**	You need to set boudary/exteranl metabolite suffix when you define exchange reaction in the form, e.g., 'R_EX_e = R_EX_b' (default '_b'); ignore this option when set exchange reaction as 'R_EX_e = '.

USAGE EXAMPLE

1. Examples for command lines:

Find growth rate.

```
python trfba.py -m TRiJO1366.txt -a obj -o R_Ec_biomass_iJO1366_WT_53p95M
```

Find a FBA solution for all reactions and genes under optimal growth.

```
python trfba.py -m TRiJO1366.txt -a fba -o R_Ec_biomass_iJO1366_WT_53p95M –f
FBA_TRiJO1366.txt
```

Flux variability analysis for all reactions under optimal growth.

```
python trfba.py -m TRiJO1366.txt -a fvaR -o R_Ec_biomass_iJO1366_WT_53p95M –f
FVAR_TRiJO1366.txt
```

Flux variability analysis for all genes under optimal growth.

```
python trfba.py -m TRiJO1366.txt -a fvaG -o R_Ec_biomass_iJO1366_WT_53p95M –f
FVAG_TRiJO1366.txt
```

Flux variability analysis for selected 2 reactions of R_ALCD19 and R_AMANK under optimal growth.

```
python trfba.py -m TRiJO1366.txt -a fvaR -o R_Ec_biomass_iJO1366_WT_53p95M –f
FVAR_TRiJO1366.txt –l "R_ALCD19 R_AMANK"
```

Transcriptional flux balance analysis for all reactions under optimizing the consistency to the expression profile of WT_aerobic, and keeping positive growth rate.

```
python trfba.py -m TRiJO1366.txt -a tfbaR –e expr_aeroshift.txt –o WT_aerobic –f
TFBAR_TRiJO1366_WT_aerobic.txt -c "R_Ec_biomass_iJO1366_WT_53p95M 1 1000"
```

Transcriptional flux variability analysis for all reactions under optimizing the consistency to the expression profile of WT_aerobic, and keeping positive growth rate.

```
python trfba.py -m TRiJO1366.txt -a tfvaR –e expr_aeroshift.txt –o WT_aerobic –f
TFVAR_TRiJO1366_WT_aerobic.txt -c "R_Ec_biomass_iJO1366_WT_53p95M 1 1000"
```

Knock out essentiality analysis for all genes, where a selected list of target knock-out genes are put in problem file, and solver Gurobi is specified with option TimeLimit set to be 10second, and also set knock-out bounds to [0, 1e-8].

```
python trfba.py -m TRiJO1366.txt -a koG -o R_Ec_biomass_iJO1366_WT_53p95M –p prob_ko –f
KOG_TRiJO1366.txt –s gurobi -i TimeLimit:10 –t 1e-8
```

2. Examples for Python console:

First you need to import TRN-FBA package into the working console.

```
>>>import TRN-FBA.analysis as ta
```

Read model file in.

```
>>>mod = ta.readModel('TRiJO1366.txt', id='ecoli', name='E. coli TRMN model')
```

Read problem file in.

```
>>>prob = ta.readProblem('Problem_file')
```

Read expression file in.

```
>>>expr = ta.readExprData('Expression_file')
```

Get model component information.

```
>>>mod.reactions['R_MINCYCtpp'].show()
>>>mod.erules['E609'].show()
>>>mod.genes['acrA'].show()
>>>mod.trules['acrA'].show()
>>>mod.transcriptions['T_pepA.1'].show()
```

Perform flux balance analysis.

```
>>>fba = ta.Fba(mod, cmt='FBA analysis’)
>>>fba.solveMedia(obj='R_Ec_biomass_iJO1366_WT_53p95M')
>>>fba.write_fba(‘FBA_output_file’)
```

Perform Transcriptional flux variability analysis for reactions.

```
>>>tfva = ta.Tfva(mod, grp='reaction', eps=0, prob=prob, cmt='TFVA test')
>>>tfva.solveMedia(expr, 'WT_aerobic')
>>>tfva.write(‘TFVAR_output_file’)
```

Perform knock out essentiality analysis for genes.

```
>>>ko = ta.Ko(mod, grp='gene', cmt='KO test')
>>>ko.solveMedia(obj='R_Ec_biomass_iJO1366_WT_53p95M')
>>>ko.write('KO_output_file')
```

Get detailed solver stdout message.

```
>>>fba.optsolve(tee=True)
```

Set solver GLPK option tmlim to be 10 second.

```
>>>fba.setSolverOptions('glpk', 'tmlim', value=10)
```

Get Pyomo model component information.

```
>>>fba.pmod.myprint('R_MINCYCtpp')
```

Write Pyomo LP model out.

```
>>>fba.writePmodel2('Pyomo_model_LP')
```

FILE FORMAT

1. Model file

The model file is in tabular format, the columns are explained as follows. With '#' in the head of a line indicates comment line which will be ignored when reading model in.

C1	**ID**	Reaction/Interaction ID with predefined prefixes. 'R_' for all reactions, 'R_EX_' for exchange reactions, 'R_EF_' for TF conformation demand reactions, 'T_' for transcriptional interactions, and 'Te_' for TF conformation interactions.
C2	**Formula**	Formula for reactions/Interactions.
C3	**LB**	Lower bound for flux (reactions) or activity probability (interactions).
C4	**UB**	Upper bound for flux (reactions) or activity probability (interactions).
C5	**GPR/TF type/Effector type**	It is gene-protein-reaction (GPR) rules for reactions; it is TF type for transcriptional interaction, type '+' denotes activator and type '-' denotes repressor; it is effector type for TF conformation, type 'e+' means activating TF regulation and type 'e-' means inactivating TF regulation in the presence of related effectors.
C6	**Comment**	Comments for reactions/interactions

Reaction formula example:

```
2 M1 + M2 = 0.5 M3 + M4
```

where M1, M2 are reactant IDs in the left hand, M3 and M4 are product IDs in the right hand; numbers before molecule IDs are stoichiometric coefficients which is a positive float number (the coefficient 1 can be skipped). All molecule IDs in one side are grouped with operator '+', and connect two sides with operator '=', all numbers and IDs and operators are delimited with one space.

Transcriptional interaction formula example:

```
TF1 + TF2 -> 0.9 G1 + 0.6 G2
```

where TF1 and TF2 are transcription factor IDs in the left hand, G1 and G2 are target gene IDs in right hand; numbers before gene IDs are activity probabilities (the probability 1 can be skipped). All IDs in one side are grouped with operator '+', and connect two sides with operator '->', all numbers and IDs and operators are delimited with one space.

TF conformation formula example:

```
TF + ( M1 ) -> TF(M1)
```

where in the left hand TF is transcription factor ID, and M1 are effector wrapped by brackets, they are grouped with operator '+'. The right hand is TF-effector complex ID linked with operator '->'. All IDs, brackets and operators are delimited with one space.

Gene-protein-reaction (GPR) rules are supposed in disjunctive normal form (DNF), for example:

```
( G1 AND G2 ) OR (G3 AND G4)
```

where G1, G2, G3 and G4 are gene IDs, this rule could be explained that target reaction can be catalysed either by enzyme complex from G1 and G2, or by enzyme complex from G3 and G4.

2. Expression file

The expression file is in simple tabular format, in which rows correspond genes and columns corresponds arrays. First column is for gene IDs and first row is for array IDs. The expression levels are normally discretized into 3 levels which are highly expressed (denoted 1), lowly expressed (denoted -1) and moderately expressed (denoted 0). An example of expression file is shown below.

```
Gene        WT_aerobic    WT_anaerobic
thrL            1             0
thrA            1             1
thrB            1             1
thrC            1             1
yaaX            0             0
  …             …             …
```

3. Problem file

The problem file is used to set parameters which provide a flexible way to store and set/modify parameters and you can also set batch test for different media. There are 8 parameter fields, each field start with '>' plus field name, e.g. '>ANALYSIS' then you can put parameter content in the same line after a TAB (here denoted as \TAB\); and some parameter content starts from a new line. Parameter fields and content formats are explained as follows.

Field name	Parameter content
>ANALYSIS	Analysis type (obj, fba, fvaR, fvaG, koR, koG, tfbaR, tfbaG, tfvaR or tfvaG).
>OBJSENSE	Optimization direction (maximize or minimize).
>OBJECTIVE	1) Objective function, defined as a linear expression including reaction or metabolite IDs, stoichiometric coefficients (the coefficient 1 can be skipped), and '+' operators, all space-delimited, e.g. ' 0.5 R1 + M1'. Note that the '-' operators are not allowed, so negative stoichiometric coefficients should be used instead, e.g. 'M1 + -0.5 R2'. 2) Expression array ID for analysis tfbaR, tfbaG, tfvaR or tfvaG.
>REACTIONS	List of target reactions for analysis, all space-delimited.
>GENES	List of target genes for analysis, all space-delimited.
>CONSTRAINTS	Start from a new line, each line corresponds a constraint in the form: formula\TAB\lower_bound\TAB\upper bound\TAB\comment, where the formula can be single reaction ID or as a linear expression similar to the objective function.
>SOLVER	Specify solver name in the same line; set solver options from a new line, each line corresponds an option in the form: option_id\TAB\option.
>MEDIA	Start from a new line, each line corresponds a media constraint in the form: medium_name\TAB\exchange_reaction_id\TAB\lower_bound\TAB\upper_bound.

With '#' in the head of a line indicates comment line which will be ignored when reading model in. An example of problem file is shown below.

```
#TRN-FBA problem file format
>ANALYSIS    obj[fba, fvaR, fvaG, koR, koG, tfbaR, tfbaG, tfvaR, tfvaG]
>OBJSENSE    maximize[minimize]  #comment
>OBJECTIVE   0.5 R1 + -1 R2 + R3 + M1
>REACTIONS   R1 R2 R3     #reactions
>GENES G1 G2 G3
>CONSTRAINTS
R1     0      10     #comment
R2     -1     1      #comment
```

```
R3 + 0.5 R4  -1     1      #comment
R5 + -0.6 R6 -1     1      #comment
>SOLVER      glpk   #set slover and parameters/options
msg_lev      GLP_MSG_ERR
meth   GLP_DUAL
tol_bnd      1e-6
tol_dj 1e-8
>MEDIA #set different substrate media for tests
medium1      R_EX_1 -10    10
medium1      R_EX_2 -10    10
medium2      R_EX_3 -10    10
medium3      R_EX_4 -10    10
```

4. Output file

Output files from all analyses are basically in tabular format. Column IDs are indicated in output files, which are explained in following table.

ID	Reaction/Interaction ID.
LB	Lower bound for flux (reactions) or activity probability (interactions).
UB	Upper bound for flux (reactions) or activity probability (interactions).
Value	Flux value for FBA analysis.
Growth	Optimal growth rate for KO analysis.
Ratio	Ratio equals knock-out growth rate divided by growth of original model for KO analysis.
Status **MIN_status** **MAX_status**	Solver's final optimizing status, where 'optimal' means optimal solution found, 'infeasible' means LP problem is infeasible, 'other' means other reason for no optimal solution.
MIN	Minimal value for FVA analysis.
MAX	Maximal value for FVA analysis.
Activity	Predicted reactions/genes activities for FVA analysis, where '1' means active, '-1' means inactive, '0' means undetermined.
State	Original expression level for TFVAG analysis.

COPYRIGHT

SEE ALSO

https://en.wikibooks.org/wiki/GLPK/Using_GLPSOL for Documentation about GLPK parameters.
http://www.gurobi.com/ for Documentation of Gurobi optimizer and its installation information.