SurreyFBA user manual

NAME

SurreyFBA (version 2.3) – GLPK/Gurobi-based LP/MIP solver for metabolic modelling.

SYNOPSIS

SurreyFBA -i model_file [-j expression_file] [-o obj_fun] [-a arg_list] [-x ext_list] [-X exteranl_tag] [-s solver_algorithm] [-p problem_type] [-b problem_file] [–u dpaplot_file] [-f output_file] [-g msg_level] [-l time_limit] [-t] [-m] [-c] [-w]

OPTIONS

–i (or **–input**) : model_file

read a metabolic model from a text file. Each line in the file (except the comment lines starting with '#') describes one reaction in the tab-delimited format "name equation lb ub [rule] [#comment]", where "name" is the reaction name, "equation" is the equation, "lb" and "ub" are the lower and the upper flux rate bounds, "rule" is an optional logical rule and "comment" is an optional comment. The equation consists of metabolite identifiers, stoichiometric coefficients (the coefficient 1 can be skipped), '+' and '=' operators, all space-delimited. The rule represents the association with the parents (enzymes or genes) and may include their IDs, AND and OR operators (case insensitive), and parentheses, all space-delimited. Here is an example of a line: "r1 2 H2 + O2 = 2 H2O 0 100 g1 and (g2 or g3) #comment". The metabolites whose identifiers start with 'X_' or 'x_' or end with '_ext' or '_xt' are considered as external.

-j (or **-expfile**) : expression_file

input gene/enzyme expression data file. For imat and fimat problem, the data should be pre-processed and discretized as highly (1), lowly (-1) and moderately expressed (0); for dpasig problem data should be in the form of log2 ratios of treatment and reference sample signals; for problem gimme and gimmefva, the expressions are pre-processed absolute expression levels (positive number). The file format is as follows: first row is titles and first column consists of gene names. From second column, each column is associated with a tissue name and their gene expression level. Columns are tab-delimited, with 'NA' denoting empty value.

–o (or **–objective**) : obj_fun

objective function, defined as a linear expression including reaction or metabolite names, stoichiometric coefficients (the coefficient 1 can be skipped), and '+' operators, all space-delimited, e.g. 'm1 + 0.5 r2'. Note that the '-' operators are not allowed, so negative stoichiometric coefficients should be used instead, e.g. 'm1 + -0.5 r2'. Here you can specify expression array for iMAT problem; and specify a biomass reaction for GNI problems. For gim3e problem, you can specify an objective reaction and a metabolic array, delimited by a comma.

–a (or **–arguments**) **:** arg_list

Space-delimited list of problem-specific arguments, see below.

–x (or **–externals**) **:** ext_list

Space-delimited list of external metabolites, overriding the list defined by the model.

–X (or **–Xtag**) **:** external tag

assigning external reactions with designated tag which will overwrite the ext_list.

–s (or **–solver**) : solver_algorithm

the solvers employed including GLPK and Gurobi. The valid solver algorithm options are:

simplex (GLPK) simplex algorithm (default)

exact (GLPK) simplex algorithm, followed by a multiprecision solver, preventing rounding errors.

milp [,mip_gap] (GLPK) MILP (mixed-integer linear programming) solver with parameter mip_gap tolerance of MILP (default mip_gap: 1e-6).

grb[,tol[,foc]] Gurobi solver which can automatically select simplex/milp for specific problem. Parameter `tol` is used to set Dual&Primal feasibility tolerance for Gurobi solver, tightening this tolerance can produce smaller constraint violations (default: 1e-6, Min: 1e-9, Max: 1e-2). Parameter foc is used to set MILP solution strategy, 4 integer values can be chosen which are 1: focus on finding feasible solutions quickly; 2: focus on proving optimality; 3: focus on moving objective bound; 0 (default): balancing between finding new feasible solutions and proving that the current solution is optimal. If MILP solver is very slow for a problem then try foc=3.

–p (or **–problem**) : problem_type

states the type of the problem or analysis. The valid problem type options and the corresponding argument list formats are:

show show the model contents (no args)

objvalue compute the optimal objective function value and show it (no args)

objstat compute the optimal objective function value, show it and the solution status, e.g. OPTIMAL, FEASIBLE or UNDEFINED (no args)

fba (Flux Balance Analysis) compute a flux distribution corresponding to an optimal value of the objective function (no args)

ko same as objstat, but the query reactions/genes are knocked out (args: space-separated list of reactions/genes)

fva (Flux Variability Analysis) compute the minimal and maximal flux rates of the query reactions, corresponding to the optimal objective function value (args: space-separated list of reactions; if not provided, all reactions are considered)

plot iteratively compute the optimal objective function values corresponding to the incrementing/decrementing flux rates of the query reaction (args: "reaction from to step", e.g. "r2 0 1 0.1")

plot3d iteratively compute the optimal objective function values corresponding to the incrementing/decrementing flux rates of two query reactions (args: "reaction1 from to step reaction2 from to step", e.g. "r2 0 1 0.1 r3 -1 1 0.2")

em compute an elementary flux mode involving the query reaction. The objective is to minimise the number of reactions involved (if the solver is MILP), or the sum of flux rates in the flux distribution (otherwise). The first argument is a reaction to be involved in the mode. The second argument is the number of modes to be computed (1 by default). If the solver is MILP, an additional scaling factor between 0 and 1 can be provided as the third argument, since all flux rates in the resulting distribution must be in this range. The default scaling factor is 0.001; note that decreasing it may result in longer solution times, while increasing may result in the impossibility of a solution. (args: "reaction [scale_factor]").

mss compute one or more minimal (i.e. irreducible) sets of external substrates which are sufficient for the objective function to achieve its optimum. E.g. if the objective function is biomass, the result would represent minimal

media, in which biomass can be synthesised at a maximal rate. The first argument is the allowed range of flux rates; the second argument is the number of sets to be calculated (1 by default) (args: 'reaction [range, number]').

mps compute one or more minimal (i.e. irreducible) sets of external products which can be released while the objective function has achieved its optimum. E.g. if the objective function is glucose uptake, the result in a fermenting organism would represent minimal sets of fermentation products at the maximal glucose uptake rate. The arguments are identical to those for minimal substrate sets.

live detect the live reactions, i.e. those able to carry a steady state flux (args: space-separated list of query reactions; if not provided, all reactions are considered).

ess detect the essential reactions, by blocking which the value of the objective function becomes zero (args: space-separated list of query reactions; if not provided, all reactions are considered).

uncons detect the unconserved metabolites; e.g. those whose molecular masses cannot be simultaneously assigned positive values (no args).

orphans
detect the orphan metabolites (i.e. those internal metabolites used by less than two reactions).

cc detect the weakly connected components.

imat[,threshold]
flux Activity Analysis by shilomi's method which gives a qualitative prediction about flux activity state for reactions and genes without requiring the biomass synthesis reaction. Apart from genome-scale model, imat only needs absolute gene expression data which should reflect the absolute transcript abundances of genes under one particular growth condition. Parameter 'threshold' is positive threshold of active flux which is used to judge if a reaction is active or not. If not set, the default value is 1. (args: space-separated list of query reactions and genes; if not provided, all reactions and genes are considered).

fimat
an alternative method to iMAT method which is much faster than iMAT method but only for reactions, the method relies on MILP. (args: space-separated list of query reactions; if not provided, all reactions are considered).

gimme[,threshold]
the objective function of GIMME method is to minimize the fluxes of lowly expressed reactions weighted by the deviations of reaction expression state from predefined threshold for low expression (default 12). The output is similar as FBA. The objective value is inconsistency score indicating the degree of disagreement between the gene expression data and the assumed objective function under specified required functionalities. (args: space-separated list of query reactions; if not provided, all reactions are considered).

gimmefva[,threshold]
the objective function is the same as gimme, but the output is similar as FVA.

gim3e predict reaction flux activities using metabolic constraints, the output is similar as FVA. (args: space-separated list of query reactions; if not provided, all reactions are considered).

dsgni predict strong Gene-Nutrient Interactions (sGNI) which uncover the dependence of gene essentiality on the presence/absence of nutrients given a growth medium of interest. The algorithm only needs a well-defined genome-scale model (args: biomass reaction; space-separated list of query genes; if not provided, all reactions are considered).

wgni[,samsize]
: predict Weak Gene-Nutrient Interactions (sGNI) using sampling approach in which a set of growth media is randomly sampled and normal FBA method are applied to predict gene knockout effects under each sampled medium. 'samsize' is sample size of media for wgni. For medium size (number of nutrients) less than 15, the algorithm searches the whole media space so as to get whole essential media space; otherwise the algorithm samples n random mediums. The number of sampled growth media for wgni problem which should be proportionate to the media space (2^n, where n is number of nutrients), is not set, default value is 3300. (args: biomass reaction; space-separated list of query genes; if not provided, all reactions are considered).

dpaplot
: find producibility plot for Differential Producibility Analysis (DPA). The goal is to identify a sets of genes that participate in the production/consumption of each metabolite. This is a binary matrix that links genes with metabolites on the basis of whether or not each gene is essential for production/consumption of each metabolite (no args).

dpasig
: from producibility plot produced by dpaplot and by incorporating microarray gene expression data, here we are able to compute a value called 'metabolite signal' which reflects the metabolic state. The metabolite signal can be calculated as the median of associated gene expression ratios of its associated up/down-regulated genes (no args).

All problems can be solved using GLPK solver, and currently Gurobi solver supports following problem: objvalue, objstat, fba, fva, ko, imat, fimat, sgni, gim3e, wgni and dpaplot.

–b (or **–bfile**) : problem_file
: read a problem description from a text file. The file consists of records of one type of problem listed above delimited by semicolons; each record describes one specific problem to be solved without rebuilding the underlying GLPK linear program. This enables efficient iterative solution of metabolic modelling problems of the same type applied to the same model; the solutions are printed sequentially and delimited with semicolons. Apart from comments (which may appear in any line, starting with a hash-symbol and occupying the rest of the line), a record may contain meaningful lines of the following four formats:

'!dir: obj_fun'
: where dir is either "min" for minimisation or "max" for maximisation, and obj_fun is the objective function, as defined above.

'![expression names] [nutrient names][objective,array]'
: for imat and fimat problems, here you can specify one expression name indicating a column of gene/enzyme expression profile will be used for this record of problem. For sgni and wgni problem, here nutrient names are indicated, which defines the space of growth media under which we perform prediction of GNIs. The nutrient names are single space-delimited. The names of nutrients are the same as the name of external metabolites of exchange reactions in model file. For gim3e problem, you can specify an objective reaction and a metabolic array column in metabolic signal profile, delimited by a comma.

'? arg_list'
: where arg_list is the argument list in a problem-specific format as defined above. The objective function and argument list apply to all subsequent records until redefined; they are overridden by the ones supplied with the command-line options.

'$[nutrient names]'
for sgni and wgni problem, in the rest nutrients excepting defined nutrients of growth medium, here we can designate the nutrients to be always present and rest nutrients will be set absent. If no nutrient designated, all nutrients excepting nutrients of growth medium will be set absent. Names are single space-delimited.

'\ ext_list'
where ext_list is a space-delimited list of external metabolites, overriding the lists defined by the model and the command line.

'exp lb ub'
where exp is a space-separated linear expression (e.g. a single reaction name), whose lower and upper bounds are defined by lb and ub, respectively (tab-separated from the expression and each other). If ub is skipped, the value is fixed to lb.

';' end of record; note that the semicolon does not need to be in a separate line.

The objective function, the argument list, and the list of external metabolites apply to all subsequent records until redefined. The constraints apply everywhere once defined. Note that problem imat, fimat, sgni, wgni, dpaplot and dpasig can only have one record.

-u (or **-dpafile**) : dpaplot_file
the output file of problem dpaplot. This file is only used for problem dpasig. This file provides the mapping from genes to metabolite to that will be used to calculate the metabolite signals.

–f (or **–file**) : output_file
output file (standard output stream by default).

–g (or **–msg**) : msg_level
the level of verbose output during solution. The valid options are:
off no output (default)
on normal output
err error and warning messages only
all full output (including informational messages)

–l (or **–limit**) : time_limit
limit the solution time to time_limit, which is the time in milliseconds

–t (or **–trim**)
reduce the stoichiometry matrix using the topological 'trimming' algorithm: remove reactions involving orphan metabolites, repeat until no orphan metabolites remain.

–m (or **–minimise**)
sets the optimisation direction to minimisation (it is maximisation by default)

–c (or **–comments**)
include comments into the output

–w (or **–write**)
write the linear program data in CPLEX LP format into the file "LP.txt" in the working directory.

SYSTEM REQUIREMENTS

SurreyFBA2.3 so far has been successfully tested on platforms of Linux, Windows and Mac. Two versions of software are provided, one is called sfba-glpk which only uses GLPK library, and the other is called sfba-grb which uses GLPK and Gurobi libraries, note that before use this version you have to install licensed Gurobi solver.

REPORTING BUGS

Please report bugs to <h.wu@surrey.ac.uk>.

COPYRIGHT

Copyright © 2012 Albert Gevorgyan and Andrzej Kierzek and Huihai Wu. This program is free software; you may redistribute it under the terms of the GNU General Public License. This program has absolutely no warranty.

SEE ALSO

Documentation about GLPK and its C API.
See also the official GNU webpage dedicated to GLPK at
http://www.gnu.org/software/glpk/glpk.html.
Documentaion of Gurobi optimizer and its installation information at
http://www.gurobi.com/