Analysis if ~omics data in the context of Genome Scale Metabolic Networks.

MUFINS manual

1. Usage of iMAT method and alternatives.

Synopsis

```
./sfba  -i model_file  -X external_tag  -j gexp_file  -p problem_type  -b problem_file
[-o array] [-s solver] [-c] [-f output_file]
```

Description

`-i model_file` : input model file in MUFINS reaction table format.

`-j gexp_file` : input gene expression data file
For iMAT and Fast iMAT methods (-p imat, -p fimat), gene expression data should be pre-processed and discretized as 3 levels: highly (1), lowly (-1) and moderately expressed (0). Here is an example:

```
Gene          Tissue1       Tissue2
g1             1             1
g2             1            -1
g3            -1             1
…
```

Where, first column consists of gene names. Each of the following columns is associated with a gene expression profile for particular condition. Since this method is has been initially formulated to generate "tissue-specific models", the columns are labelled as "Tissue*" in JyMet interface. Columns are tab-delimited, with 'NA' denoting empty value. The expression data can also be enzyme expression level (protein abundances) which is in the same way as the gene expressions. For GIMME methods (-p gimme and –p gimmefva), the expressions are pre-processed absolute expression levels (positive numbers).

`-p problem_type` : analysis method
The following methods are implemented:
`imat[,thr]` : iMAT method where `thr` is positive threshold for active flux, used to establish whether reaction is active or not. If not set, the default value is 1.
`fimat`: Fast iMAT approach (see MUFINS manuscript).
`gimme[,thr]`: 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 `thr` (default 12) for low expression. The output is similar as FBA.
`gimmefva[,thr]`: the objective function is the same as gimme, but the output is in FVA format.

`-b problem_file` : input problem file
Problem file follows the format of FBA described in MUFINS manual (see sfba.pdf). The following parameters are specific to ~omics data analysis:
`![expression name]` - In ~omics data analysis approaches objective function represents congruency between data and flux distribution, rather than reaction flux. Thus we use the name of ~omics data sample (Tissue) in place as objective function. For gimme and gimmefva, the problem should define reaction bounds which define Required Metabolic Functionalities (RMF) that the cell is assumed to achieve, such as for example growth rate where the biomass reaction should be fixed at specific value.
`?[reaction name and gene name]` - perform analysis only for specific reactions and genes. If no reaction or gene designated, all reactions and genes will be investigated. Names are single space-delimited.

`-o array` : expression array
Without the problem file you can directly specify which expression array column in expression table you would like to analyse.

`-s solver` : solver and algorithms
For GLPK solver you can choose `simplex`, `exact` or `milp[,mip_gap]` with mip_gap indicating the tolerance of MILP (default mip_gap: 1e-6). The default solver is GLPK's simplex.
You can use Gurobi solver by set option as `-s grb[,tol[,foc]]` where 2 parameters can be set. 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`. Because Gurobi's MILP is much efficient than GLPK's MILP, so it is better choose Gurobi for problems which need MILP solver (such as imat and fimat) on big models.

Example

Run following commands:

```
./sfba -i model.sfba -X _xt -j gexp -p imat -o Tissue1 –s grb -c -f out_imat
./sfba -i model.sfba -X _xt -j gexp -p fimat -o Tissue1 –s grb -c -f out_fimat
./sfba -i model2.sfba -X _xt -j gexp_level -p gimme,12 -b problem_gimme -c -f
out_gimme_gni
./sfba -i model2.sfba -X _xt -j gexp_level -p gimmefva,12 -b problem_gimme -c -f
out_gimmefva_gni
```

Output for imat:

```
 Flux-activity analysis results for expression profile of Tissue1


 Flux-activity state for reactions:
 Reac(direc)  Active        Inactive     ACT_LEV  State    Comment
 R1  (+)      4: OPTIMAL    2: OPTIMAL   2        1        Reaction    1
 R1  (-)      0: UNDEFINED  5: OPTIMAL   -5       1        Reaction    1
 R2  (+)      4: OPTIMAL    2: OPTIMAL   2        1        Reaction    2
 R3  (+)      4: OPTIMAL    1: OPTIMAL   3        -1       Reaction    3
 R4  (+)      4: OPTIMAL    4: OPTIMAL   0        1        Reaction    4
 R5  (+)      4: OPTIMAL    4: OPTIMAL   0        1        Reaction    5
 R5  (-)      0: UNDEFINED  5: OPTIMAL   -5       1        Reaction    5
 ///
 Flux-activity state for genes:
 Number of genes: 5
 Gene         Active        Inactive     ACT_LEV  EXP_LEV  Postregulation
 g1           4: OPTIMAL    2: OPTIMAL   2        1        no
 g2           4: OPTIMAL    2: OPTIMAL   2        1        no
 g3           4: OPTIMAL    1: OPTIMAL   3        -1       up
 g4           4: OPTIMAL    4: OPTIMAL   0        1        no
 g5           4: OPTIMAL    4: OPTIMAL   0        1        no
 ;
```

Explanation:
Symbol (+) indicates the forward direction of reaction, that is, from left side to right side of the reaction formula in the model, whereas (-) indicates the reverse direction. `Active` denotes the maximum similarity when reaction forced to be active. `Inactive` denotes the maximum similarity when reaction forced to be

inactive. ACT_LEV denotes the activity state of reactions, that is, `Active - Inactive`. If ACT_LEV >0, reaction is predicted to be active; if ACT_LEV <0 reaction is predicted to be inactive; if ACT_LEV =0, reaction is predicted to be undetermined. For reversible reactions, the directionality can be judged by comparing the ACT_LEV of both directions, for example, ACT_LEV of R1 (+) is 2 against -5 of R1 (-), so reaction R1 prefers the forward direction. State column denotes reaction state determined by genes expressions and rules, where 1 means highly activated; -1 means lowly activated; 0 means moderately activated.

Flux-activity state for genes can be used to test if the genes have post-translational regulations. For example, the expression level of g3 is -1 (lowly expressed), while its ACT_LEV value is 3 (active), so the corresponding gene could be post-up-regulated. The state of reactions can also give post-regulation information, for example R3 has STATE of -1 (lowly expressed) and its ACT_LEV is 4, so the reaction must be post-up-regulated through its regulation genes or enzyme.

Output for fimat:

```
#666666: OPTIMAL (R1 + R1__r + R2 + -1 R3 + R4 + R5 + R5__r)
Reac(direc)   Minimum            Maximum          Activity  State  Comment
R1            500000: OPTIMAL    500000: OPTIMAL     1       1     #Reaction 1
R1__r         0: OPTIMAL         0: OPTIMAL         -1       1     #R1:Reverse
R2            500000: OPTIMAL    500000: OPTIMAL     1       1     #Reaction 2
R3            999999: OPTIMAL    999999: OPTIMAL     1      -1     #Reaction 3
R4            333333: OPTIMAL    333333: OPTIMAL     1       1     #Reaction 4
R5            333333: OPTIMAL    333333: OPTIMAL     1       1     #Reaction 5
R5__r         0: OPTIMAL         0: OPTIMAL         -1       1     #R5:Reverse
;
```

Explanation:
First row shows the maximized optimal flux `666666` and its state `OPTIMAL` on the objective function `max(R1 + R1__r + R2 + -1 R3 + R4 + R5 + R5__r)`. Column `State` denotes reaction state determined by genes expressions and rules and the coefficients are determined by these states. After fixing the nonzero-state reactions' fluxes, the normal FVA then be applied to each zero-state reaction. Let predicted flux range be [Fmin, Fmax], if Fmin >0 or Fmax<0 then reaction is predicted to be active (1); if Fmin = Fmax = 0 then reaction is predicted to be inactive (-1); otherwise reaction is predicted undetermined (0). By this reaction `R1, R2, R3, R4` and `R5` are active, and reverse reactions `R1__r` and `R5__r` are inactive which is consistent with the results of `imat`. Reactions with suffix '__r' are split reverse reactions for nonzero state reversible reactions.

Output for gimme:

```
#75: OPTIMAL (3 R6 + 3 R6__r)
BIOMASS       50     0      #Biomass
GROWTH        50     0      #Growth
NR1           0      0      #Nutrient M1
NR2           100    0      #Nutrient M2
NR3           25     0      #Nutrient M3
NR4           0      0      #Nutrient M4
R1            50     100    #Reaction 1
R2            25     100    #Reaction 2
R2__r         0      100    #R2:Reverse
R3            0      100    #Reaction 3
R3__r         0      100    #R3:Reverse
R4            0      0      #Reaction 4
R5            25     200    #Reaction 5
R5__r         0      200    #R5:Reverse
R6            25     9      #Reaction 6
R6__r         0      9      #R6:Reverse
```

```
;
```

Explanation:
First row shows the minimized optimal flux `75` and its state `OPTIMAL` on the objective function `minimize (3 R6 + 3 R6__r)` under maximal growth (biomass fixed at 50). The value 75 is inconsistency score indicating the degree of disagreement between the gene expression data and the assumed objective function under specified required functionalities. Second column is non-unique flux for each reaction; third column is the reaction state determined by genes expressions and rules using the same mapping rule as iMAT method. Reactions with suffix '`__r`' are split reverse reactions for all reversible reactions.

Output for gimmefva:

```
#75: OPTIMAL  (3 R6 + 3 R6__r)
Reac(direc)   Minimum       Maximum       Activity      State  Comment
BIOMASS       50: OPTIMAL   50: OPTIMAL         1       0      #Biomass
GROWTH        50: OPTIMAL   50: OPTIMAL         1       0      #Growth
NR1           0: OPTIMAL    25: OPTIMAL         0       0      #Nutrient M1
NR2           100: OPTIMAL  100: OPTIMAL        1       0      #Nutrient M2
NR3           0: OPTIMAL    25: OPTIMAL         0       0      #Nutrient M3
NR4           0: OPTIMAL    25: OPTIMAL         0       0      #Nutrient M4
R1            50: OPTIMAL   50: OPTIMAL         1       100    #Reaction 1
R2           0: OPTIMAL   100: OPTIMAL          0       100    #Reaction 2
R2__r         0: OPTIMAL    100: OPTIMAL        0       100    #R2:Reverse
R3            0: OPTIMAL    100: OPTIMAL        0       100    #Reaction 3
R3__r         0: OPTIMAL    100: OPTIMAL        0       100    #R3:Reverse
R4            0: OPTIMAL    0: OPTIMAL          -1      0      #Reaction 4
R5            25: OPTIMAL   100: OPTIMAL        1       200    #Reaction 5
R5__r         0: OPTIMAL    75: OPTIMAL         0       200    #R5:Reverse
R6            25: OPTIMAL   25: OPTIMAL         1       9      #Reaction 6
R6__r         0: OPTIMAL    0: OPTIMAL          -1      9      #R6:Reverse
;
```

Explanation:
The results format is the same as output of fimat and the judgement of reaction activities is also the same as fimat method. The fluxes of gimme output are contained within the flux range of gimmefva output. State column denotes reaction state determined by genes expressions and rules, with larger number implying greater certainty that reaction is present.

2. Usage of flux activity analysis by GIM3E method

With GIM3E method, we can exploit metabolomics data to further constrain the solution space of flux variability analysis and thus to get more condition-specific predictions on reaction activities.

Synopsis

```
./sfba  -i model_file  -X external_tag  -j msig_file  -p problem_type  -b problem_file
[-o objective,array] [-s solver] [-c] [-f output_file]
```

Description

`-j msig_file` : Input metabolic signal data file
This is discretized data indicating whether a metabolite is detected (1) or not (0) for a set of metabolites of interest. Here is an example:

```
metabolite          arr1          arr2
m1                    1             1
m2                    1             0
m3                    0             1
…
```

Where, first column consists of metabolite names. From second column, each column is associated with a metabolic signal profile. Columns are tab-delimited, with ‘NA’ denoting empty value.

`-o objective,array` : objective reaction and metabolic signal array
Without the problem file you can directly specify the objective reaction (often biomass reaction) to be maximized and specify which metabolic signal profile in metabolic signal table you would like to analyse, delimited by a comma.

`-b problem_file` : input problem file
Problem file follows the format of FBA described in MUFINS manual, excepting follows.
`![objective,array]` -specify the objective reaction and metabolic array column in metabolic signal table you would like to analyse, delimited by a comma.
`?[reaction name]` - perform analysis only for appointed reactions. If no reaction designated, all reactions will be investigated. Names are single space-delimited.

Example

Run following commands:

```
./sfba -i model2.sfba -X _xt -j msig -p gim3e -o BIOMASS,arr1 –s grb -c -f out_gim3e
```

Output for gim3e:

```
#50: OPTIMAL (BIOMASS)
BIOMASS 49.5: OPTIMAL   50: OPTIMAL     ACTIVE  #Biomass
GROWTH  49.5: OPTIMAL   50: OPTIMAL     ACTIVE  #Growth
NR1     0: OPTIMAL      50: OPTIMAL     UND     #Nutrient M1
NR2     99: OPTIMAL     100: OPTIMAL    ACTIVE  #Nutrient M2
NR3     1.01e-006: OPTIMAL      25: OPTIMAL     ACTIVE  #Nutrient M3
NR4     0: OPTIMAL      25: OPTIMAL     UND     #Nutrient M4
R1      49.5: OPTIMAL   50: OPTIMAL     ACTIVE  #Reaction 1
R2      1.01e-006: OPTIMAL      25: OPTIMAL     ACTIVE  #Reaction 2
R2__r   0: OPTIMAL      0: OPTIMAL      INACT   #R2:Reverse
R3      0: OPTIMAL      25: OPTIMAL     UND     #Reaction 3
R3__r   0: OPTIMAL      0: OPTIMAL      INACT   #R3:Reverse
R4      0: OPTIMAL      50: OPTIMAL     UND     #Reaction 4
R5      1.01e-006: OPTIMAL      25: OPTIMAL     ACTIVE  #Reaction 5
R5__r   0: OPTIMAL      0: OPTIMAL      INACT   #R5:Reverse
R6      24.75: OPTIMAL  50: OPTIMAL     ACTIVE  #Reaction 6
R6__r   0: OPTIMAL      0: OPTIMAL      INACT   #R6:Reverse
TS_M1   37.125: OPTIMAL 50: OPTIMAL     ACTIVE  #Turnover sink: M1
TS_M2   74.25: OPTIMAL  75: OPTIMAL     ACTIVE  #Turnover sink: M2
TS_M3   1.01e-006: OPTIMAL      25: OPTIMAL     ACTIVE  #Turnover sink: M3
TS_M4   0: OPTIMAL      25: OPTIMAL     UND     #Turnover sink: M4
TS_M5   74.25: OPTIMAL  75: OPTIMAL     ACTIVE  #Turnover sink: M5
TS_M6   37.125: OPTIMAL 50: OPTIMAL     ACTIVE  #Turnover sink: M6
TS_M7   24.75: OPTIMAL  50: OPTIMAL     ACTIVE  #Turnover sink: M7
TS_M8   49.5: OPTIMAL   50: OPTIMAL     ACTIVE  #Turnover sink: M8
;
```

Compared to the output of FVA:

```
#50: OPTIMAL (BIOMASS)
BIOMASS 50: OPTIMAL     50: OPTIMAL     ACTIVE  #Biomass
GROWTH  50: OPTIMAL     50: OPTIMAL     ACTIVE  #Growth
NR1     0: OPTIMAL      50: OPTIMAL     UND     #Nutrient M1
```

```
NR2     100: OPTIMAL    100: OPTIMAL    ACTIVE  #Nutrient M2
NR3     0: OPTIMAL      25: OPTIMAL     UND     #Nutrient M3
NR4     0: OPTIMAL      25: OPTIMAL     UND     #Nutrient M4
R1      50: OPTIMAL     50: OPTIMAL     ACTIVE  #Reaction 1
R2      0: OPTIMAL      25: OPTIMAL     UND     #Reaction 2
R3      0: OPTIMAL      25: OPTIMAL     UND     #Reaction 3
R4      0: OPTIMAL      50: OPTIMAL     UND     #Reaction 4
R5      0: OPTIMAL      25: OPTIMAL     UND     #Reaction 5
R6      25: OPTIMAL     50: OPTIMAL     ACTIVE  #Reaction 6
;
```

Explanation:
The first row is maximized objective flux value, here is the optimal biomass growth rate. The output table has the same format as FVA output, where, the first column is reaction IDs where reaction IDs suffixed with `'__r'` represent reverse reactions split for reversible reactions. Reaction IDs headed with `'TS_'` represent turnover sink reactions for metabolites which have signals in metabolic signal file. The detected metabolites (having signal of 1) will be set positive flux for their turnover sink reactions. The second and third columns are minimum flux and maximum flux for the reactions. The fourth column shows the reaction activities where 'ACTIVE' indicates the reaction was predicted to be active; 'INACT' indicates the reaction was predicted to be inactive; and 'UND' indicates the reaction was predicted to be undetermined. Comparing to the FVA output, we can see that reactions NR2, NR3, R2 and R5 became active in GIM3E output by using metabolic constraints.

3. Usage for predicting gene-nutrient interactions (GNIs)

Synopsis

```
./sfba  -i model_file  -X external_tag  -p problem_type  -b problem_file  [-o biomass] [-s solver] [-c] [-f output_file]
```

Description

`-p problem_type` : analysis for certain problem
The analysis includes:
`sgni` : strong GNI analysis.
`wgni[,sam]` : weak GNI analysis, optional parameter `sam` is sample size of media for wgni. 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 33000.

`-b problem_file` : input problem file
Problem file follows the format of FBA described in MUFINS manual, excepting follows.
`![nutrient names]` - here nutrient names are specified, 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.
`$[nutrient names]` - for the rest nutrients excepting defined nutrients of growth medium, here we can designate the nutrients to be set 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.
`?[gene name]` - here you can designate a list of genes for GNI analysis. The gene names are single space-delimited. If no gene designated, all genes will be investigated.

`-o biomass`: biomass reaction
You have to specify a biomass/growth reaction for GNI analysis.

Example

Run following commands:

```
./sfba -i model2.sfba -X _xt -p sgni -b problem_gni -o BIOMASS -c -f out_gni_sgni
./sfba -i model2.sfba -X _xt -p wgni -b problem_gni -o BIOMASS -c -f out_gni_wgni
```

Output for sgni

```
Gene:  g1
KO Reactions: R1
BLP Result:   Drop = 50:OPTIMAL   Level = 1
Growth Medium:
M1_xt : 1
M2_xt : 1
M3_xt : 0
M4_xt : 0
Nutrients of GNI-SP:       M2_xt

Gene:  g2
KO Reactions: R6
BLP Result:   Drop = 50:OPTIMAL   Level = 1
Growth Medium:
M1_xt : 1
M2_xt : 1
M3_xt : 0
M4_xt : 0
Nutrients of GNI-SP:       M2_xt

Gene:  g3
KO Reactions: R5
BLP Result:   Drop = 50:OPTIMAL   Level = 1
Growth Medium:
M1_xt : 0
M2_xt : 1
M3_xt : 1
M4_xt : 0
Nutrients of GNI-SP:       M2_xt  M3_xt
Nutrients of GNI-SN:       M1_xt

Gene:  g4
KO Reactions: R2     R3
BLP Result:   Drop = 50:OPTIMAL   Level = 1
Growth Medium:
M1_xt : 0
M2_xt : 1
M3_xt : 1
M4_xt : 0
Nutrients of GNI-SP:       M2_xt  M3_xt
Nutrients of GNI-SN:       M1_xt

Gene:  g6
KO Reactions: R4
BLP Result:   Drop = 50:OPTIMAL   Level = 1
Growth Medium:
M1_xt : 1
M2_xt : 1
M3_xt : 0
M4_xt : 0
Nutrients of GNI-SP:       M1_xt  M2_xt
Nutrients of GNI-SN:       M3_xt  M4_xt
;
```

Explanation:

The predicted results for each gene are delimited by a space line, and only the genes having knockout reactions are listed. Explanation of output format is as follows. `Gene:` name of gene; `KO Reactions:` knockout reactions associated with this gene; `BLP Result:` bi-level LP optimization results, where `Drop =` growth rate of wild type – growth rate of knockout strains, followed by solution status, `Level:` the percentage of drop over growth rate of wild type, if Level > 20% then this gene is considered as essential under this growth medium; `Growth Medium:` consisting of a set of nutrients whose present/absent states are indicated by 1/0; `Nutrients of GNI-SP:` the predicted nutrients which have strong positive GNIs with this gene; `Nutrients of GNI-SP:` the predicted nutrients which have strong negative GNIs with this gene. For example gene g4 has spGNIs with M2_xt and M3_xt, and has snGNI with M1_xt.

Output for wgni

```
The analysis results of weak GNI with sample size of 16

Gene:  g1
KO Reactions:R1
Number of Essential medium = 6
Nutrients:
M1_xt  GNI-WP: Number = 4  GNI-WN: Number = 2
M2_xt  GNI-WP: Number = 6  GNI-WN: Number = 0
M3_xt  GNI-WP: Number = 4  GNI-WN: Number = 2
M4_xt  GNI-WP: Number = 3  GNI-WN: Number = 3

Gene:  g2
KO Reactions:R6
Number of Essential medium = 6
Nutrients:
M1_xt  GNI-WP: Number = 4  GNI-WN: Number = 2
M2_xt  GNI-WP: Number = 6  GNI-WN: Number = 0
M3_xt  GNI-WP: Number = 4  GNI-WN: Number = 2
M4_xt  GNI-WP: Number = 3  GNI-WN: Number = 3

Gene:  g4
KO Reactions:R2     R3
Number of Essential medium = 2
Nutrients:
M1_xt  GNI-WP: Number = 0  GNI-WN: Number = 2
M2_xt  GNI-WP: Number = 2  GNI-WN: Number = 0
M3_xt  GNI-WP: Number = 2  GNI-WN: Number = 0
M4_xt  GNI-WP: Number = 1  GNI-WN: Number = 1

Gene:  g3
KO Reactions:R5
Number of Essential medium = 2
Nutrients:
M1_xt  GNI-WP: Number = 0  GNI-WN: Number = 2
M2_xt  GNI-WP: Number = 2  GNI-WN: Number = 0
M3_xt  GNI-WP: Number = 2  GNI-WN: Number = 0
M4_xt  GNI-WP: Number = 1  GNI-WN: Number = 1

Gene:  g6
KO Reactions:R4
Number of Essential medium = 1
Nutrients:
M1_xt  GNI-WP: Number = 1  GNI-WN: Number = 0
M2_xt  GNI-WP: Number = 1  GNI-WN: Number = 0
M3_xt  GNI-WP: Number = 0  GNI-WN: Number = 1
M4_xt  GNI-WP: Number = 0  GNI-WN: Number = 1
;
```

Explanation:

The predicting results for each gene are separated by a space line, and only the genes having knockout reactions are listed. Explanation of output format is as follows. `Gene:` name of gene; `KO Reactions:`

knockout reactions associated with this gene; `Number of Essential medium:` the number of media under which the gene is essential; `Nutrients:` where a set of nutrients and their predicted results are listed; At each line of a nutrient, `GNI-WP:` the predicted results of weak positive GNI where the value of `Number` is the number of presents among all sampled essential growth media. `GNI-WN:` the predicted results of weak negative GNI. The sum of number of wpGNI and wnGNI is the total number of sampled essential media. In cases where a nutrient is found to be present (or absent) in all sampled media, this may also hint to the existence of strong GNIs. As expected, the predicted wGNI results are consistent with the predicted sGNI results. For example, gene g4 has spGNIs with M2_xt and M3_xt which are shown present in all 2 essential media in the wGNI results. Also g4 has snGNI with M1_xt which is shown absent in all 2 essential media in the wGNI results.

Statistical scripts

To address the statistical significance of wgni results, we offer three scripts for this purpose.

1) Perl script " wgni2table.pl" - extract wgni results into a table format which can be read for a R script.

Usage:

```
perl wgni2table.pl wgni_file [ess_gene_file] count_table_file
```

`wgni_file`: wgni file from predicted wgni results.

`ess_gene_file`: a optional file contains a list of essential genes, one gene each line, only these genes will be counted for table, otherwise without this file all genes will be considered.

`count_table_file`: output file of a count table.

Example:

For above wgni results, run perl command:

```
perl wgni2table.pl out_gni_wgni table_wgni
```

get output:

```
     M1_xt.WP  M1_xt.WN  M2_xt.WP  M2_xt.WN  M3_xt.WP  M3_xt.WN  M4_xt.WP  M4_xt.WN
 g1  4         2         6         0         4         2         3         3
 g2  4         2         6         0         4         2         3         3
 g4  0         2         2         0         2         0         1         1
 g3  0         2         2         0         2         0         1         1
 g6  1         0         1         0         0         1         0         1
```

Note:

Rows correspond with genes and columns correspond with nutrients. The numbers are counts associated with the related genes and nutrients. Column name is formatted as "nutrient name.WP(WN)" where WP(WN) means weak positive (weak negative).

2) R script "wgniStat" - an R function performing statistical computing for input of above table.

Usage:

```
wgniStat("count_table_file","stat_table_file")
```

`count_table_file`: count table file from output of wgni2table.pl script.

`stat_table_file`: output file of the statistical significance for each count in count table.

Example:

For above count table results, run R command:

```
wgniStat("table_wgni","stat_wgni")
```

get output:

```
 The p-value for weak GNIs

     M1_xt.WP     M1_xt.WN     M2_xt.WP  M2_xt.WN  M3_xt.WP  M3_xt.WN  M4_xt.WP  M4_xt.WN
 g1  0.1066       0.6581       0.0000    0.9853    0.1066    0.6581    0.3419    0.3419
 g2  0.1066       0.6581       0.0000    0.9853    0.1066    0.6581    0.3419    0.3419
 g4  0.7510       0.0000       0.0000    0.7510    0.0000    0.7510    0.2490    0.2490
 g3  0.7510        0.0000        0.0000     0.7510     0.0000     0.7510     0.2490     0.2490
 g6  0.0000        0.5000        0.0000     0.5000     0.5000     0.0000     0.5000     0.0000
```

Explanation:
The number is estimated p-value indicating the statistical significance of weak GNIs, the smaller this p-value the more significant the GNI is, normally it is considered as weak GNI if p-value < 0.05. It should be noted that given a media space the larger sampled essential media space the more statistically reliable the p-value would be, and also the larger the media space the more sampled essential media are needed to satisfy reliable statistics. This may be problematic for genes that are essential for growth in a small fraction of the media space.

3) R script "predictMedia" - an R function used for predicting growth medium composition.

Usage:
`predictMedia("count_table_file", "media_file", threshold)`
`count_table_file` : count table file from output of wgni2table.pl script.
`media_file` : output file contains predicted statistical results of growth medium composition.
`threshold` : a p-value threshold used for counting the number of weak GNIs for each nutrient, the smaller this threshold the more conservative the predicted results could be.

Example:
For above count table results, run R command:
`predictMedia("table_wgni", "media", 1e-7)`
get output:

```
         Presence  Absence  Sum      Pval_Pre  Pval_Abs
 M1_xt   1         2        3        0.9167    0.5
 M2_xt   5         0        5        0.0040    1
 M3_xt   2         1        3        0.5       0.9167
 M4_xt   0         1        1        1         0.5
```

Explanation:
`Presence` : number of weak positive GNIs for each nutrient; `Absence` : number of weak negative GNIs for each nutrient; `Sum = Presence + Absence`; `Pval_Pre` : p-value indicating the statistical significance of presence; `Pval_Abs` : p-value indicating the statistical significance of absence. The smaller the p-value the more significant the presence/absence would be. We can set a cut-off of p-value (such as 0.05) used for judging if the nutrient present or absent.

4. Usage for differential producibility analysis (DPA)

Synopsis

```
./sfba  -i model_file  -X external_tag  -j gexp_file  -u dpaplot_file  -p problem_type  -b
problem_file  [-c] [-f output_file]
```

Description

`-j gexp_file` : Input gene expression data file (only for dpasig)
Gene expression data should be in the form of log2 ratios of treatment and reference sample signals, no processing of data was conducted; Here is an example:

```
Gene     Array1      Array2
g1       2.5         -1.8
```

```
g2       1.3        - 1.5
g3      -1.6         1.6
…
```

Where, first column consists of gene names. From second column, each column is associated with a microarray name and their gene expression ratios. Columns are tab-delimited, with 'NA' denoting empty value. This file is only used for problem dpasig.

`-u dpaplot_file` : the output file of problem dpaplot (only for dpasig)
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.

`-p problem_type` : analysis for certain problem
The analysis includes:
`dpaplot` : DPA analysis of producibility plot using GLPK library
`dpaplotGrb`: DPA analysis of producibility plot using Gurobi library
`dpasig` : DPA analysis of metabolite signals

`-b problem_file` : input problem file
Problem file follows the format of FBA described in MUFINS manual, excepting follows.
`![array name]` - for problem dpaplot, just leave it empty; for problem dpasig, we can designate microarray names of interest, if not set all microarrays in gexp_file will be used. The array names are single space-delimited.

Example

Run following commands:

```
./sfba -i model2.sfba -X _xt -p dpaplot -c -f out_dpaplot
./sfba -j gexp_ratio -u out_dpaplot -p dpasig -b problem_dpasig -c -f out_dpapsig
```

Output for dpaplot

```
 Metabolites to genes:

 M3                g4
 M4                g4         g6       g3          g2
 M5                g1         g3
 M6                g6         g3       g2
 M7                g1         g4       g6          g3       g2
 M8                g6         g2
 M_BIOMASS         g1         g2
 ///

 Genes to metabolites:

 Gene:             g1
 KO Reactions:     R1
 Metabolites:      M5         M7       M_BIOMASS

 Gene:             g2
 KO Reactions:     R6
 Metabolites:      M4         M6       M7          M8       M_BIOMASS

 Gene:             g3
 KO Reactions:     R5
 Metabolites:      M4         M5       M6          M7
```

```
Gene:             g4
KO Reactions:     R2        R3
Metabolites:      M3        M4       M7

Gene:             g6
KO Reactions:     R4
Metabolites:      M4        M6       M7         M8
///

Matrix of producibility of wildtype and knockouts:

                  WildType  g1       g4         g6       g3
M1                100       100      100        100      100
M2                100       100      100        100      100
M3                200       200      100        200      200
M4                200       200      100        100      150
M5                25        0        25         25       0
M6                100       100      100        50       50
M7                75        50       50         25       50
M8                25        25       25         12.5     25
M_BIOMASS         50        0        50         50       50
;
```

Explanation:
The output of dpaplot is represented as four parts of information about producibility plot. Here, the model used for this analysis is the same as the model in GNI analysis. The first part is for `Metabolites to genes`, which showing the mapping from metabolites to its essential gene sets. For each line, the first column is the name of a metabolite which having essential genes. The other columns are the set of gene names associated with this metabolite, where the names are tab-delimited. By this simple format, the mapping information can be easily read by dpasig problem, where the output of dpaplot will be used as input file for problem dpasig. The second part of `Genes to metabolites` shows the mapping from gene to its affected metabolites. The predicted results for each gene are delimited by a space line, and only the genes having knockout reactions are listed. Explanation of output format is as follows. `Gene:` name of gene; `KO Reactions`: knockout reactions associated with this gene; `Metabolites:` a set of metabolites associated with this gene. The fourth part present an `matrix of producibility` plot. The matrix is represented as metabolites in rows by genes in columns, where first column corresponds names of metabolites and first row corresponds `wildtype` and names of genes. The `Matrix of producibility` shows the maximal fluxes towards a metabolite in wild-type mode and the maximal flux in gene-knockout models.

Output for dpasig

```
Metabolite signals from microarray data: Array1 Array2

Signals for up-regulated genes:

                        Array1              Array2
M3                      1.8                 2.2
M4                      1.55                2
M5                      2.5                 1.6
M6                      1.3                 1.8
M7                      1.8                 2
M8                      1.3                 2
M_BIOMASS               1.9                 0
///

Signals for down-regulated genes:

                        Array1              Array2
```

```
M3                  0                   0
M4                  -1.8                -1.5
M5                  -1.6                -1.8
M6                  -1.8                -1.5
M7                  -1.8                -1.65
M8                  -2                  -1.5
M_BIOMASS           0                   -1.65
;
```

Explanation:
Above results give the metabolite signals which are calculated as median of expression ratios of essential genes for each metabolite. In terms of whether expression ratios being positive or negative, two signals for each metabolite can be obtained which separately shown as two matrices in output, they are `Signals for up-regulated genes` and `Signals for down-regulated genes`. The matrix is represented as metabolites in rows by microarray in columns, where first column corresponds names of metabolites and first row corresponds names of microarrays, here zero means no gene can be found for this metabolite signal.

5. Add signalling constrains on model

Cellular signalling is important part in controlling the metabolic network. In MUFINS, you are able to add signalling constrains to control reaction fluxes. Signalling constrains were implemented in pure linear formulation based on paper "A Linearized Constraint-Based Approach for Modeling Signaling Networks".

Steps of adding signalling constrains are explained as follows:
1) Define the signalling species (activators/inhibitors). You can let an existing metabolite becoming activator/inhibitor by adding a transfer reaction. You can also use external activator/inhibitor by adding a transport reaction. You can also add an inhibitor directly using an existing metabolite.
In example model, transfer reaction S1 defines an activator aM1 which is provided by M1 with upper bound flux of 45; using M5 as inhibitor for R4 so that flux of R4 is bounded by production flux of M5. Note that here the production of metabolite is only for non-reversible reactions which can produce this metabolite.
2) Add signalling species into target reactions. You can add activators/inhibitors on the right-hand of target reaction equation as following formats:

```
+ & Activator
+ ~ Intensity Inhibitor
```

where symbol '&' and '~' indicates activator and inhibitor respectively; For inhibitor you can set inhibition intensity, so the sum of the flux of target reaction and the multiplication of intensity and inhibitor production flux should be less than the maximum flux of target reaction. One reaction can have multiple activators/inhibitors and one activator/inhibitor can affect multiple reactions. Note that here inhibited target reactions are assumed in forward direction.
In example model, reaction R4 is regulated by activator aM1 and inhibitor iM3 with intensity of 10.

Example model

```
R4         M6 + & aM1 + ~ 3 M5 = M7      0       100     g6     #Reaction 4
S1         M1 = aM1                      0       20             #M1 -> activator aM1
R1         M1 + 2 M2 = M5 + M6           0       100     g1     #Reaction 1
R2         M3 = 2 M1 + M7                -100    100     g4     #Reaction 2
R3         M4 = M1 + M7                  -100    100     g4     #Reaction 3
R5         2 M6 = M5 + M8                -100    100     g3     #Reaction 5
R6         M7 = M5 + M8                  -100    100     g2     #Reaction 6
BIOMASS    2 M5 + M8 = M_BIOMASS         0       1000           #Biomass
GROWTH     M_BIOMASS = M_BIOMASS_xt      0       1000           #Growth
```

```
 NR1       M1_xt = M1                     0       100           #Nutrient M1
 NR2       M2_xt = M2                     0       100           #Nutrient M2
 NR3       M3_xt = M3                     0       100           #Nutrient M3
 NR4       M4_xt = M4                     0       100           #Nutrient M4
```

Run FBA by maximizing the R4 flux with command:

```
./sfba -i model_sig.sfba -X _xt -p fba -o R4 -c
```

We will get following results:

```
#20: OPTIMAL (R4)
BIOMASS      20      #Biomass
GROWTH       20      #Growth
NR1          40      #Nutrient M1
NR2          40      #Nutrient M2
NR3          0       #Nutrient M3
NR4          0       #Nutrient M4
R1           20      #Reaction 1
R2           -0      #Reaction 2
R3           -0      #Reaction 3
R4           20      #Reaction 4
R5           0       #Reaction 5
R6           20      #Reaction 6
S1           20      #M1 -> activator aM1
aM1__col     20      #
```

Explanation:

Without signalling constraints, reaction R4 can get maximum flux of 50. However, R4 can be inhibited by M5 by the amount of flux of inhibitor's production flux multiplied by its intensity that is 20 × 3 = 60, so R4 can only be activated by amount of flux of 100 – 60 = 40. Reaction R4 can only be activated by activator aM1 by the amount of flux of 20 as upper bound flux of aM1 = 20, so overall R4 can get maximum flux of 20.