Red text denotes information that applies only to the “reaction-based penalty” (RBP) iMRM.

Green text denotes information that applies only to the “gene-based penalty” (GBP) iMRM.

Black text denotes information that applies to either version of the iMRM.

Throughout this document we consider the method as run for a single, given, experimental condition. We also ignore some computational optimizations (for instance, omitting columns for nonreversible reactions), presenting the method purely mathematically.

**Inputs to the iMRM process**

* SBML file for the metabolic model
* Aijs for each gene (in the given experimental condition)
  + Technically, we only need aijs for each gene in the metabolic model, as other aijs will not be used in this process. We will refer to “genes in the metabolic model” as “metabolic genes” in the remainder of this document.
* Description of the media components present in the given experimental condition, and their concentrations
* Description of gene knockouts present in the given experimental condition

**Procedure**

1. Set up the MILP (see section “Structure of the MILP”).
2. Set and ( being irrelevant in this case). In particular, note that with these choices for and , the objective function reduces to (see section “Objective Function” under “Structure of the MILP”) – that is, the objective will be simply to maximize the forward flux through the biomass reaction given these media and gene knockout conditions, disregarding the aijs.
3. Solve the MILP. Obtain ComputedMaxBiomass = the negative of the minimum value of the objective function, that is, the maximum value of . ComputedMaxBiomass will be nonnegative, since is nonnegative (see section “Columns (Variables)” under “Structure of the MILP”).
4. Assuming ComputedMaxBiomass (else the objective function is undefined), re-solve the MILP under the same conditions but with the desired and parameters, and with ComputedMaxBiomass. Obtain whatever results are desired, for instance the values of the any of the variables in the optimal solution.

**Structure of the MILP**

Columns (Variables):

* for each reaction , representing the forward flux through that reaction;
* for each reaction , representing the reverse flux through that reaction; for reversible reactions, and for nonreversible reactions.
  + For exchange reactions (which are always reversible), bounds on may be modified to account for media conditions; see section “Media Conditions”
* for each reaction ; ; for properties, see “Rows (Constraints)”
  + Gene knockouts may produce additional restrictions on ; see section “Gene Expression and Knockouts”
* for each reaction ; ; for properties, see “Rows (Constraints)”
* for each reaction ; ; for properties, see “Rows (Constraints)”
* for each metabolic gene ; (analogous to but for genes)
  + For each metabolic gene that was indicated as knocked-out in the given experimental condition, we constrain . This will prevent gene from being activated by the optimizer, and cannot be overridden in any way.
* for each metabolic gene ; (analogous to but for genes)
* for each reaction ; this is defined in section “Gene Expression and Knockouts”
* All of the variables in the (possibly empty) set for each reaction ; this is defined in section “Gene Expression and Knockouts”

Rows (Constraints):

* for each compound , where is the stoichiometric coefficient of compound in reaction ; is positive if the compound is produced by the forward direction of the reaction, or negative if the compound is consumed by the forward direction of the reaction. This constraint specifies that the net amount of each compound produced/consumed must be equal to 0, accounting for the possibility of reverse flux through any reaction.
* for each reaction . This constraint specifies that if a given reaction has any flux at all, must be 1.
  + The inverse is not true: if there is no flux through reaction , this constraint does not force to any particular value. Either 0 or 1 for satisfies the constraint in that case.
* for each reaction . This constraint specifies that if a given reaction has no flux, must be 1.
  + The inverse is not true: if there is flux through reaction , this constraint does not force to any particular value. Either 0 or 1 for satisfies the constraint in that case.
* and for each reaction .
  + For a given reaction , if is nonzero, the first constraint forces to be 1. Likewise if is nonzero, the second constraint forces to be 0. Therefore, these constraints together specify that a reaction cannot have both forward and reverse flux simultaneously, because cannot be both 0 and 1 simultaneously: either the forward or the reverse flux (or both) must be 0.
  + If both fluxes are zero for a reaction , is not forced to any particular value. Either 0 or 1 for satisfies the constraints.
* for each reaction and for each metabolic gene . The first constraint specifies that for each reaction , either and , or vice versa – that is, the reaction is either on or off respectively. The second constraint does analogously for genes.
  + Originally the first constraint was not included in the RBP iMRM, as it was not necessary, but including it makes the behavior of the optimization more intuitive in some sense (each reaction must either be on or off, as indicated by and ) and makes and more interpretable.[[1]](#footnote-1) Regardless, it is structurally necessary in the GBP iMRM.
* All of the constraints in the (possibly empty) set for each reaction ; this is defined in section “Gene Expression and Knockouts”
* for each reaction ; this is defined in section “Gene Expression and Knockouts”
* for each metabolic gene ; this is defined in section “Gene Expression and Knockouts”

Objective Function:

Minimize:

where:

where varies over all reactions,

where varies over all metabolic genes,

where is the number of reactions metabolic genes,

denotes the biomass reaction (and thus the forward flux through the biomass reaction),

is the maximum value of given these media and gene-knockout conditions,

is a weighting factor with, which controls the weighting of matching the gene expression data (lower ω) vs. maximizing biomass (higher ω),

is a weighting factor with , which controls the weighting of the penalty for deactivating high-expression reactions genes (higher ) vs. the penalty for activating low-expression reactions genes (lower ),

and and and are as defined in section “Gene Expression and Knockouts”.

**Media Conditions**

Some compounds can be exchanged between the organism and the environment, as the organism either takes in the compound from the environment or secretes it into the environment. This is modeled by “exchange reactions” for these compounds in the SBML file. Each compound capable of being exchanged with the environment has an exchange reaction, which has that compound as a reactant but has no product.[[2]](#footnote-2) This means that when an exchange reaction has forward flux, some amount of the compound is consumed by the exchange reaction, modeling secretion of the compound by the organism into the environment. Likewise, when an exchange reaction has reverse flux, some amount of the compound is produced by the exchange reaction, modeling intake of the compound by the organism from the environment. These exchange reactions, implemented this way, ensure that the stoichiometric constraints for all compounds can still be satisfied (see section “Rows (Constraints)” under “Structure of the MILP”) – that is, that the net amount of the compound produced/consumed can still appear to be 0 thanks to the exchange reaction. Media conditions, which specify what compounds are available in the environment and in what concentrations, therefore yield restrictions on exchange reactions, but not on their forward flux – only on their reverse flux.

For each exchange reaction and associated compound , we enforce , where is the concentration at which compound is available in the given experiment, or if the compound is not available. Some experiments are considered “complete”, in which case the bounds on are left at for all exchange reactions . This means the experiment has access to all compounds for which an exchange reaction is included in the SBML model.

**Gene Expression and Knockouts**

Each reaction in the SBML file is associated with a “gene-string” consisting of zero or more “gene-forms” connected by the binary operators “and” and “or”, where a “gene-form” is either a gene name or the word “Unknown”, and where parentheses are used sufficiently in order to completely define the correct order of operations.

Let be the aij for gene in the given experimental condition[[3]](#footnote-3), or, if gene was indicated as knocked-out in the given experimental condition, then , a sentinel value indicating that cannot ever be activated, as it was knocked out.

Let us define a function which maps a gene-string to an element of the set , where:

if is a single valid gene name referring to gene

if is of the form “” for valid gene-strings and

if is of the form “” for valid gene-strings and

if is the empty string or consists only of whitespace

if = “”

Intuitively, indicates that the gene-string cannot ever be activated given the gene knockouts for the given experimental condition. In order to match this intuition, we define (the ‘and’ expression cannot ever be activated because one of its constituent components cannot ever be activated), and (the NA portion can’t ever be activated, so the expression in total reduces to just the non-NA part).

If for any reaction , that reaction is considered to be knocked-out because there is no way for the reaction to be activated given the gene knockouts for the given experimental condition. For each such reaction , we constrain . This will prevent any flux at all, whether forward or reverse, from occurring through the reaction, and cannot be overridden in any way. For all other reactions we allow as specified above.

Let us define a function which maps a gene-string to an ordered triple , where

is a new MILP binary variable ()

and are (possibly empty) sets of additional new MILP constraints and variables, respectively.

All of the MILP variables and constraints in , , and jointly enforce the property that, given the values of MILP variables and for all , will indicate whether ’s Boolean expression is true () or false (). and are defined as follows:

If is a single valid gene name referring to gene :

constraint ; . This constrains that .

Else if is of the form “” for valid gene-strings and :

Let and likewise .

Then , where

constraint (which specifies that if is true, and must both be true; but enforces nothing if is false)

constraint (which specifies that if is false, at least one of and must be false; but enforces nothing if is true)

and .

Else if is of the form “” for valid gene-strings and :

Let and likewise .

Then , where

constraint (which specifies that if is true, at least one of and must be true; but enforces nothing if is false)

constraint (which specifies that if is false, and must both be false; but enforces nothing if is true)

and .

Else if is the empty string, consists only of whitespace, or is the string “”:

. This leaves the optimizer completely free to select or for this .

This concludes the definition of .

For each reaction , recall that is its associated gene-string. Then define , , and (used in section “Structure of the MILP”) by .

For each reaction , define (used in section “Structure of the MILP”) as constraint ; enforces the association between reaction and its gene-string . This specifies that if reaction is active (), the Boolean expression associated with its gene-string must be true () [[4]](#footnote-4). However, the converse is not true: gene-string being true () does not constrain reaction whatsoever.[[5]](#footnote-5) (Although does not enforce the converse, (defined below) partially enforces it.)

For each metabolic gene , let be the set of reactions in whose gene-strings appears. Then define (used in section “Structure of the MILP”) as constraint ; specifies that if is active then at least one of the reactions in must be active as well. Note that since is a metabolic gene, by definition there must be at least one reaction in ; so is always satisfiable. Also note that if is inactive, has no effect whatsoever.

In the GBP iMRM, gene knockouts are handled as mentioned in section “Columns (Variables)” under “Structure of the MILP”, with a constraint on the gene’s . No other adjustments are necessary.

Finally, for each reaction metabolic gene , define:

if

if

otherwise

and

if

if

otherwise

represents the penalty to be applied to the objective function if the reaction carries nonzero flux gene is activated. If , the gene expression and knockout data indicate the reaction should carry nonzero flux gene should be activated, so there is no penalty; however, if , then there will be a penalty in proportion to the distance between 0.5 and – the stronger the gene-expression evidence, the higher the penalty for overriding it[[6]](#footnote-6). Likewise, represents the penalty to be applied to the objective function if the reaction carries no flux gene is inactive. If , there is no penalty, as the gene expression and knockout data agree that the reaction should not carry flux gene should be inactive; but if , there will be a penalty in proportion to the distance between 0.5 and . If , the reaction will never carry flux gene will never be activated due to the restriction on , so no penalty is needed.

1. Technically, this makes and redundant variables – we could eliminate and substitute everywhere occurs in this document. Likewise for and . [↑](#footnote-ref-1)
2. Technically, in the SBML, there is a product, which is a compound with the suffix “\_b”. Our processing of the SBML throughout this process ignores compounds with the suffix “\_b” for all purposes – they effectively appear in no reactions and produce no stoichiometric constraints. This ensures that the exchange reactions work correctly, as described here. [↑](#footnote-ref-2)
3. I realize this is a terrible abuse of notation in that I’m completely ignoring the fact that “aij” originally already had two subscripts, for the gene and for the experiment. In this document, generally represents a reaction, so also using it for a gene would be confusing. However, I desired to keep “aij” as a word rather than renaming subscripts, because by now in this group we all think of “aij” as a well-defined term and immediately know what is meant. [↑](#footnote-ref-3)
4. The way this is stated is more intuitive to see how the constraint itself works. However, from a biology standpoint, it is more intuitive to view this relationship in the contrapositive: if reaction ’s gene-string is false (), reaction cannot be active (we must have ). [↑](#footnote-ref-4)
5. Since this is the converse of the original statement (in the main text), it is also the inverse of the contrapositive (in footnote 6). We could also use, equivalently, the contrapositive of this statement, which is the inverse of the original statement (in the main text) and the converse of the contrapositive (in footnote 6): reaction being inactive does not constrain or any associated genes whatsoever. [↑](#footnote-ref-5)
6. As an aside, I think it could be useful to experiment with non-linear functions here instead of a linear penalty based on the distance from 0.5. For instance, it might be more realistic to have a function which imposes twice the penalty when as it does when , since in some sense, in the former case we are “twice as sure” that the reaction gene is actually inactive. The function has this property, although the penalty doesn’t nicely go to 0 as approaches 0.5. Less extreme and probably better would be the function , which does nicely go to 0 as approaches 0.5. [↑](#footnote-ref-6)