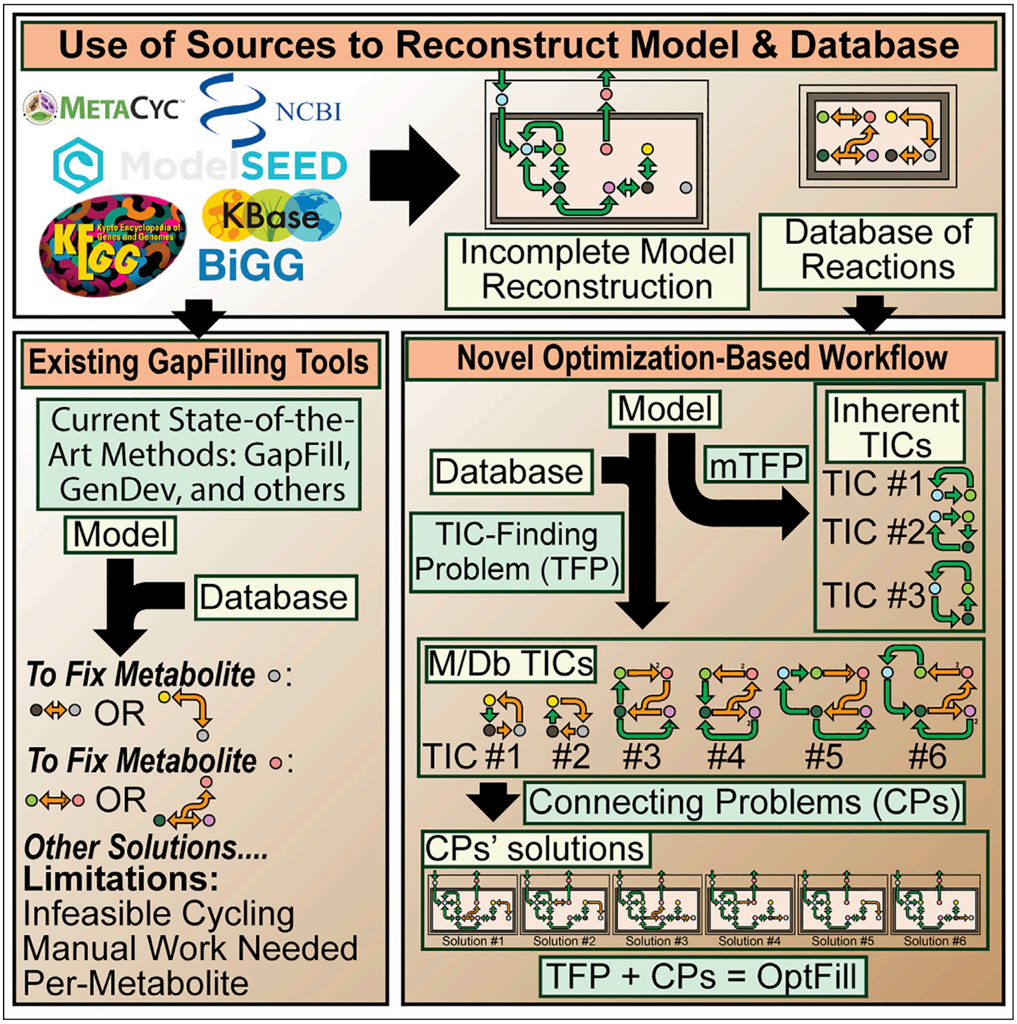
iScience

CellPress

Article

OptFill: A Tool for Infeasible Cycle-Free Gapfilling of Stoichiometric Metabolic Models



Wheaton L. Schroeder, Rajib Saha

rsaha2@unl.edu

HIGHLIGHTS

This work presents an alternative to state-of-the-art methods for gapfilling

Unlike current methods, this method is holistic and infeasible cycle free

This method is applied to three tests and one published model

This method might also be used to address infeasible cycling

Schroeder & Saha, iScience *23*, 100783
January 24, 2020 © 2019 The Author(s).
https://doi.org/10.1016/j.isci.2019.100783

iScience

CellPress

Article

OptFill: A Tool for Infeasible Cycle-Free Gapfilling of Stoichiometric Metabolic Models

Wheaton L. Schroeder[1] and Rajib Saha[1,2,*]

SUMMARY

Stoichiometric metabolic modeling, particularly genome-scale models (GSMs), is now an indispensable tool for systems biology. The model reconstruction process typically involves collecting information from public databases; however, incomplete systems knowledge leaves gaps in any reconstruction. Current tools for addressing gaps use databases of biochemical functionalities to address gaps on a per-metabolite basis and can provide multiple solutions but cannot avoid thermodynamically infeasible cycles (TICs), invariably requiring lengthy manual curation. To address these limitations, this work introduces an optimization-based multi-step method named OptFill, which performs TIC-avoiding whole-model gapfilling. We applied OptFill to three fictional prokaryotic models of increasing sizes and to a published GSM of *Escherichia coli*, iJR904. This application resulted in holistic and infeasible cycle-free gapfilling solutions. In addition, OptFill can be adapted to automate inherent TICs identification in any GSM. Overall, OptFill can address critical issues in automated development of high-quality GSMs.

INTRODUCTION

The use of systems biology in uni- and multi-cellular organisms (e.g. plants and animals) to engineer or enhance desirable phenotypes and study system-wide metabolic processes is well-established and capable of affecting the lives of millions of individuals, such as in the case of artemisinin production in yeast or enhancing the nutritional value of agricultural products (Beyer et al., 2002; Hall et al., 2008). As opposed to traditional qualitative approaches, computational approaches based on stoichiometric genome-scale models (GSMs) of metabolism can be used to predict non-intuitive genetic interventions (Srinivasan et al., 2015) by accounting for gene-protein-reaction (GPR) links. GSMs may also lead to increased understanding of how a change in environment, organism nutrition, or a gene knockout can affect the entire metabolic system of an organism through tools such as flux balance analysis (FBA) (Orth et al., 2010), OptKnock (Burgard et al., 2003), and OptForce (Ranganathan et al., 2010). GSMs have been developed for many prokaryotic (Magnúsdóttir et al., 2016; Shoaie et al., 2013), animal (Brunk et al., 2018), plant (Gomes de Oliveira Dal'Molin et al., 2015; Saha et al., 2011), and fungal (Andersen et al., 2008; Liu et al., 2013) systems, enhancing mechanistic understanding and exploration of system-wide metabolism in such organisms as *E. coli* (Ranganathan et al., 2010), cyanobacteria (Saha et al., 2016), yeast (Ng et al., 2012), and other species (Gudmundsson et al., 2017; Islam et al., 2018; Saha et al., 2011; Shoaie et al., 2013). GSMs are typically reconstructed by gleaning information on gene annotations, enzyme functions, associated reactions, and reaction directionality from major public databases such as KEGG (Kanehisa et al., 2017), ModelSEED (Overbeek et al., 2005), the NCBI (Limviphuvadh et al., 2018), MetaCyc (Caspi, 2006), K-Base (Arkin et al., 2018), and BIGG (King et al., 2016). At present, there is no complete knowledge of any genome. For instance, the annotated genome of one of the most prolifically studied organisms, *Escherichia coli* strain K-12 substrain MG1655, contains about 6.8% putative proteins and 16.1% uncharacterized proteins (UniProtKB, 2018). Furthermore, approximately 61% of proteins lack an enzyme commission (EC) number, which is important for the identification of GPR links in any GSM reconstruction (UniProtKB, 2018). Inevitably, incomplete gene annotation and system knowledge (including reaction direction) leaves metabolic gaps, imbalances, or thermodynamically infeasible cycles (TICs) in any initial GSM reconstructions, leaving the model incomplete. Particularly problematic are TICs, sets of reactions that can carry flux in the absence of nutrition provided to the model because their net stoichiometry is zero, also known as futile cycles or type III reactions (Thiele and Palsson, 2010). These cycles can negate metabolic costs (Thiele and Palsson, 2010), report infeasibly large reaction rates, be difficult to identify (De Martino et al., 2013; Schellenberger et al., 2011), and inhibit the proper function of optimization-based

[1]Department of Chemical and Biomolecular Engineering, University of Nebraska-Lincoln, Lincoln, NE 68588, USA

[2]Lead Contact

*Correspondence: rsaha2@unl.edu

https://doi.org/10.1016/j.isci.2019.100783

tools that rely on duality to optimize multiple objectives such as OptKnock (Burgard et al., 2003) and OptForce (Ranganathan et al., 2010).

A significant challenge to reconstruct GSMs is the amount of time and manual labor required to curate these incomplete reconstructed models, addressing various issues such as element and charge balances, reaction directionality, metabolic gaps, TICs, and other inconsistencies. Hence, it often requires months to years of manpower before a predictive model is generated (Thiele and Palsson, 2010), which is a prerequisite for conducting research on phenotypic enhancement or study metabolism. Two of the most challenging aspects of model development are the identification and elimination of TICs, as well as the resolving of metabolic gaps.

The existing methods/tools that have been developed to address the identification and resolution of TICs can be broadly categorized into four groups: (1) methods that can identify existing TICs in a model (De Martino et al., 2013), (2) methods that can force no-flux through existing TICs in a model (Schellenberger et al., 2011; Nigam and Liang, 2007; Chan et al., 2018), (3) a combination of the previous two (Chan et al., 2018), and (4) methods eliminating TICs by manipulating the metabolic network. Although developing these is a significant step toward building a better and more predictive GSM, there remain challenges that need to be addressed. For the first approach, Monte Carlo sampling-based method (De Martino et al., 2013) cannot guarantee the identification of all TICs as it is a stochastic approach. The second approach is the avoidance of TICs by the application of Kirchhoff's loop law in methods such as Loopless COBRA (Schellenberger et al., 2011). This approach does successfully avoid TICs but does not address the root cause in the model that can make some models problematic for tools such as OptForce that require no TICs (Ranganathan et al., 2010). Another approach is the addition of thermodynamic constraints to the model using known thermodynamic quantities (Nigam and Liang, 2007), which works well for well-studied organisms for which these *in vivo* parameters are known but is more difficult to implement for non-model organisms. The third approach that combines these two approaches, such as the one demonstrated by Chan et al. (2018), has shown promise and computational tractability. However, this has generally been employed as a set of loopless constraints, rather than as a method to avoid the inclusion of TICs in gapfilling. The fourth method has been used to address TICs in energy metabolism, which can allow the model to produce unlimited energy severely hampering model accuracy, by applying a variation of optimization-based tool GLOBALFIT (Fritzemeier et al., 2017). GLOBALFIT has been used by Fritzemeier et al. (2017) to identify the minimal network changes to address erroneous energy cycling in metabolic network models. These changes could take the form of removal of reactions and/or correcting of reaction direction and address root causes of TICs without using loopless constraints when applying *in silico* analysis tools.

It should be noted that not all the cycles in biological systems are infeasible cycles. Some cycles, such as the Calvin cycle or the citric acid cycle are well-known biological cycles. These differ from infeasible cycles in that these cycles has some net effect. In the case of the Calvin cycle this net effect of each revolution is to fix carbon dioxide to a sugar by expending cellular energy. In contrast, thermodynamically infeasible cycles result in no net production or consumption per each revolution. It should also be noted that some reactions do proceed in both directions at the same time in the same subcellular compartment in a cell, with their relative rates limited by thermodynamic considerations. Although some models do include *in vivo* thermodynamic information, the precise value, or more often range of values, for the Gibbs free energy and other important thermodynamic properties of a reaction are often unknown aside from being able to specify reaction direction (Thiele and Palsson, 2010). Therefore, for all but the best-studied organisms, imposing thermodynamics-based limitations on reaction rates to preclude thermodynamic cycling is very difficult if not impossible.

To address and resolve metabolic network reconstruction gaps, GapFind and GapFill (Satish Kumar et al., 2007) are some of the most common tools used (Pitkänen et al., 2014; Henry et al., 2010; Kim et al., 2012). GapFind and GapFill are optimization-based Mixed Integer Linear Programming (MILP) problems and have been successfully implemented in the reconstruction of metabolic models, prokaryotic, and eukaryotic biological systems such as cyanobacteria (*Synechocystis* sp. PCC 6803 and *Cyanothece sp ATCC 51142*) (Saha et al., 2012), corn (*Zea mays*) (Simons et al., 2014), yeast (*Saccharomyces cerevisiae*), and Chinese hamster ovary cells (Chowdhury et al., 2015). Other methods of automated gapfilling that build on the capabilities of GapFill include GenDev (Latendresse and Karp, 2018), FastDev (Latendresse and Karp, 2018), likelihood-based

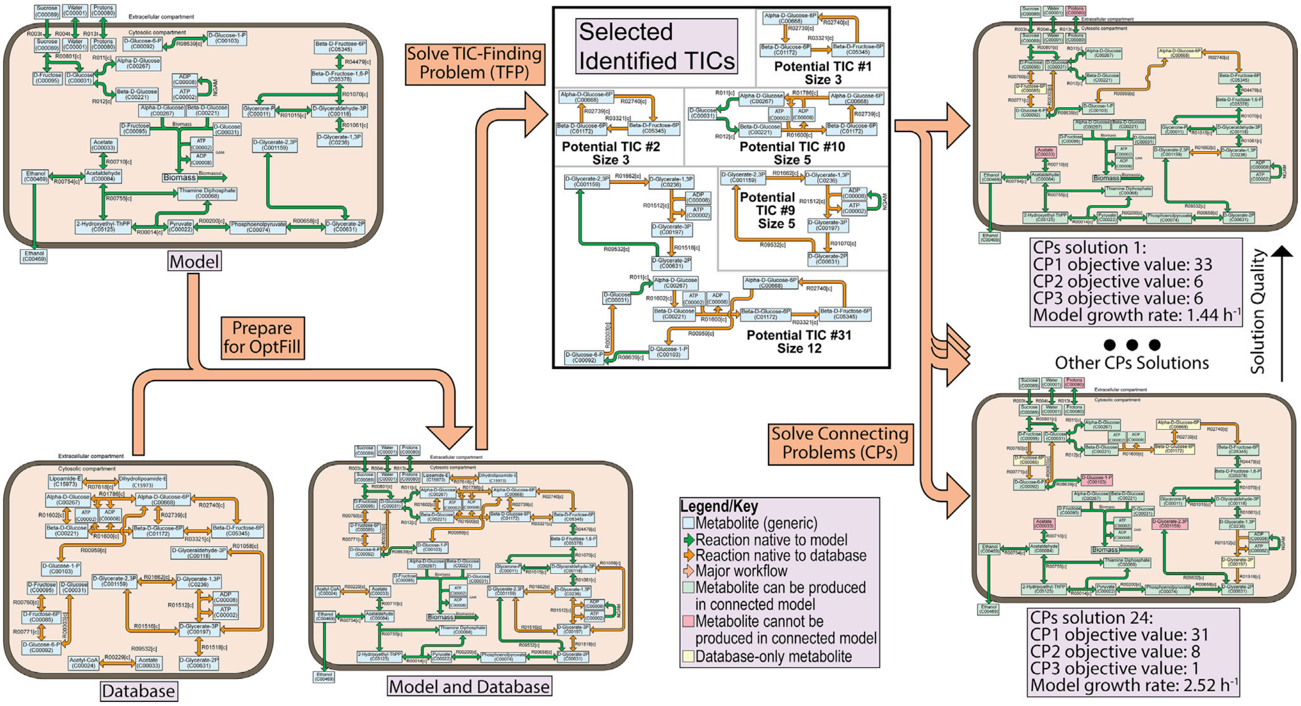


Figure 1. Visualization of OptFill Results with Respect to the First Test Model and Database

This figure shows that the model and database which are inputs to OptFill are separate but are both used in the workflow to prepare for OptFill and in OptFill itself.

Then, the model and database are combined to show how they might appear and how this combination is used in the TIC-finding problem to identify potential TICs that might occur between the model and the database. Selected identified potential TICs are shown here as illustrative examples. Potential TICs #1 and #2 illustrate how TICs occurring in different directions are identified as separate TICs, how identified TICs might only occur between database reactions, and the two of the smallest identified TICs. Potential TICs #9 illustrates a larger TIC that makes use of an irreversible reaction (NGAM), and therefore has no opposite-direction TIC, making the direction of the other reactions important. Potential TICs #10 and #31 illustrate infeasible cycling involving an energy molecule (ADP/ATP), in addition to potential TIC #31 being the largest identified TIC.

Finally, this figure shows the application of the connecting problems (CPs), which make use of the database, model, and TIC-Finding Problem solutions. Shown here are the first (most optimal) and last (least optimal) solutions of the CPs. These solutions differ in the number of model metabolites that could not be connected (red boxes); the number of metabolites introduced to the model (yellow boxes); the number and reversibility of database reactions added (orange arrows); and the resultant model growth rate.

gapfilling (Karp et al., 2018), and phenotype-based gapfilling (Cuevas et al., 2019). All these tools are constructed with the aim of increasing the accuracy of the GapFilling method, through comparison to some level of data such as phylogenetic, phenotypic, or genetic. In this work, a problematic aspect of all these tools is considered, which these other tools were not built to address. Despite their success, the tools for gapfilling have significant limitations including the following: (1) gaps are addressed on a per-metabolite basis (as opposed to a whole-model holistic approach), (2) thermodynamic feasibility is often not considered, and (3) reaction direction is not considered in gapfilling, rather all reactions are added reversibly. From the first and second limitations, several problems arise including (1) inability to guarantee that the minimum number of reactions are added to fix metabolic gaps on a whole-model (holistic) basis; (2) inability to identify and avoid unfavorable interactions between multiple gap fixes (often, TICs); and (3) differences in the resultant model dependent of the individual curator.

To address current TIC-finding and gapfilling method limitations, this work introduces a multi-step optimization-based MILP method. The first step is to solve an iterative optimization-based TIC-Finding problem (TFP), which identifies potential TICs, which may be caused by adding reactions from a database in a given direction (see Figure 1). This method uses optimization and binary variables as opposed to null space matrices used by other methods that identify reactions participating in TICs (Saa and Nielsen, 2016) or TICs (Chan et al., 2018) and thus can provide a greater level of detail for each inherent or potential TIC. This problem is unique as it considers the direction and relative flux rate of reactions participating in

TICs and can be easily adapted for the purposes of model curation sans database for the resolution of inherent TICs. The second step involves the solving of three optimization-based problems, the connecting problems (CPs), which are highly similar but have different objectives. The first connecting problem (CP1) is the maximization of model metabolites successfully connected to metabolic network, e.g. maximizing the number of metabolites that the connected model can now produce, while avoiding the addition of TICs. The second connecting problem (CP2) is the minimization of the number of reactions required to achieve the objective of CP1. The third connecting problem (CP3) is the maximization of the number of reactions to be added reversibly from the database to achieve the objectives of CP1 and CP2 subject to avoiding TICs. The connecting problems are unique in that, unlike other gapfilling algorithms, CP solutions provide whole model gapfilling solutions guaranteeing the minimum number of reactions being added for the maximum number of fixed metabolites. As proof of concept, the OptFill approach is applied to three test stoichiometric models of increasing sizes (models of 28–210 reactions, databases of 17–77 reactions) with designed metabolic gaps and one smaller (1074 reactions) GSM of *Escherichia coli* with acknowledged metabolic gaps (Reed et al., 2003) using another GSM of *E. coli* as the basis for a database (Feist et al., 2007). With the computational resources at hand, the full OptFill method is limited to relatively smaller stoichiometric models and databases but should be applicable to larger models and databases given access to greater computational power.

RESULTS

Development of OptFill

OptFill was conceived and developed to address the limitations of the current state-of-the-art GapFind/GapFill (Satish Kumar et al., 2007) tool. The initial stages of the design-build-test (DBT) cycle contained the first test model (TM1) and the first test database (TDb1) and involved only a single connecting problem. TM1 was constructed as a small stoichiometric model involving starch and glycolysis metabolism to produce ethanol but with metabolic gaps preventing growth (see Figure 1). TDb1 was designed to have the capacity to fill these gaps, at the expense of potentially producing TICs. In the DBT cycle, it was soon realized that the TFP was necessary to define the potential TICs that might occur. The TFP was built to solve for the smallest TICs (i.e., the TICs with the smallest number of participant reactions) first and then solve for larger TICs to prevent multiple TICs masquerading as a single TIC solution. The workflow representing the TFP is shown in Figure 2. The CPs were developed to ensure consistency in the number, order, and identity of the CP solutions while avoiding the addition of the whole set of TICs identified as potentially occurring between the model and database. See Figure 3 for the conceptual formulation of each type of problem. All problems that are part of the OptFill tool are mixed integer linear programming (MILP) problems that ensure global optimality of each solution in each iteration.

On occasion, the feasibility constraints used might be too strict to return a feasible solution to the CP problems, which could result in execution errors prematurely ending OptFill before completion. Therefore, an error handling framework was built around each CP problem allowing a one-time relaxation of feasibility constraints. These frameworks are shown in Figure 2. OptFill is ended when CP1 no longer has a feasible solution even when feasibility constraints are relaxed (which occurs because previous solutions are prevented from being re-identified) because at that point none of the CP2 and CP3 will have a feasible solution. Further, all OptFill runs described used non-standard CPLEX solver options, which effectively eliminated most types of cuts. This caused some level of reduction to the solution space, particularly those that could result in non-optimal solutions being reported as optimal. These included flow, zero-half, and Gomory fractional cuts, among others. This was done because the order of solutions is important in the OptFill method, and the order of solutions also has bearing on the number of solutions returned. See Transparent Methods for further detail.

Application of OptFill to Test Models

After finalizing the formulation (see Figure 3 and Transparent Methods) and workflow (Figure 2) of OptFill, a detailed analysis of OptFill results with respect to TM1 and TDb1 was undertaken. Some qualitative results of the application of the OptFill workflow to TM1/TDb1 are shown in Figure 1, which include the initial model and database (Figure 1), the combination of the model and database (Figure 1), selected identified potential TICs (Figure 1), and selected identified CPs' solutions (Figure 1). As is shown in Figure 1, TM1 is too disconnected to produce biomass but in combination with TDb1 can potentially produce biomass. When the TFP is applied (Figure 1), 31 potential TICs consisting of 3–12 reactions (hereafter, sizes 3–12) were identified. The average solution time (when a solution was found) was 0.175 s (σ = 0.0727 s, min =

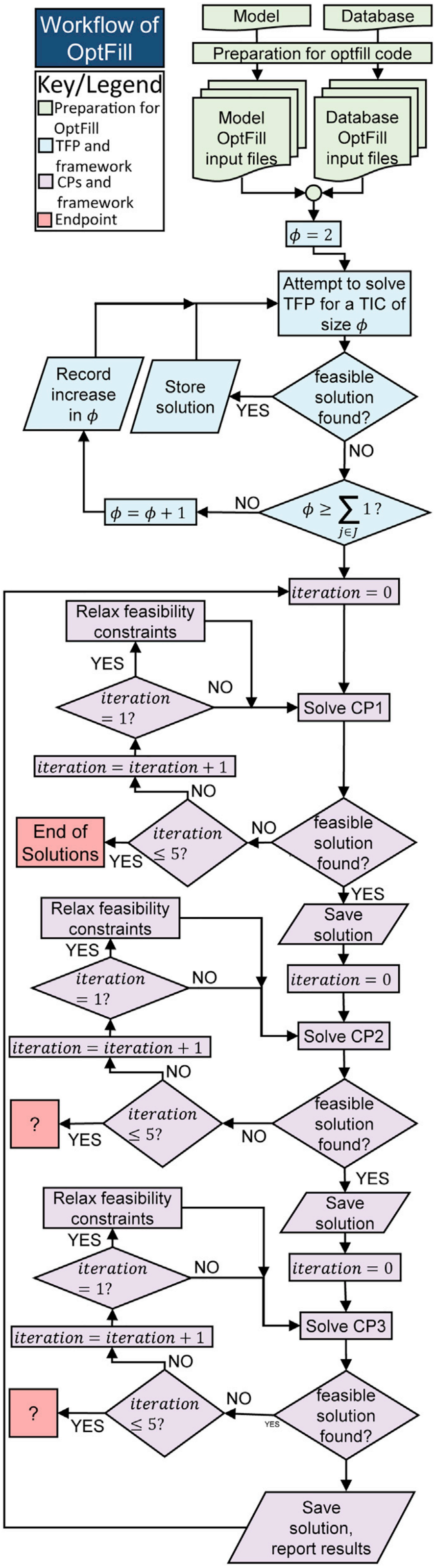


Figure 2. Workflow of OptFill

This is a workflow diagram of the OptFill tool. Green nodes represent the preparatory workflow, blue the workflow of the TIC-finding problem (TFP), purple the workflow of the connecting problems (CPs) including the error-handling workflow imbedded in the CPs workflow, and red the endpoints of the workflow. It should be noted that only one endpoint truly exists, when a solution to CP1 is not found, because the other problems, CP2 and CP3, will have solutions if CP1 has a solution, hence the workflow exit points being represented by a question mark at these points.

0.0870 s, max = 0.378 s). It should be noted that all solve times reported here are not constant, even if using same resources. Figure 1 highlights five potential TICs that were identified. The first two TICs identified, TIC #1 and #2, show that the TFP can identify TICs occurring only in the database; that TICs consisting of the same metabolites and reactions are identified separately if reaction directions are different; and that two of the smallest TICs are identified. Potential TIC #9 shows a TIC of moderate size (for TM1/TDb1), which contains an irreversible model reaction related to non-growth-associated maintenance (NGAM) and, therefore, will not have a companion potential TIC of opposite direction, unlike potential TIC #1 and TIC #2 (in the opposite direction). Further, this highlights the potential for infeasible cycling, which effectively negates the cost of NGAM of the model. If added in its entirety, NGAM would be irrelevant to the model at any value and would significantly reduce model accuracy. This TIC might not be manually identified because NGAM is usually a fixed quantity. Potential TIC #10 highlights another type of infeasible cycling involving ADP/ATP, but this cycling essentially negates the cost of phosphorylation/dephosphorylation of glucose-6-phosphate isomers. Finally, potential TIC #31 is included to highlight a non-intuitive TIC, in addition to be the largest TIC identified. This TIC involves the separate cycling of sugars and 3-carbon molecules linked and is made possible by ADP/ATP cycling (sugar cycling consumes ATP and 3-carbon cycling produces ATP). These examples illustrate that many, but not all, potential TICs involve the infeasible cycling of energy molecules, which should be particularly avoided in the reconstruction of models of metabolism, as this can result in negated costs for various biological activities with which a cost should be associated. This negated cost can often result in increased model growth rate and reaction fluxes, reducing the model's accuracy.

The model, database, and TFP solutions form the input for the CPs. Before solving the CPs, a modified

A

Conceptual Formulation of TFP

Minimize number of reactions in TIC

Subject to

- ➢ Bounds on each reaction fluxed based on reaction direction
- ➢ Determination if each reaction is participating in the TIC
- ➢ Mass balance
- ➢ Number of reaction in TIC is Φ
- ➢ Determination of each reaction direction, if participating
- ➢ No repeated solutions

B

Conceptual Formulation of CPs

CP1: Maximize number of connected model metabolites

CP2: Minimize number of reactions

CP3: Maximize number of reversible reactions

Subject to

- ➢ (all) Bounds on each reaction fluxed based on reaction direction
- ➢ (all) Determination if each reaction is participating in the CP solution
- ➢ (all) Determination of each reaction direction, if participating
- ➢ (all) Determination if each metabolite can be produced
- ➢ (all) Mass balance
- ➢ (all) No repeated solutions
- ➢ (all) No TICs identified by the TFP
- ➢ (CP2&3) fixed number of connected model metabolites
- ➢ (CP3) fixed number of reactions in solution

Figure 3. Conceptual Formulation of Each Problem in OptFill

This figure gives a conceptual formulation of the TIC-finding problem, TFP, in part (A) and the connecting problems, CP1, CP2, and CP3, in part (B). In part (B), as three connecting problems are solved, each conceptual constraint has indicated CPs to which it is applied. Conceptual constraints may require multiple mathematical constraints to be realized, see Transparent Methods for mathematical formulation.

version of CP1 was run, which prohibited the addition of database reactions. This modified CP1 reported that the raw TM3 model was capable of producing no metabolites. The CPs, when applied to TM1 and TDb1, identified 24 potential solutions that connected between 31 and 33 metabolites with the additions of 6–10 reactions, of which 0 to 6 could be reversible without TICs. The average time to solve all three CPs for each solution was 0.639 s (σ = 0.147 s, min = 0.433 s, max = 0.950 s) (see Figure 4). From the FBA performed on each connecting problem solution with the objective of maximization of biomass, the mean maximum biomass production rate of the set of connected models was 2.43 h^{-1} (σ = 0.394 hr^{-1}, min = 1.44 hr^{-1}, max = 2.90 hr^{-1}). Solution times for the FBA code were not recorded, as FBA solution time is generally low. Two connecting problem solutions, the first and the last, are shown in Figure 1. These solutions are notably different in terms of the number of model metabolites connected by the CPs' solution (green boxes in the metabolic sketch), the number of intermediate metabolites introduced by these solutions (yellow boxes), the number of database reactions introduced (orange arrows), and even the use of energy molecules. For instance, CPs' solution 1 introduces only two additional metabolites and six reactions reversibly from the database, which are part of the CPs' solution and connects all but two model metabolites. The first is acetate, which is a dead-end metabolite. The second is the extracellular proton, which suggests that the model is small enough that all protons produced are also consumed. This solution has the slowest growth rate of all connecting problem solutions. On the other hand, CPs' solution 24 connects two fewer metabolites than CPs' solution 1, requires two more reactions, introduces two more intermediate metabolites, and has a higher growth rate. It is hypothesized that this is due to the more efficient production of ATP allowed by reaction R01512[c] (enzyme ATP:3-phospho-D-glycerate 1-phosphotransferace in the cytosol), which is present in many other high-biomass solutions. This reaction allows two dephosphorylation events to produce ATP, as opposed to only one (the other event occurring by hydrolysis).

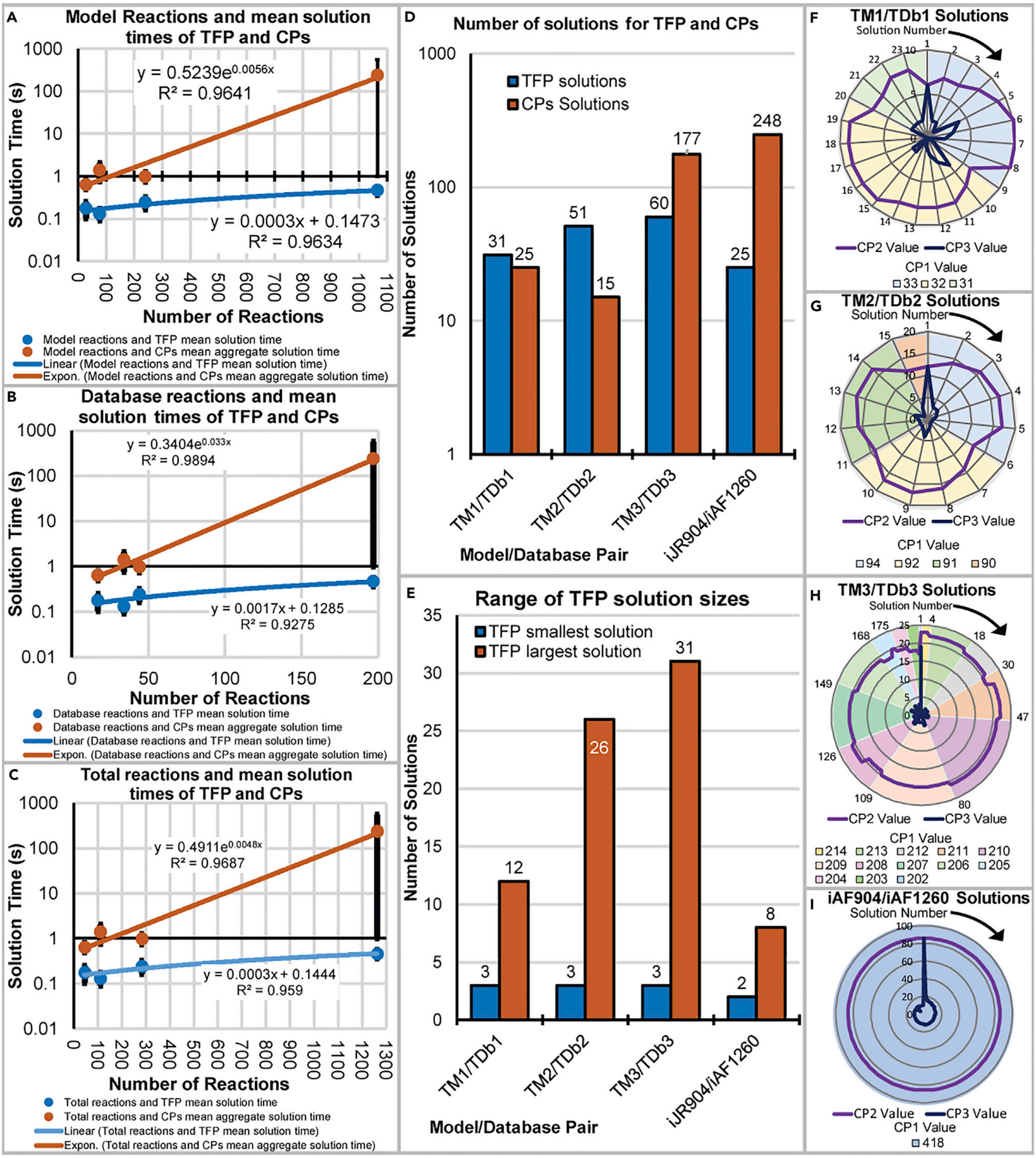


Figure 4. Visualization of OptFill Solution Time and Results

This figure show the trends in solution time, (A) through (C), of the TIC-finding problem (TFP, blue) and the connecting problems (CPs, brown) with trend lines with the highest Pearson's correlation coefficient of linear, exponential, power, and logarithmic fits. These trends are considered with respect to the number of reactions in the model (A), database (B), and total reactions (C). Parts (D) through (F) highlight the trends of solutions. Part (D) highlights the number of solutions found by the TFP and CPs; part (E) highlights the range in size of the identified potential TICs by the TFP. Parts (F), (G), (H), and (I) highlight the variety of CPs' solutions. In these figures, the pie chart indicates the number of metabolites connected by the CP1 solution, and the radar chart is used to indicate the CP2 solution (number of reactions added) and the CP3 solution (number of those reactions that are added reversibly).

Two larger test models were built next to study the increase in number of solutions and time required to reach those solutions by OptFill and, ultimately, to investigate its scale-up potential. Each test model was built from an OptFill solution of a previous solution to highlight the ability of this tool to be applied in sequence. In the application of OptFill to the existing models of organisms, careful attention must be paid in selection of a CPs' solution to accept, including considerations of energy metabolism, predicted growth rates, and remaining unconnected metabolites. Here, CPs' solution 1 was selected and combined with TM1 as the base of the second test model (TM2). Reactions and metabolites from the fatty acid biosynthesis and the pentose phosphate pathway were added to this base, in which gaps were manually created. Reactions that could address these gaps formed the second test database (TDb2). Redundant metabolic functions were added to TDb2 to allow for potential TICs. Similarly, the third test model (TM3) was built from the first CPs' solution of TM2 and TDb2. Additionally, a bank of reactions from the amino acid synthesis pathways, including redundant functionalities, was created. This bank was automatically (randomly) sorted between those reactions that would be added to complete TM3 (~80% of bank reactions) and those that would constitute the third test database (TDb3, ~20% of bank reactions). As random sorting was used, a modified version of the TIC-finding problem (modified TIC-finding problem, mTFP), was used to identify inherent TICs in TM3 and TDb3, which resulted from the random assortment of the bank reactions. The reactions most commonly participating in identified inherent TM3 TICs were moved to the TDb3 until no inherent TICs remained (five reactions in total).

For OptFilling of TM2/TDb2, 51 TICs consisting of 3–26 reactions were identified by the TFP, with a mean solution time of 0.131 s ($\sigma = 0.0405$ s, min = 0.0850 s, max = 0.308 s). The largest TIC, potential TIC #51 consisting of 26 reactions, would have largely been very difficult to be identified by a non-automated method, as it spans six KEGG pathways including glycolysis, the pentose phosphate pathway, purine metabolism, nicotinate and nicotinamide metabolism, starch and sucrose metabolism, and riboflavin metabolism. TIC #51 involves the cycling of 3-, 4-, 5-, and 6-carbon molecules, energy molecules (ATP, NADH, and NADPH), and energy molecule hydrolysis. This TIC can be found in GitHub and Mendeley Data repositories accompanying this work.

Before solving the CPs, the modified CP1 was run and reported that the raw TM2 model was capable of producing no metabolites. Fifteen potential CPs' solutions were identified, which each connected 90 to 94 metabolites with the addition of 17–23 reactions, of which 0 to 19 could be reversible without TICs. The average time to solve all three CPs for each solution was 1.40 s ($\sigma = 0.639$ s, min = 0.404 s, max = 2.65 s) (see Figure 4). From the FBA performed, the biomass production rate of most CPs' solutions applied to TM2 was 1.31 h^{-1}, for 10 solutions, and 1.36 h^{-1} for the remaining five. In the CPs' solutions, those with the highest biomass have fewer metabolites that could be connected (all solutions with higher biomass production were generated after lower biomass production solutions). Those with the higher biomass production rates generally have one fewer reaction that requires ATP hydrolysis and therefore has slightly more energy in the system to spend on the production of biomass than their lower biomass counterparts.

Similarly, OptFill applied to TM3/TDb3 resulted in the identification of 60 TICs consisting of 3–31 reactions by the TFP and 177 potential CPs' solutions, which each connected 202 to 214 metabolites with 12–17 reactions, of which 1 to 12 could be reversible without TICs. As earlier, the modified CP1 was used to identify 54 metabolites that the raw TM3 was capable of producing. The mean TFP solution time was 0.240 s ($\sigma = 0.0756$ s, min = 0.141 s, max = 0.541 s), whereas the mean CPs' solution time was 0.985 s ($\sigma = 0.249$ s, min = 0.573 s, max = 1.86 s). From the FBA performed, the mean biomass production rate of the connected model was 3.29 h^{-1} ($\sigma = 0.179$ h^{-1}, min = 3.11 h^{-1}, max = 3.47 h^{-1}). Runtime and solution metrics for all solutions are shown in Figure 4. Unlike TM1 and TM2 OptFilling results, there was no solution where all database reactions to be added by the CPs' solution could be added reversibly. This indicates that, for all solutions, the direction in which database reactions are added is important to avoid TICs to produce a model without the disadvantages of TICs described previously. Furthermore, the biomass production rate does not appear as dependent on either the number of metabolites connected or reactions added as in previous CPs' solution sets. Instead, the biomass production rate seems to most depend on the method of sulfate assimilation.

Application of OptFill to *i*JR904

In order to show how the OptFill workflow might scale up to a GSM, the *i*JR904 model of *Eschericia coli* consisting of 761 metabolites, 1,074 reactions, and 904 genes (Reed et al., 2003) was selected as the base model to fix. The *i*AF1260 model, a model extending onto *i*JR904, consisting of 1,598 metabolites; 2,381 reactions; and 1,260 genes (Feist et al., 2007) was selected to serve as the set of reactions from which to build the database. *i*JR904 contains 70 dead-end metabolites (Reed et al., 2003) that need fixing. Before applying OptFill, some minor formatting changes were made (described in Transparent Methods and in the related GitHub and Mendeley Data repositories accompanying this work), and it was decided that carbon-limited aerobic growth using acetate would be the condition for which *i*JR904 model would be fixed. Metabolite exchange rates were taken from Reed et al. (2003) to describe this growth condition.

In order to create the database that would be applied to *i*JR904, all *i*AF1260 exchange reactions and reactions with names identical to those in *i*JR904 (which were assumed to be the same reaction as the former was built from the latter) were removed from *i*AF1260 to form the initial database that consisted of 1,441 reactions. This proved too computationally intensive for the resources, and therefore this database was further simplified in a manner that it is suggested others with limited computational resources might also use. First, the *i*AF1260-based database and *i*JR904 were combined in single model file, and flux variability analysis (FVA) (Gudmundsson and Thiele, 2010) was performed (see Table S1. iJR904, Related to Figure 4). Those *i*AF1260 reactions capable of holding flux as determined by FVA (715 reactions) were defined as the database of functionalities to be used with OptFill.

OptFill was performed on *i*JR904 using this database. This still resulted in a slow OptFill process; therefore, solutions that were reported (i.e., 4 identified) in the allotted solve time of 24 h were collected. All *i*AF1260 reactions that participated in at least one solution (a total of 182 reactions) were selected as the basis of the third *i*AF1260-based database. This resulted in significantly lower computational requirements for the application of OptFill. This database was found, upon application of OptFill, to be without TICs. For the purposes of demonstration and showing how the increase of TFP solution time changes with model and database size, it was arbitrarily decided to add six reactions manually from the previous database, which could participate in potential TICs between the model and database but which did not create TICs only within the database. Further, the mTFP was applied to the *i*JR904 model. From the mTFP results, it was noticed that in *i*JR904, some reactions were included in the model twice, both as reversible and irreversible, causing inherent TICs in the *i*JR904 model involving these duplicate reactions. It was decided to move the irreversible reactions of each duplicate pair to the database (nine reactions in total) so that all *i*JR904 models were still present in the OptFill in some capacity. The final *i*AF1260-based database for the OptFilling of *i*JR904 totals 188 reactions. Initial, final, and intermediate iAF1260-based databases used can be found in Table S1. iJR904, related to Figure 4 iJR904 and in the GitHub and Mendeley Data repositories accompanying this work.

Demonstrated here is a procedure by which the database applied to a model can be significantly decreased in size to reduce computational cost of the OptFill method, while still effectively addressing metabolic gaps. This can be summarized as follows: (1) eliminate all duplicate reactions; (2) perform FVA on a pseudomodel that is a combination of the database and model and use the results to eliminate reactions that cannot carry flux; and (3) perform OptFill using databases with larger solution time, collect a few sample solutions, and use the set of reactions participating in sampled solutions as the database. Applications of steps (1) and (2) as well as iterative applications of (3) might be used by modelers to shrink the database used in OptFilling to a size that is possible to solve in a modest period of time given the computational resources available.

This final *i*AF1260-based database was used to OptFill *i*JR904 model. In this final iteration, there were 25 TICs of size two to eight reactions identified. The associated mean TFP solution time was 0.410 s (σ = 0.0978 s, min = 0.330 s, max = 0.687 s). The TICs identified were generally simple, as they stem from reactions manually added to the database, which cause TICs. Eleven TICs occur between just two reactions, and a further four involving only a single database reaction. Each of these effectively precluded a single database reaction from being added in a certain direction. When the CPs were applied to *i*JR904, it was found that the CPs' solution time had increased considerably from that of other models, to a mean of 236 s (σ = 329 s, min = 15.3 s, max = 1010 s). The solution time of this model was significantly increased due to disabling of many types of cuts that a solver might use to decrease solution time but that lead to non-optimal solutions being reported as optimal. These are particularly relevant because minor cuts, such as those that accept a 0.5% reduction in the optimal solution value, can change the number of metabolites connected by the CPs by two or more for GSMs. As the order of solutions is important, even these minor relaxations were deemed problematic and were therefore mostly disabled, leading to increased solution time. If these cuts were allowed, CPs' solution time would have been approximately an order of magnitude less than reported here. The modified CP1 problem reported that the *i*JR904 model was capable of producing 358 metabolites under the given aerobic growth on acetate conditions, and all CPs' solutions connected 418 metabolites with the addition of 86 reactions. All CPs' solutions produced biomass at a rate of 0.108 h^{-1}. This is likely a result of the database reduction steps taken. The variation on the CPs' solution occurred in the number of connecting reactions that could be added reversibly, ranging from 5 to 86. The full set of solutions can be found in the GitHub and Mendeley Data repositories accompanying this work. It can be seen in Figure 4I that efforts to prevent non-optimal solutions from being reported as optimal were not entirely successful. There exists one CPs' solution, solution #72, where the optimal (maximum) CP3 solution value is 5, whereas the optimal (maximum) CP3 solution value was 11 from solutions #71 and #73. This occurred when all solutions were subject to approximately the same constraints (save the integer cuts necessary to prevent repeated solutions). It is noted earlier that many types of cuts were disabled, but not all, and one type of cut or other solver setting allowed this non-optimal solution to be reported as optimal; however, eliminating all such cuts and settings proved prohibitively time-consuming. Therefore, the settings, which can be found in the GitHub and Mendeley Data repositories accompanying this work, were selected as those that, for this work, best balanced solution order and solution time.

In the OptFilling solutions of *i*JR904, several trends can be noticed that were not present in the smaller test models. First, when performing FBA, with the objective of maximizing biomass, on the resultant OptFilled *i*JR904 model, not all reactions from the database held flux when biomass was maximized. This is because these reactions make it possible for the model to produce metabolites that are not required for the production of biomass or provide an alternative pathway for the production of biomass that might be less efficient. This does not mean that these connected metabolites are unimportant under other, equally valid, objective functions, for instance the connected metabolites may be bioproduction targets. Further, some TICs exist between *i*JR904 model reactions in the OptFill solutions and notably one database reaction. For most model reactions, these TICs occur because forward and reverse reactions are written separately. The TIC involving the database reaction resulted from the proton uptake exchange reaction being allowed a very high reaction flux in the *i*JR904 model. The TFP was performed with all exchange reactions fixed to a flux of zero; therefore, the TFP did not identify this TIC, which involved an exchange reaction. When the exchange reactions were allowed to carry flux again in the CPs, the high proton uptake rate (here, 1000 mmol/gDW·h) allowed the cycling of reactions. These resulting TICs highlight two important considerations in using OptFill. First, the mTFP should be used in combination with manual editing of the model to ensure that the model does not contain inherent TICs, as the usual OptFill workflow will not address inherent TICs. Second, reasonable bounds should be applied to all exchange reactions (such as the proton uptake reaction) and forward and reverse reaction pairs to prevent TICs in the OptFilled model.

OptFill Solution Times

With the caveats of the available resources (see Transparent Methods for information on the software and hardware tools available for this work), the TFP seems to have a per-TIC average solution time with linear dependence ($R^2 \geq 0.89$) on size of model and/or database used (see Figures 4A through 4C). The same procedure was applied to the aggregated CPs' solution time but with significantly different results. Exponential trend lines were able to fit with a high correlation coefficient ($R^2 \geq 0.96$) between model, database, total system size, and CPs aggregated solution time. This is indicative of a strong correlation between CPs aggregate solution time, number of reactions in the total system, and that increasing total system reactions greatly increases CPs aggregate solution time.

DISCUSSION

Introduced here is an optimization-based tool, OptFill, which can be used to increase the automation of the curation of GSMs. This tool can either be used to automate the filling of metabolic gaps in a reconstructed model or to automate the identification of TICs for manual resolution (via mTFP). In this work, the OptFill was applied in sequence to three test models of increasing size as well as to a GSM of *E. coli*, *i*JR904. These applications combined with some solutions for holistically gapfilling metabolic models, the computational expense of the tool, and a method for reducing that expense highlighted the utility of OptFill.

This method has considerable potential to be adapted to other metabolic systems (both eukaryotic and prokaryotic) and is not specific to any identifier system such as KEGG or ModelSeed. For instance, although all test models as well as *i*JR904/*i*AF1260 have been prokaryotic systems, there is no reason why this approach would not similarly work in a eukaryotic organism. Further, the framework is flexible enough that any system of reaction and metabolite identifiers, such as KEGG (Kanehisa et al., 2017), MetaCyc (Caspi et al., 2014), BIGG (King et al., 2016), K-Base (Arkin et al., 2018), or custom identifiers, may be used for metabolites and/or reactions, making this tool applicable to a wide variety of existing GSM-building methods. This was demonstrated in this work as KEGG identifiers were used in the test models, whereas BIGG identifiers were used by the *i*JR904 and *i*AF1260 models (Reed et al., 2003; Feist et al., 2007).

From the observation of TFP solution times, it is evident that the TFP and mTFP could scale-up to genome-scale models of metabolism, as a linear trend line ($R^2 \geq 0.89$) strongly describes the per-TIC solution time given the computational resources at hand. So long as the number of TICs in the system remains reasonable, this portion of OptFill is transferrable to large-scale GSM systems or to situations where computational resources are limited. The transferability of the OptFill method is likely limited by the computational resources available to the end-user, as the aggregate solution time of the three CPs is well described by an exponential trend line ($R^2 \geq 0.97$). This suggests that those without access to powerful computational resources may have difficulty implementing OptFill in a reasonable time frame, unless, for instance the end-user makes trade-offs between the solution order (e.g. each subsequent solution is truly globally optimal) and solution time. These trade-off issues, such as shown in a minor way with the OptFilling of *i*JR904, may likely be fixed by more advanced MILP solvers that are currently available or by advances in optimization that may be made in future.

When implementing OptFill in other systems, a high-quality model and database should be used in order to limit both the number of solutions and the time the OptFill method takes to complete. This is primarily due to the number of feasible and unique combinations possible. For instance, if a multi-step reaction is included in a database in addition to its component reaction steps, this can potentially double the number of solutions found by both the TFP and CPs. To explain, if the multi-step reaction participates in *n* TICs, then its component step reactions would participate in *n* TICs. This results in *2n* TICs, where only *n* TICs need be identified. The same argument applies for CPs' solutions. This error in model reconstruction could then double (or more) the number of TICs and CPs' solutions as well as the total OptFill runtime in a stroke. In larger models, such issues can result in a significant expenditure of time (potentially days) and computational resources that need not be expended should the model and database used to be of high quality. Such an issue is elsewhere referred as a combinatorial explosion (Burgard et al., 2003). This was shown in this work in the failure to achieve a reasonable number of solutions or reasonable solution times in the OptFilling of *i*JR904 with a poorly curated database based on *i*AF1260; however, when the database was better curated, reasonable numbers of solutions and solution times were achieved. Therefore, it is important to address as many inherent TICs that occur both in the model and in the database as feasible using the mTFP on both the model and the database to identify and address these TICs.

Although throughout this text reaction cycling in the absence of nutrition (i.e., thermodynamically infeasible cycling) is described as a phenomenon that is to be avoided in GSMs, this is not always the case. In many biological systems, cycling of some type does occur and the absence of that cycling in the models might affect their accuracy. However, cycles included in a GSM should be carefully considered with respect to their biological relevance, magnitude, and effect, particularly when they occur in the absence of nutrition provided to the model. In essence, this work can be used to remove and/or avoid all cycling that can occur in the absence of nutrition provided to the model or to ensure that cycles retained are deliberate and have biological relevance if included. If cycles occur in a GSM model in the absence of nutrition provided to the model and are biologically relevant, best practice should be to use other literature data available to limit the scope of the cycling to feasible number. This trade-off must be considered when applying the OptFill algorithm or when choosing to use some type of algorithm that employs the loopless constraints.

This is the essential difference between what is proposed here as the OptFill tool and other algorithms such as the algorithm employed by Chan et al. (2018) to identify all TICs in a model and avoid them. The TIC finding portions of the algorithm are largely equivalent, although Chan et al. (2018) may identify TICs faster. OptFill then precludes these TICs from being added as part of a gapfilling solution so that the resultant reconstructed metabolic model contains no inherent TICs. However, Chan et al. (2018) accepts these TICs in the reconstructed network and seeks to limit flux through these TICs so that the resulting model fluxes are feasible. The OptFill approach presents an alternative to the need to use loopless algorithms on the gapfilled model and allows use of algorithms that are sensitive to the presence of TICs, such as OptForce (Burgard et al., 2003; Chan et al., 2018) without modifying these algorithms for the use of various loopless algorithms that may be computationally expensive.

In future, this work will be used as a gapfilling and curation strategy for the development of GSMs of any prokaryotic and eukaryotic systems. In concert with advances in optimization solvers and available computational resources, these methods (i.e., the TFP, CPs, and their modified versions) will provide an alternative holistic method of model curation. At present, those model-building tools with high computational power at their disposal, such as ModelSeed (Overbeek et al., 2005) and K-Base (Arkin et al., 2018), may well be able to implement OptFill and its components for large GSMs to improve their automated curation capabilities. In addition, with the available computational resources and some adjustments (as explained earlier), Optfill is being implemented to improve the connectivity and predictive capability of the GSM of a non-model purple non-sulphur bacterium (Alsiyabi et al., 2019) and to develop the GSM of a melanized fungal strain.

Limitations of the Study

As already discussed in this work, this study does have multiple limitations. These limitations include solution speed, both of the CPs and as an overall result of combinatorial explosion; the need for powerful computational resources to efficiently use this tool; and that this tool might miss cycles that do occur in biological systems but require thermodynamic data or constraints to prevent infeasible cycling.

METHODS

All methods can be found in the accompanying Transparent Methods supplemental file.

DATA AND CODE AVAILABILITY

The published article does not include all datasets and code generated or analyzed during this study. The datasets and code generated during this study are available at GitHub in the ssbio/OptFill repository [https://doi.org/10.5281/zenodo.3560302] and Mendeley Data OptFill repository [https://doi.org/10.17632/npdwbmb7d7.1].

SUPPLEMENTAL INFORMATION

Supplemental Information can be found online at https://doi.org/10.1016/j.isci.2019.100783.

ACKNOWLEDGMENTS

This work has been completed utilizing the Holland Computing Center of the University of Nebraska, which receives support from the Nebraska Research Initiative, United States of America. The authors gratefully acknowledge funding from UNL Faculty Startup Grant, United States of America, and Nebraska Center for Energy Sciences Research, University of Nebraska-Lincoln (NCESR), United States of America, Grant to R.S..

AUTHOR CONTRIBUTIONS

Conceptualization, W.L.S. and R.S.; Data curation, W.L.S.; Formal analysis, W.L.S.; Funding Acquisition, R.S.; Investigation, W.L.S.; Methodology, W.L.S.; Project administration, R.S.; Resources, R.S.; Software, W.L.S.; Supervision, R.S.; Validation, W.L.S.; Visualization, W.L.S.; Writing—original draft, W.L.S. and R.S.; Writing—reviewing & editing, W.L.S. and R.S.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: October 7, 2019
Revised: December 3, 2019
Accepted: December 12, 2019
Published: January 24, 2020

REFERENCES

Alsiyabi, A., Immethun, C.M., and Saha, R. (2019). Modeling the interplay between photosynthesis, CO2 fixation, and the quinone pool in a purple non-sulfur bacterium. Sci. Rep. *9*, 1–9.

Andersen, M.R., Nielsen, M.L., and Nielsen, J. (2008). Metabolic model integration of the bibliome, genome, metabolome and reactome of Aspergillus Niger. Mol. Syst. Biol. *4*, 178.

Arkin, A.P., Cottingham, R.W., Henry, C.S., Harris, N.L., Stevens, R.L., Maslov, S., Dehal, P., Ware, D., Perez, F., Canon, S., et al. (2018). KBase: the United States department of energy systems biology knowledgebase. Nat. Biotechnol. *36*, 566–569.

Beyer, P., Al-Babili, S., Ye, X., Lucca, P., Schaub, P., Welsch, R., and Potrykus, I. (2002). Golden Rice: introducing the beta-carotene biosynthesis pathway into rice endosperm by genetic engineering to defeat vitamin A deficiency. J. Nutr. *132*, 506S–510S.

Brunk, E., Sahoo, S., Zielinski, D.C., Altunkaya, A., Dräger, A., Mih, N., Gatto, F., Nilsson, A., Preciat Gonzalez, G.A., Aurich, M.K., et al. (2018). Recon3D enables a three-dimensional view of gene variation in human metabolism. Nat. Biotechnol. *36*, 272–281.

Burgard, A.P., Pharkya, P., and Maranas, C.D. (2003). OptKnock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. Biotechnol. Bioeng. *84*, 647–657.

Caspi, R. (2006). MetaCyc: a multiorganism database of metabolic pathways and enzymes. Nucleic Acids Res. *34*, D511–D516.

Caspi, R., Altman, T., Billington, R., Dreher, K., Foerster, H., Fulcher, C.A., Holland, T.A., Keseler, I.M., Kothari, A., Kubo, A., et al. (2014). The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases. Nucleic Acids Res. *42*, 459–471.

Chan, S.H.J., Wang, L., Dash, S., and Maranas, C.D. (2018). Accelerating flux balance calculations in genome-scale metabolic models by localizing the application of loopless constraints. Bioinformatics *34*, 4248–4255.

Chowdhury, R., Chowdhury, A., and Maranas, C.D. (2015). Using gene essentiality and synthetic lethality information to correct yeast and CHO cell genome-scale models. Metabolites *5*, 536–570.

Cuevas, D.A., Garza, D., Sanchez, S.E., Rostron, J., Henry, C.S., Vonstein, V., Overbeek, R.A., Segall, A., Rohwer, F., Dinsdale, E.A., et al. (2019) Elucidating genomic gaps using phenotypic profiles [version 2; peer review: 1 approved, 1 approved with reservations], (May), pp. 1–28.

Feist, A.M., Henry, C.S., Reed, J.L., Krummenacker, M., Joyce, A.R., Karp, P.D., Broadbelt, L.J., Hatzimanikatis, V., and Palsson, B.Ø. (2007). A genome-scale metabolic reconstruction for Escherichia coli K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information. Mol. Syst. Biol. *3*, 1–18.

Fritzemeier, C.J., Hartleb, D., Szappanos, B., Papp, B., and Lercher, M.J. (2017). Erroneous energy-generating cycles in published genome scale metabolic networks: identification and removal. PLoS Comput. Biol. *13*, 1–14.

Gomes de Oliveira Dal'Molin, C., Quek, L.E., Saa, P.A., and Nielsen, L.K. (2015). A multi-tissue genome-scale metabolic modeling framework for the analysis of whole plant systems. Front. Plant Sci. *6*, 1–12.

Gudmundsson, S., Agudo, L., and Nogales, J. (2017). Applications of Genome-Scale Metabolic Models of Microalgae and Cyanobacteria in Biotechnology, Microalgae-Based Biofuels and Bioproducts: From Feedstock Cultivation to End-Products (Elsevier Ltd). https://doi.org/10.1016/B978-0-08-101023-5.00004-2.

Gudmundsson, S., and Thiele, I. (2010). Computationally efficient flux variability analysis. BMC Bioinformatics *11*, 2–4.

Hall, R.D., Brouwer, I.D., and Fitzgerald, M.A. (2008). Plant metabolomics and its potential application for human nutrition. Physiol. Plant. *132*, 162–175.

Henry, C.S., DeJongh, M., Best, A.A., Frybarger, P.M., Linsay, B., and Stevens, R.L. (2010). High-throughput generation, optimization and analysis of genome-scale metabolic models. Nat. Biotechnol. *28*, 977–982.

Islam, M.M., Al-Siyabi, A., Saha, R., and Obata, T. (2018). Dissecting metabolic flux in C 4 plants: experimental and theoretical approaches. Phytochem. Rev. *17*, 1253–1274.

Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y., and Morishima, K. (2017). KEGG: New perspectives on genomes, pathways, diseases and drugs. Nucleic Acids Res. *45*, D353–D361.

Karp, P.D., Weaver, D., and Latendresse, M. (2018). How accurate is automated gap filling of metabolic models? BMC Syst. Biol. *12*, 1–11.

Kim, T.Y., Sohn, S.B., Kim, Y.B., Kim, W.J., and Lee, S.Y. (2012). Recent advances in reconstruction and applications of genome-scale metabolic models. Curr. Opin. Biotechnol. *23*, 617–623.

King, Z.A., Lu, J., Dräger, A., Miller, P., Federowicz, S., Lerman, J.A., Ebrahim, A., Palsson, B.O., and Lewis, N.E. (2016). BiGG Models: a platform for integrating, standardizing and sharing genome-scale models. Nucleic Acids Res. *44*, D515–D522.

Latendresse, M., and Karp, P.D. (2018). Evaluation of reaction gap-filling accuracy by randomization. BMC Bioinformatics *19*, 1–13.

Limviphuvadh, V., Tan, C.S., Konishi, F., Jenjaroenpun, P., Xiang, J.S., Kremenska, Y., Mu, Y.S., Syn, N., Lee, S.C., Soo, R.A., et al. (2018). Discovering novel SNPs that are correlated with patient outcome in a Singaporean cancer patient cohort treated with gemcitabine-based chemotherapy. BMC Cancer *18*, 1–16.

Liu, J., Gao, Q., Xu, N., and Liu, L. (2013). Genome-scale reconstruction and in silico analysis of Aspergillus terreus metabolism. Mol. BioSyst. *9*, 1939–1948.

Magnúsdóttir, S., Heinken, A., Kutt, L., Ravcheev, D.A., Bauer, E., Noronha, A., Greenhalgh, K., Jäger, C., Baginska, J., Wilmes, P., et al. (2016). Generation of genome-scale metabolic reconstructions for 773 members of the human gut microbiota. Nat. Biotechnol. *35*, 81–89.

De Martino, D., Capuani, F., Mori, M., De Martino, A., and Marinari, E. (2013). Counting and correcting thermodynamically infeasible flux cycles in genome-scale metabolic networks. Metabolites *3*, 946–966.

Ng, C.Y., Jung, M.Y., Lee, J., and Oh, M.K. (2012). Production of 2,3-butanediol in Saccharomyces cerevisiae by in silico aided metabolic engineering. Microb. Cell Fact. *11*, 68.

Nigam, R., and Liang, S. (2007). Algorithm for perturbing thermodynamically infeasible metabolic networks. Comput. Biol. Med. *37*, 126–133.

Orth, J.D., Thiele, I., and Palsson, B.O. (2010). What is flux balance analysis? Nat. Biotechnol. *28*, 245–248.

Overbeek, R., Begley, T., Butler, R.M., Choudhuri, J.V., Chuang, H.Y., Cohoon, M., de Crécy-Lagard, V., Diaz, N., Disz, T., Edwards, R., et al. (2005). The subsystems approach to genome annotation and its use in the project to annotate 1000 genomes. Nucleic Acids Res. *33*, 5691–5702.

Pitkänen, E., Jouhten, P., Hou, J., Syed, M.F., Blomberg, P., Kludas, J., Oja, M., Holm, L., Penttilä, M., Rousu, J., and Arvas, M. (2014). Comparative genome-scale reconstruction of gapless metabolic networks for present and ancestral species. PLoS Comput. Biol. *10*, e1003465.

Ranganathan, S., Suthers, P.F., and Maranas, C.D. (2010). OptForce: an optimization procedure for identifying all genetic manipulations leading to targeted overproductions. PLoS Comput. Biol. *6*, e1000744.

Reed, J.L., Vo, T.D., Schilling, C.H., and Palsson, B.O. (2003). An expanded genome-scale model of Escherichia coli K-12 (iJR904 GSM/GPR). Genome Biol. *4*, 1–12.

Saa, P.A., and Nielsen, L.K. (2016). Fast-SNP: a fast matrix pre-processing algorithm for efficient loopless flux optimization of metabolic models. Bioinformatics *32*, 3807–3814.

Saha, R., Verseput, A.T., Berla, B.M., Mueller, T.J., Pakrasi, H.B., and Maranas, C.D. (2012). Reconstruction and comparison of the metabolic potential of cyanobacteria Cyanothece sp. ATCC 51142 and Synechocystis sp. PCC 6803. PLoS One *7*, e48285.

Saha, R., Liu, D., Hoynes-O'Connor, A., Liberton, M., Yu, J., Bhattacharyya-Pakrasi, M., Balassy, A., Zhang, F., Moon, T.S., Maranas, C.D., and Pakrasi, H.B. (2016). Diurnal regulation of cellular processes in the Cyanobacterium Synechocystis sp. strain PCC 6803: insights from transcriptomic. MBio *7*, 1–14.

Saha, R., Suthers, P.F., and Maranas, C.D. (2011). Zea mays iRS1563: a comprehensive genome-scale metabolic reconstruction of maize metabolism. PLoS One *6*, e21784.

Satish Kumar, V., Dasika, M.S., and Maranas, C.D. (2007). Optimization based automated curation of metabolic reconstructions. BMC Bioinformatics *8*, 1–16.

Schellenberger, J., Lewis, N.E., and Palsson, B. (2011). Elimination of thermodynamically infeasible loops in steady-state metabolic models. Biophys. J. *100*, 544–553.

Shoaie, S., Karlsson, F., Mardinoglu, A., Nookaew, I., Bordel, S., and Nielsen, J. (2013). Understanding the interactions between bacteria in the human gut through metabolic modeling. Sci. Rep. *3*, 2532.

Simons, M., Saha, R., Amiour, N., Kumar, A., Guillard, L., Clément, G., Miquel, M., Li, Z., Mouille, G., Lea, P.J., et al. (2014). Assessing the metabolic impact of nitrogen availability using a compartmentalized maize leaf genome-scale model. Plant Physiol. *166*, 1659–1674.

Srinivasan, S., Cluett, W.R., and Mahadevan, R. (2015). Constructing kinetic models of metabolism at genome-scales: a review. Biotechnol. J. *1359*, 1345–1359.

Thiele, I., and Palsson, B.Ø. (2010). A protocol for generating a high-quality genome-scale metabolic reconstruction. Nat. Protoc. *5*, 93–121.

UniProtKB (2018). E. coli K12. www.uniprot.org/uniprot/?query=E.+coli+K-12+strain+1655&sort=score.

iScience, Volume *23*

Supplemental Information

OptFill: A Tool for Infeasible
Cycle-Free Gapfilling
# of Stoichiometric Metabolic Models

Wheaton L. Schroeder and Rajib Saha

TRANSPARENT METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Software and Algorithms		
Generalized Algebraic Modeling System (GAMS) version 24.7.4	GAMS Development Corp. (URL: www.gams.com/)	N/A
CPLEX Solver version 12.6	GAMS Development Corp. (URL: www.gams.com/)	N/A
Perl version 5.26 (for Unix)	Perl.org (URL: www.perl.org)	N/A
Strawberry Perl version 5.24.0.1 (for Windows)	Perl.org (URL: http://strawberryperl.com/)	N/A
The world-wide-web library for Perl, module 6.39	Perl meta::cpan (URL: metacpan.org/pod/LWP)	N/A
Python version 3.3 (for Unix)	Python Software Foundation (URL: www.python.org)	N/A
Kyoto Encyclopedia of Genes and Genomes (KEGG)	Kanehisa Laboratories (URL: www.kegg.jp)	RRID:SCR_012773
iJR904	Reed et al. 2003	N/A
Other		
Holland Computing Center, Crane Cluster (64 GB RAM, Intel Xenon E5-2670 2.60 GHz processor, 2 CPUs per node)	Holland Computing Center, University of Nebraska (URL: hcc.unl.edu/)	N/A

LEAD CONTACT AND MATERIALS AVAILABILITY

Further information and requests for resources and reagents should be directed to and will be fulfilled by the Lead Contact, Rajib Saha (rsaha2@unl.edu). No materials were generated in this work including cells, DNA, antibodies, reagents, organisms, mouse strains, or ES cells.

Materials Availability Statement

This study has produced several unique software codes in the form of GAMS, Perl, or Python programming languages/tools. These are included in the GitHub (DOI: 10.52.81/zenodo.8475) and Mendeley Data (DOI 10.17632/npdwbmb7d7.1) repositories which accompany this work.

METHOD DETAILS

Model-Database TIC-Finding Problem (TFP)

The first step of the OptFill method requires the iterative solving of the Mixed Integer Linear Programming (MILP) TIC-Finding Problem (TFP) applied to the model and database. This problem is defined below and is designed such that a TIC which could exist between the model and database with given reaction flux bounds will be a solution to the TFP.

$$minimize \sum_{j \in J} \eta_j \tag{1}$$

Subject to (s.t.)

$$\beta_j v_j^{LB} + \epsilon \eta_j \leq v_j \leq (1 - \beta_j) v_j^{UB} - \epsilon \beta_j \qquad \forall j \in J \tag{2}$$

$$\eta_j v_j^{LB} \leq v_j \leq \eta_j v_j^{UB} \qquad \forall j \in J \tag{3}$$

$$\sum_{j \in J} S_{ij} v_j = 0 \qquad \forall i \in I \tag{4}$$

$$\sum_{j \in J} \eta_j = \phi \tag{5}$$

$$\sum_{j_{db} \in J^{DB}} \eta_{j_{db}} \geq 1 \qquad (6)$$

$$\alpha_j \leq \eta_j \qquad \forall j \in J \qquad (7)$$

$$\alpha_j \leq 1 - \beta_j \qquad \forall j \in J \qquad (8)$$

$$\alpha_j \geq \eta_j - \beta_j \qquad \forall j \in J \qquad (9)$$

$$\sum_{j \in J} \alpha_j \left(\alpha'_{s_f,j}\right) \leq \sum_{j \in J} \alpha'_{s_f,j} - \gamma_{s_f} \qquad \forall s_f \in S_f \qquad (10)$$

$$\sum_{j \in J} \beta_j \left(\beta'_{s_f,j}\right) \leq \sum_{j \in J} \beta'_{s_f,j} - \left(1 - \gamma_{s_f}\right) \qquad \forall s_f \in S_f \qquad (11)$$

Fixed Values

$\epsilon = 1E - 3 \equiv a\ small\ number$

$v_j^{LB} \in \mathbb{R} \equiv lower\ bound\ of\ reaction\ j\ flux$

$v_j^{UB} \in \mathbb{R} \equiv upper\ bound\ of\ reaction\ j\ flux$

$S_{ij} \in \mathbb{R} \equiv stoichiometric\ coefficient\ of\ metabolite\ i\ in\ reaction\ j$

$$\alpha'_{s_f,j} = \begin{cases} 1\ if\ reaction\ j\ participates\ in\ previous\ TFP\ solution\ s_f\ with\ positive\ flux \\ 0\ otherwise \end{cases}$$

$$\alpha'_{s_f,j} = \begin{cases} 1\ if\ reaction\ j\ participates\ in\ previous\ TFP\ solution\ s_f\ with\ negative\ flux \\ 0\ otherwise \end{cases}$$

Variables

$$v_j \in \mathbb{R} \equiv flux\ of\ reaction\ j\ in \frac{mmol}{gDW \cdot h}$$

$$\eta_j = \begin{cases} 1\ if\ reaction\ j\ participates\ in\ current\ TIC\ solution \\ 0\ otherwise \end{cases}$$

$$\alpha_j = \begin{cases} 1\ if\ reaction\ j\ participates\ in\ current\ TIC\ solution\ with\ positive\ flux \\ 0\ otherwise \end{cases}$$

$$\beta_j = \begin{cases} 1\ if\ reaction\ j\ participates\ in\ current\ TIC\ solution\ with\ negative\ flux \\ 0\ otherwise \end{cases}$$

$\gamma_j \in [0,1] \equiv binary\ value\ which\ ensures\ that\ at\ least\ 1\ integer\ cut\ holds$

The set s_f is the set of all previously found TICs and represents the solution space that is known. It should be noted that set *J* is the set of all reactions in the database and model, of which set J^{DB}, the set of all reactions in the database, is a subset. Further, it should be noted that *I* is the set of all metabolites in the database and the model. Parameters (fixed values) and variables are defined after all constraints have been listed. The TIC-finding problem is run with all nutrient uptakes turned off, so that any reaction flux is unrealistic and due to one or more TICs. The TFP is included in File S3 as GAMS (Generalized Algebraic Modeling System) code. The following subsections will describe the above equations constituting the TFP in detail.

Objective function and sought TIC size

The solution of the TFP is itself a TIC. The objective function, equation (1), is minimization of the number of reactions participating in the TIC solution. This objective function is irrelevant in the solution due to equation (5), as equation (5) specifies the size of the TIC sought, and thus the objective function value, and is included to ensure that each possible TIC size is investigated. The order of solutions, when the workflow in Figure 2 is followed, is unimportant, and may vary each time the TFP is applied to a model.

Enforcing flux bounds and reaction participation

Equations (2) and (3) are constraints which enforce the given reaction flux bounds and determine if a reaction participates in the identified TIC. The variable η_j stores if a reaction participates in a TIC, while variables α_j and β_j store direction of participation. Reaction flux bounds v_j^{LB} and v_j^{UB} are determined manually based on reaction direction (reversible, irreversible forward, or irreversible backward), limitations on nutrient uptake rates, and reaction state (either on or off depending on genotype, nutrient availability). Equation (6) ensures that at least one database reaction holds flux. Equation (2) specifically identifies if reaction *j* participates in the solution TIC by requiring some small, minimum reaction flux, ϵ, for participating reactions such that equation (12) is true. Further, it identifies the direction of that reaction.

$$|v_j| \geq \epsilon \qquad \forall j \in \{J|\eta_j = 1\} \qquad (12)$$

Equation (3) ensures that if any reaction does not meet the reaction flux threshold to participate in the TIC solution, that the reaction flux is constraint to zero.

Identifying positive flux participation in the TIC

Equations (7) through (9) are a linearized version of the following statement.

$$\alpha_j \leq \eta_j(1 - \beta_j) \qquad (13)$$

The linearization in equations (7) through (9) functions the same as (13) because η_j and β_j are binary variables. This linearization is made in order to preserve the linear nature of the TFP. A linear optimization problem can guarantee both global solution optimality and that all solutions in the solution space can be enumerated, which in this case guarantees that all TICs are found of a given size.

Integer Cuts for Repeated Solutions

Equations (10) and (11) are integer cuts which prevent repetition of solutions. It should be noted that these repeated solutions include direction. Therefore, to be identified as the same TIC, the set of participating reactions and the directions in which they participate must be the same. Consider the following set of chemical equations for an illustration of how these integer cuts prevent repeated solutions.

$1\,A \leftrightarrow 1\,B$
$1\,B \leftrightarrow 1\,C$
$1\,C \leftrightarrow 1\,A$

The TFP, because of these integer cuts, would identify two TICs existing in this set of chemical reactions. The first would be all reactions listed above proceeding in the forward direction, while the second would be all reactions listed above proceeding in the backward direction. These are identified separately because their reaction directions are different, although the participating reactions are the same.

Modified TIC-Finding Problem (mTFP)

The TFP can be modified for the identification of TICs inherent to a metabolic model to aid in model curation. The modified TIC-Finding Problem (mTFP) can be formulated via equations (1) through (5) and equations (7) through (11). All set, parameter, and variable definitions are the same as in the unmodified TFP.

First Connecting Problem (CP1)

The connecting problems are the series of optimization problems which are solved following the solving of the TFP. First discussed will be the first Connecting Problem (CP1). The solution to a CP is a set of database reactions which, when added to the model, will increase model connectivity. The solution to CP1 gives the maximum number of model metabolites which could be connected using the database. The formulation of CP1 is given below.

$$maximize\ Z_{met} = \sum_{i_m \in I^M} x_{i_m} \qquad (14)$$

Subject to

$$\sum_{j_{db} \in J^{DB}} \zeta_{j_{db}} \geq 1 \qquad (15)$$

$$\rho_{j_{db}} v_{j_{db}}^{LB} \leq v_{j_{db}} \leq \delta_{j_{db}} v_{j_{db}}^{UB} \qquad \forall j_{db} \in J^{DB} \qquad (16)$$

$$\theta_j v_j^{LB} \leq v_j \leq (1 - \theta_j) v_j^{UB} - \epsilon\theta_j \qquad \forall j \in J \qquad (17)$$

$$(1 - \lambda_j) v_j^{LB} + \epsilon\lambda_j \leq v_j \leq \lambda_j v_j^{UB} \qquad (18)$$

$$x_i \le \sum_{j \in J} \left[\lambda_j \xi_{i,i} + \theta_j \psi_{i,j} \right] \qquad \forall i \in I \tag{19}$$

$$x_b = 1 \qquad \forall b \in B \subset I \tag{20}$$

$$\sum_{j \in J} S_{ij} v_j = 0 \qquad \forall i \in I \tag{21}$$

$$\zeta_{j_{db}} \le \sum_{i \in I} \left[\lambda_j \xi_{i,i} + \theta_j \psi_{i,j} \right] \qquad \forall j_{db} \in J^{DB} \tag{22}$$

$$\delta_{j_{db}} + \rho_{j_{db}} - \omega_{j_{db}} = \zeta_{j_{db}} \qquad \forall j_{db} \in J^{DB} \tag{23}$$

$$\omega_{j_{db}} \le \delta_{j_{db}} \qquad \forall j_{db} \in J^{DB} \tag{24}$$

$$\omega_{j_{db}} \le \rho_{j_{db}} \qquad \forall j_{db} \in J^{DB} \tag{25}$$

$$\sum_{j_{db} \in J^{DB}} \delta_{j_{db}} \left(\delta'_{s_c, j_{db}} \right) \le \sum_{j_{db} \in J^{DB}} \delta'_{s_c, j_{db}} - \sigma_{s_c} \qquad \forall s_c \in S_c \tag{26}$$

$$\sum_{j_{db} \in J^{DB}} \rho_{j_{db}} \left(\rho'_{s_c, j_{db}} \right) \le \sum_{j_{db} \in J^{DB}} \rho'_{s_c, j_{db}} - \left(1 - \sigma_{s_c} \right) \qquad \forall s_c \in S_c \tag{27}$$

$$\sum_{j_{db} \in J^{DB}} \left(\delta'_{s_c, j_{db}} - \delta_{j_{db}} \right) + \sum_{j_{db} \in J^{DB}} \left(\rho'_{s_c, j_{db}} - \rho_{j_{db}} \right) \ge \left(\sum_{j_{db} \in J^{DB}} \omega'_{s_c, j_{db}} \right) + 1 \qquad \forall s_c \in S_c \tag{28}$$

$$\sum_{j_{db} \in J^{DB}} \delta_{j_{db}} \left(\alpha'_{s_f, j_{db}} \right) \le \sum_{j_{db} \in J^{DB}} \alpha'_{s_f, j_{db}} - \tau_{s_f} \qquad \forall s_f \in S_f \tag{29}$$

$$\sum_{j_{db} \in J^{DB}} \rho_{j_{db}} \left(\beta'_{s_f, j_{db}} \right) \le \sum_{j_{db} \in J^{DB}} \beta'_{s_f, j} - \left(1 - \tau_{s_f} \right) \qquad \forall s_f \in S_f \tag{30}$$

Fixed Values Unique to CP1

$M = 1E3 \equiv$ *a very large number*

$$\delta'_{s_c, j_{db}} = \begin{cases} 1 \text{ if reaction } j_{db} \text{ is added in the forward direction from the} \\ \text{database in solution } s_c \\ 0 \text{ otherwise} \end{cases}$$

$$\rho'_{s_c, j_{db}} = \begin{cases} 1 \text{ if reaction } j_{db} \text{ is added in the backward direction from the} \\ \text{database in solution } s_c \\ 0 \text{ otherwise} \end{cases}$$

$$\omega'_{s_o, j_{db}} = \begin{cases} 1 \text{ if reaction } j_{db} \text{ is added in the forward direction from the} \\ \text{database in solution } s_o \\ 0 \text{ otherwise} \end{cases}$$

$$\xi_{i,j} = \begin{cases} 1 \text{ if metabolite } i \text{ is on the RHS of reaction } j \left(S_{i,j} > 0 \right) \\ 0 \text{ otherwise} \end{cases}$$

$$\psi_{i,j} = \begin{cases} 1 \text{ if metabolite } i \text{ is on the LHS of reaction } j \left(S_{i,j} < 0 \right) \\ 0 \text{ otherwise} \end{cases}$$

Variables Unique to CP1

$$\delta_{j_{db}} = \begin{cases} 1 \text{ if reaction } j_{db} \text{ is added in the forward direction from the database} \\ 0 \text{ otherwise} \end{cases}$$

$$\rho_{j_{db}} = \begin{cases} 1 \text{ if reaction } j_{db} \text{ is added in the backwards direction from the database} \\ 0 \text{ otherwise} \end{cases}$$

$$\omega_{j_{db}} = \begin{cases} 1 \text{ if reaction } j_{db} \text{ is added reversibly from the database } (\delta_{j_{db}} = \rho_{j_{db}} = 1) \\ 0 \text{ otherwise} \end{cases}$$

$$\zeta_{j_{db}} = \begin{cases} 1 \text{ if reaction } j_{db} \text{ is part of the solution} \\ 0 \text{ otherwise} \end{cases}$$

$$\theta_j = \begin{cases} 1 \text{ if reaction is proceding in backwards direction } \left(v_j < 0 \right) \\ 0 \text{ otherwise} \end{cases}$$

$$\lambda_j = \begin{cases} 1 \text{ if reaction is proceding in forwards direction } \left(v_j > 0 \right) \\ 0 \text{ otherwise} \end{cases}$$

$$x_i = \begin{cases} 1 \; if \; connected \; model \; produces \; metabolite \; i \\ \qquad \qquad \qquad 0 \; otherwise \end{cases}$$

$$\sigma_{s_c} \in [0,1] \equiv binary \; variable \; which \; ensures \; that \; the \; solution \; is \; unique \; from \; previous \; solutions \; in \; at \; least \; one \; direction \; of \; one \; database \; reaction$$

$$\tau_{s_c} \in [0,1] \equiv binary \; variable \; which \; ensures \; that \; the \; solution \; is \; free \; from \; TICs \; in \; that \; at \; least \; one \; direction \; of \; one \; database \; reactions \; which \; could \; cause \; a \; TIC \; is \; not \; added \; in \; the \; TIC - causing \; direction \; for \; each \; TIC \; identified \; by \; the \; TFP$$

Where I^M is defined as the set of metabolites in the model and is a subset of I. When CP1 is solved, the optimal value of Z_{met} is the maximum number of metabolites which can be connected in the model by adding reactions from the database, given all previous solutions (if any) and all identified potential TICs. It should be noted that all sets and parameters have the same definitions here as in the TFP, with the additions of J^M being the set of model reactions which is a subset of J, of I^M being the set of model metabolites which is a subset of I, s_c being the set of all previous connecting problem solutions, s_o being the set of all previous connecting problem solutions with at least one reversible reaction being added from the database, and B being the set of all metabolites which are involved in the biomass equation which is a subset of I.

The following statements give, broadly, the rational for each constraint equation. Equation (14) ensures that at least one reaction is added from the database for each solution. Equation (15) ensures that each database reaction only has flux if it is added. Equation (16) ensures that the user-defined reaction flux bounds hold. Equations (17) through (19) determine which metabolites the fixed model can produce, equation (19) ensures that the fixed model can produce biomass. Equation (20) ensures mass balance. Equation (21) ensures that added reactions are productive, e.g. that the added reaction does produce one or more metabolites. Equations (22) through (24) ensure that each database reaction for the connecting solution is added as a forward, backward, or reversible reaction (e.g. both as a forward and a backward reaction). Equations (25) though (28) are integer cuts preventing repeated solutions, while Equations (29) and (30) are integer cuts preventing the full addition of a TIC through the CP solution. The following subsections will describe some of the above equations constituting the CPs in greater detail. The CPs are included in File S4 as GAMS (Generalized Algebraic Modeling System) code. The following subsections will describe some of the above equations constituting the CPs in greater detail.

Determination of Metabolite Production

Important to CPs is the determination of whether or not a metabolite is produced in the connected model. Equations (17) and (18) are used to determine which direction reactions proceed in the connected model. Equation (19) essentially states that a metabolite is produced if at least one reaction produces that metabolite by having flux in the direction of that metabolite (either through backwards flux and a negative stoichiometric coefficient or forward flux and a positive stoichiometric coefficient). Equation (20) ensures that all metabolites necessary for growth (those involved in biomass production) are produced, as all models of metabolism should be capable of producing biomass, even if biomass is not ultimately the objective used. For instance, alternate objectives could include the maximization of production of a given metabolite (Herrgård, Fong, & Palsson, 2006)(Price, Reed, & Palsson, 2004), the minimization of the uptake of a particular substrate (Gomes de Oliveira Dal'Molin, Quek, Saa, & Nielsen, 2015), or minimization of metabolic adjustment (MOMA) (Herrgård et al., 2006)(Price et al., 2004). Ultimately, each objective type some fixed or variable non-zero level of biomass production and therefore all models require some ability to grow, making these constraints reasonable for reconstructions regardless of the ultimate objective used. Equation (22) ensures that reactions added from the database are productive, e.g. that each added reaction is capable of producing at least one metabolite. This constraint ensures that reactions incapable of carrying flux are not added to the model.

Direction of Added Database Reactions

Equations (22) through (25) largely deal with the direction in which reactions are added from the database. Equations (22) ensures that reactions added from the database are productive. Equation (23) ensures that ζ_{jdb} is equal to 1 if reaction j_{db} is added to the model as part of this solution, and zero otherwise. Equations (23) through (25) are the linearization of the multiplication of two binary variables stated below.

$$\omega_{j_{db}} = \delta_{j_{db}} \rho_{j_{db}} \tag{31}$$

This linearization is done for the same reasons that the TFP has been linearized. The sum of these constraints ensures that any reaction added reversibly is treated as a reaction added both forward and backwards for the purposes of integer cuts to avoid repeated solutions.

Integer Cuts for Repeated Solutions

Equations (26) through (28) define integer cuts used to avoid repeat solutions. Equations (26) and (27) have been designed on similar lines to (10) and (11), designed to avoid repeat solutions. Through the integer cuts in equations (26) through (28), both the reactions and their directions are integral to the solution; therefore, any different between solutions in reaction direction or reactions included is recorded as a second solution. Equation (28) prevents the repetition of a solution that could be caused by changing a reversible database reaction addition into an irreversible one.

Integer Cuts for TIC-less Connecting

Equations (29) through (30) define integer cuts which ensure that a TIC is not added to the connecting solution. This is done by considering both reaction identity and direction for both the addition of database reactions and for the avoidance of TICs. This results in a minimum perturbation to the solution space of CPs caused by each TIC. As with other directional integer cuts, only one cut needs be in effect at minimum in order to define a new solution.

Modified First Connecting Problem

A modified CP1 was used to get an initial count of the maximum number of metabolites which the raw model can produce. This modified CP1 made use of equations (14), and (16) through (30). In place of equation (15) the following equation was used to ensure that no database reactions were considered in maximizing the number of metabolites which may be connected.

$$\sum_{j_{db} \in J^{DB}} \zeta_{j_{db}} = 0 \qquad (32)$$

Second Connecting Problem

The second Connecting Problem (CP2) is defined as equations (15) through (30) with the addition of the objective function and constraint equation (34) stated below.

$$minimize\ Z_{rxn} = \sum_{j_{db} \in J^{DB}} \zeta_{j_{db}} \qquad (33)$$

s.t.

Equations (15) through (30)

$$\sum_{i_m \in I^M} x_{i_m} = Z_{met,opt} \qquad (34)$$

Where *Z_{met,opt}* is defined as the optimal objective value of CP1. When CP2 is solved, the optimal value of Z_{rxn} is the minimum number of reactions which, when added from the database, can connect the previously determined maximum number of model metabolites, given all previous solutions (if any) and all identified potential TICs.

Third Connecting Problem

The third Connecting Problem (CP3) is defined as equations (15) through (30), equation (34), and constraint equation (36) stated below.

$$maximize\ Z_{rev} = \sum_{j_{db} \in J^{DB}} \omega_{j_{db}} \qquad (35)$$

s.t.

Equations (15) through (31), (34)

$$\sum_{j_{db} \in J^{DB}} \zeta_{j_{db}} = Z_{rxn,opt} \tag{36}$$

Where $Z_{rxn,opt}$ is defined as the optimal objective value of CP2. When CP3 is solved, the optimal value of Z_{rev} is the maximum number of reversible reactions which can be used to achieve the minimum number of reaction additions to maximize model connectivity, given all previous solutions (if any) and all identified potential TICs. The solution of CP3 is the solution accepted as optimal.

CP3 has been found to be needed due to allowing database reactions to be added forward, backward, and reversibly. Since adding a reaction reversibly rather than irreversibly in some cases has made no difference, this resulted in an inconsistent number of solutions to the set of CPs. Therefore, in one run two solutions would be returned (the irreversible solution has been returned, then the reversible), where in a subsequent run perhaps only one solution would be returned if the reversible solution has been returned first, and then integer cuts (26) and (27) would preclude the irreversible solution. This third connecting problem has been added to deal with such situations by forcing the reversible solution to be returned first, resulting in a standardized, minimized set of solutions.

FBA of Connected Model

Once the CPs have been solved and the identity and direction of models to be added from the database to the model for a given solution are known, Flux Balance Analysis (FBA) is performed on the connected model. As the models are not physically merged, this takes the following form.

$$maximize\ Z_{bio} = v_{biomass} \tag{37}$$

s.t.

$$\sum_{j \in J} S_{ij} v_j = 0 \qquad \forall i \in I \tag{38}$$

$$v_{j_m}^{LB} \leq v_{j_m} \leq v_{j_m}^{UB} \qquad \forall j_m \in J^M \tag{39}$$

$$\rho'_{j_{db},s_{curr}} v_{j_{db}}^{LB} \leq v_{j_{db}} \leq \delta'_{j_{db},s_{curr}} v_{j_{db}}^{UB} \qquad \forall j_{db} \in J^{DB} \tag{40}$$

All variables, parameters, and sets are the same as in previous equations, and in addition s_{curr} represents the current connecting solution. In the above formulation, equation (39) takes into account the current solution of the CPs. A biomass maximization objective function was chosen for this work, but other objective could be selected depending on what part of metabolism is of most interest.

Creation of Test Models and Databases

Test model have been created in tandem with their databases using KEGG maps of pathways to identify sets of reactions which might produce a functional metabolic model. The first Test Model (TM1) and Test Database (TDb1) have been built from the “starch and sucrose metabolism” (map00500) and the “glycolysis/gluconeogenesis” (map00010) metabolic maps with the goal of producing a minimal prokaryotic model which utilizes sucrose, produces ethanol and biomass, and has some TICs which exist between the database and model where TM1 cannot produce biomass (without some TDb1 reactions) and contains no inherent TICs. Since only sucrose metabolism and glycolysis have been included in this model, biomass for this model is based on glucose, fructose, and an arbitrary growth-associated maintenance (GAM) value of 2. The coefficient of glucose in the biomass equation has then been scaled such that the molecular weight of biomass is 1000 g/mol. Non-Growth Associated Maintenance (NGAM) has also been defined arbitrarily as 2. TM1 and TDb1 have been constructed rationally with as many reversible reactions as possible, such that 22 of the 28 reactions are reversible in TM1 and all 17 reactions are reversible in TDb1. Once TM1 and TDb1 have been constructed, OptFill has been applied to them. This has resulted in the identification of 31 TICs consisting of 3 to 12 reactions by the TFP using the CPLEX solver. See results section for detail.

The first solution reported by OptFill for TM1/TDb1 has been added to TM1 to create the initial second Test Model (TM2). Added manually to this initial TM2 model is portions of the “pentose phosphate” pathway (map00030) and fatty acid biosynthesis” (map00061) pathway. The biomass equation has been updated to include a small amount (stoichiometric coefficient 0.01) of fatty acid products (8-, 10-, 12-, 14-, 16-, and 18-carbon fatty acid products) and the coefficient of glucose has again been adjusted to ensure biomass molecular weight was 1000 g/mol. Certain reactions in both pathways have been selected to constitute the second Test Database (TDb2), again with the aim of being a small prokaryotic model which utilizes sucrose, produces ethanol, produces biomass, and has some TICs which exist between the database and model where TM2 cannot produce biomass (without some TDb2 reactions) and contains no inherent TICs. In total, TM2 consists of 77 reactions (with 65 being reversible), and TDd2 consists of 34 reactions (all reversible). Once TM2 and TDb2 have been constructed, OptFill has been applied to them, see results section for details.

As with the construction of TM2, the third Test Model (TM3) has initially been constructed from the first solution of OptFill applied to TM2/TDb2 added to a test model. This test model has then been expanded to include “nitrogen metabolism” (map00910, with ammonium uptake), “sulfur metabolism” (map00920, with sulfate uptake), and synthesis pathways for all 20 amino acids. The biomass equation has been updated to include a small amount (stoichiometric coefficient 0.1) of each of the 20 primary amino acids, following which the coefficient of glucose has again been adjusted to ensure biomass molecular weight was 1000 g/mol. Unlike previous test models, this working test model (e.g. capable of producing biomass) with some TICs has first been developed, split between reactions belonging to TM2 or OptFill solution thereof, and “other” reactions. Then each of these “other” reaction has been assigned a random value (between 0 and 1) and those with a value greater than or equal to 0.7 have been assigned to the third Test Database (TDb3), and those with a value less than or equal to 0.8 have been assigned to the third Test Model (TM3). The code to perform this is included as part of the GitHub OptFill (10.52.81/zenodo.8475) or Mendeley Data (10.17632/npdwbmb7d7.1) repositories accompanying this work. Following this, the mTFP has been applied to TM3 in order to ensure that the model is TIC-less. For removing TICs from TM3, the number of occurrences of each reaction participating in all TICs has been counted, that has the highest occurrence, excluding those reactions from TM2 and TDb2, has been moved to TDb3. In the case of ties, the reaction with the highest reaction ID number has been moved to TDb3. In total, TM3 consists of 210 reactions (196 reversible), and TDb3 consists of 77 reactions (all reversible). Once TM3 and TDb3 have been constructed, OptFill has been applied to them, see results section for details.

It should be noted that for all instances of OptFill applied to test models some low number of execution errors have been allowed, five are allowed in this example option allowing execution errors: “execerr=5”. This has been done because GAMS throws an execution error if the RHS and LHS of a constraint are fixed and those fixed values do not satisfy the constraint. In the case of OptFill, this is not necessarily an issue, as it simply indicates that there are no more feasible solutions and that the program should continue onto the next problem or step. Graphical summaries comparing project runtimes have then been generated in Table S2. Result summaries, graphs and biomass calculations related to Figure 4 (Microsoft Excel) to produce Figure 4. Trend line and Pearson correlation values included in this figure have been generated automatically by Microsoft Excel. Linear, logarithmic, exponential, and power trend lines have been investigated, and the best fit line is displayed for each dataset. Polynomial trend lines have not been investigated as these trend lines can lead to overfitting errors.

Application of *i*AF1260 to *i*JR904

In the application of OptFill to published *Escherichia coli* GSMs, *i*JR904 (Reed, Vo, Schilling, & Palsson, 2003) was treated as the model and *i*AF1260 (Feist et al., 2007) as the source of reactions to build the database for OptFill. Minor formatting of both of these models was accomplished using the code in the GitHub OptFill or Mendeley Data repositories accompanying this work. Such formatting changes include changing of how reaction arrows appeared and location of metabolite compartment notation. Following this formatting, all exchange reactions were removed from *i*AF1260, as it was decided to use the media definition provided for *i*JR904 by Reed *et al.*, 2003, specifically for the case of aerobic growth on acetate. Whereas very large bounds in *i*JR904 have been defined as 1e^{30}, these have been redefined as 1e^{3} as both quantities are sufficiently large in the context of GSMs to be a red flag should any reaction flux reach that quantity. Further, 1e^{3} is the value of *M* used elsewhere in the code, resulting in a standard value for a “very large number”.

Once the aforementioned changes had been made, *i*AF1260 (sans exchange reactions) and *i*JR904 were compared in Table the GitHub OptFill or Mendeley Data repositories accompanying this work so that reactions that are in both model would be removed from *i*AF1260. These modifications resulted in 1441 reactions remaining in the initial *i*AF1260-based database. The initial iAF1260-based database is provided in the GitHub OptFill or Mendeley Data repositories accompanying this work, as is the GAMS code used in this application of OptFill. The OptFilling of *i*JR904 using an *i*AF1260-based database is different from the code used for the test models/database only in formatting of the output file (identifiers used were considerably longer than KEGG identifiers causing formatting issues). This was allowed for seven days to attempt to solve, in which time it did not return a single CPs solution; therefore, it was decided that the database needed to be made smaller. Both the initial *i*AF1260-based database and *i*JR904 were combined into a single pseudo-model file, to which Flux Variability Analysis (FVA) was applied. Those reactions which hold flux, 715 reactions, formed the second iAF1260-based database.

OptFill was applied to this second database, but still resulted in very long solution times; therefore, those reactions which participated in solutions which were achieved in 24 hours (four solutions) were chosen to form the final *i*AF1260-based database. This database consists of 182 reactions. It was found that this resulted in no TFP solutions; therefore, six more reactions were added to produce a database which had 25 potential TICs with the *i*JR904 model. OptFill was then applied to *i*JR904 using this final *i*AF1260-based database of 188 reactions.

CPLEX Solver Options

As the order of solutions presented is important, solver options which allowed non-optimal solutions or created relaxations by which the truly optimal solution could not be reached, or a sub-optimal solution would be accepted, were disabled. In particular, the infeasibility gap was set to the lowest possible value, small infeasibilities were disallowed, no relaxation was allowed in the value of integers, no optimality gap was allowed in the solution, and solver cuts which could result in non-optimal solutions were disabled. These cuts included zero-half, flow, clique, cover, mixed integer rounding, GUB cover, and Gomory fractional cuts. While the lack of these relaxation options and cuts no doubt increased solution time, these relaxations would decrease solution accuracy and order which was deemed unacceptable. The list of CPLEX relaxations used in this work can be found in the GitHub OptFill or Mendeley Data repositories accompanying this work.

Available Hard- and Soft-ware Tools

The University of Nebraska-Lincoln Holland Computing Center Crane Cluster was used. The nodes used on this cluster for this work have 64 GB RAM, Intel Xenon E5-2670 2.60 GHz processors, and 2 CPUs per 16 nodes for 548 nodes. In addition, Crane uses an old version of GAMS, version 24.7.4 released in March 2016 as opposed to the current version 28.1.0 released August 2019; therefore, algorithms used in this work may be sub-optimal compared to those which may be available at present. Further, this version of GAMS corresponds to CPLEX library 12.6, which was release in January 2014. Therefore, solution time may be different, and significantly so, for users with access to different hard- and soft-ware tools. It should also be noted that the solution times are not static. That is, each time OptFill is run, it may have different values for solution times; however, the patterns of solution time and distributions should remain relatively consistent.

Acronyms Used

LHS – Left Hand Side
RHS – Right Hand Side
TFP – TIC-Finding Problem
CPs – Connecting Problems
FBA – Flux Balance Analysis
TM1 – First Test Model
TDb1 – First Test Database
TM2 – Second Test Model
TDb2 – First Test Database
TM3 – Third Test Model
TDb3 – First Test Database

GAM – Growth Associated Maintenance
NGAM – Non-Growth Associated Maintenance

QUANTIFICATION AND STATISTICAL ANALYSIS

The software used to calculate statistics presented in this work was GAMS; however, as GAMS has no built-in statistics tools, these calculations were performed in the code included in File the GitHub OptFill or Mendeley Data repositories accompanying this work. The two statistics calculated include mean and standard deviation. In this study, the arithmetic mean was used, and the standard deviation calculation used was that of a population (as opposed to a sample), as the full population of solutions was used in the test statistic.

SUPPLEMENTAL REFERENCES

All works referenced in the Transparent Methods are referenced in the main body of the this work.