SWIFTCORE: a tool for the context-specific reconstruction of genome-scale metabolic networks

Mojtaba Tefagh [*] [†]   Stephen P. Boyd [*]

March 3, 2020

Abstract

Background: High-throughput omics technologies have enabled the comprehensive reconstructions of genome-scale metabolic networks for many organisms. However, only a subset of reactions is active in each cell which differs from tissue to tissue or from patient to patient. Reconstructing a subnetwork of the generic metabolic network from a provided set of context-specific active reactions is a demanding computational task.

Results: We propose SWIFTCC and SWIFTCORE as effective methods for flux consistency checking and the context-specific reconstruction of genome-scale metabolic networks which consistently outperform the previous approaches.

Conclusions: We have derived an approximate greedy algorithm which efficiently scales to increasingly large metabolic networks. SWIFTCORE is freely available for non-commercial use in the GitHub repository at https://mtefagh.github.io/swiftcore/.

Keywords: systems biology; metabolic network analysis; metabolic network reconstruction, context-specific models.

Mathematics Subject Classification: 92C42; 90C05; 49N15

Background

Constraint-based reconstruction and analysis (COBRA) is the current state-of-the-art in the genome-scale metabolic network modelling [OMP15]. COBRA methods systematize biochemical constraints into a mathematical framework which synthetic biologists can utilize to quantitatively simulate metabolic pathways in order to answer the relevant biological questions. There is a critical mass of studies that combine these curated high-dimensional models and *in silico* analysis for drug discovery or many other applications [BMKP14, SOA$^+$16, FJF$^+$11a].

Context-specific metabolic networks are extensively studied because of their higher explanatory and predictive power [JSR10, FJF$^+$11b, BMN$^+$12, FM17]. To date, a wide variety of computational methods have been developed to extract context-specific metabolic networks from the available comprehensive genome-scale reconstructions. *Gene inactivity moderated by metabolism and expression* (GIMME) [BP08] uses quantitative gene expression data and presupposed cellular functions to predict the subset of reactions that a cell uses under particular conditions. *Integrative metabolic analysis tool* (iMAT) [SCH$^+$08, ZRS10] integrates tissue-specific gene- and protein-expression data to produce context-specific metabolic networks. *Integrative network inference for tissues* (INIT) [ABM$^+$12, ALS$^+$13, AMA$^+$14] uses cell type specific proteomic data from the *human proteome atlas* (HPA) to reconstruct tissue-specific metabolic networks. Metaproteomic and taxonomic data have been exploited for a context-specific reconstruction procedure applied to a naphthalene-degrading bacterial community [TBP$^+$15]. RegrEx [EN15, REN17] is based on regularized least

[*] Information Systems Laboratory, Department of Electrical Engineering, Stanford University, Stanford, CA

[†] Correspondence: mtefagh@stanford.edu

squares optimization using publicly available RNAseq expression profiles. *Context-specificity assessed by deterministic reaction evaluation* (mCADRE) [WEP12] is also based on gene expression data but evaluates the functional capabilities during model building too. Another approach is to infer the metabolic functionalities of a cell or tissue from transcriptomic data, and then protect these functions during the implementation of context-specific reconstruction [RCKL19]. *Cost optimization reaction dependency assessment* (CORDA) [SQ16] relies solely on *flux balance analysis* (FBA), rendering it more computationally efficient.

One approach to this problem is to curate a list of active reactions and develop a framework to find the sparsest subnetwork containing the specified reactions. Although this subnetwork is assumed to be as sparse as possible, we should avoid flux inconsistencies such as when one of the desired reactions is included in the subnetwork, yet it is blocked and cannot carry non-zero flux at the steady-state condition. In this regard, context-specific reconstructions seek to generate minimal consistent subnetworks including a given set of core reactions.

It is known that identifying a consistent subnetwork with the minimum possible size such that it contains some core reactions is an NP-hard problem [ACL+09]. To address this issue, approximate greedy algorithms have been developed which either prune the original network [JSR10] or increment the core set [VPS14] recursively to arrive at a consistent subnetwork in between them. Among these competing approaches, FASTCORE [VPS14] exhibits superior performance both in terms of the sparseness of the subnetwork and the computational efficiency.

Let $\mathcal{M} = \{M_i\}_{i=1}^m$ denote m specific metabolites in an organism, and $\mathcal{R} = \{R_i\}_{i=1}^n$ be the set of n reactions involving at least one of these metabolites. Under average physiological conditions, the irreversible reactions $\mathcal{I} \subseteq \mathcal{R}$ are thermodynamically constrained to proceed in the forward direction only, in contrast to the reversible reactions which may proceed in the reverse direction as well.

We call a vector v of length n a flux distribution if the absolute values of its entries are the rates of the corresponding reactions in $\mathcal{R}$, and the signs of its entries indicate the forward and reverse directions. Unless stated otherwise, all flux distributions are assumed to respect the irreversibility conditions in the sense that $v_i \geq 0$ for all $R_i \in \mathcal{I}$.

We represent the relative quantities of metabolites in each reaction by an associated vector of length m and distinguish reactants from products by negative signs. Afterward, we construct the stoichiometric matrix by stacking these vectors for all the reactions in $\mathcal{R}$ as the columns of an $m \times n$ matrix S. The mass balance constraint asserts that the concentration of each metabolite is constant throughout the time-scale of interest which is equivalent to say that $Sv = 0$ in our notation.

We refer to any solution of $Sv = 0$ such that $v_i \geq 0$ for all $R_i \in \mathcal{I}$ as a steady-state flux distribution. We call $R_i \in \mathcal{R}$ a blocked reaction if $v_i = 0$ for all the steady-state flux distributions, and unblocked otherwise. We call a metabolic network with no blocked reactions a flux consistent metabolic network [SH94].

In this paper, we present an algorithm that given a flux consistent metabolic network and the subset $\mathcal{C} \subset \mathcal{R}$ of core reactions as input, computes a flux consistent subnetwork $\mathcal{N} \subseteq \mathcal{R}$ such that $\mathcal{C} \subseteq \mathcal{N}$ as output. Ideally, we are interested to find a sparse subnetwork, and accordingly, we search to minimize the size of $\mathcal{N}$.

Related works

A closely related problem to the context-specific reconstruction is to check the flux consistency of a given metabolic network by detecting the blocked reactions. FASTCC [VPS14] which is based on the same ideas as used for FASTCORE, is currently the fastest algorithm dedicated to this task.

As a simple observation, all the irreversible reactions in $\mathcal{I}$ are unblocked if and only if we can find a flux distribution v such that

$$\begin{aligned} Sv &= 0 \\ v_{\mathcal{I}} &> 0, \end{aligned} \tag{1}$$

where $v_{\mathcal{I}} > 0$ is the shorthand of $v_i > 0$ for all $R_i \in \mathcal{I}$. Assuming that such a flux distribution v exists, an arbitrary (possibly reversible) $R_j \in \mathcal{R}$ is unblocked if and only if there exists u such that

$$\begin{aligned} Su &= 0 \\ u_j &\neq 0, \end{aligned} \tag{2}$$

since if $c > 0$ is large enough, then $u_j + cv_j \neq 0$ and $u + cv$ would be a steady-state flux distribution in which R_j is active.

Quantitative flux coupling analysis (QFCA) [TB19] uses this observation to develop a consistency checking technique which we call SWIFTCC in this paper. However, this is presented as only a preprocessing step in the original paper instead of a separate algorithm. For the sake of completeness, we have also compared SWIFTCC as implemented in the QFCA against FASTCC. Additionally, we have benchmarked FASTCC++ and SWIFTCC++ which are the original algorithms plus the preprocessing step explained in the Appendix. Later on, we extend the ideas of SWIFTCC to develop an analogous method for the context-specific reconstruction problem with the same order of speed-up which SWIFTCC offers.

Methods

By similar arguments to the previous section, we concluded that a subnetwork $\mathcal{N}$ is flux consistent if and only if

I. there exists v in analogy to (1) such that

$$\begin{aligned} &Sv = 0 \\ &v_{\mathcal{I}\cap\mathcal{N}} > 0 \\ &v_{\mathcal{R}\setminus\mathcal{N}} = 0, \end{aligned}$$

II. and there exists a set $\{u^k\}_{k=1}^{K}$ in analogy to (2) such that

$$\begin{aligned} &Su^k = 0 \\ &u^k_{\mathcal{R}\setminus\mathcal{N}} = 0, \end{aligned} \tag{3}$$

holds for all $1 \leq k \leq K$. Furthermore, for any $R_j \in \mathcal{N} \setminus \mathcal{I}$, $u^k_j \neq 0$ holds for at least one $1 \leq k \leq K$.

We start by finding a v which is sparse in $\mathcal{R} \setminus \mathcal{C}$ by minimizing the l_1 norm [Don06]

$$\begin{array}{ll} \text{minimize} & \left\|v_{\mathcal{R}\setminus\mathcal{C}}\right\|_1 \\ \text{subject to} & Sv = 0 \\ & v_{\mathcal{I}\cap\mathcal{C}} > 0 \\ & v_{\mathcal{I}\setminus\mathcal{C}} \geq 0, \end{array}$$

and setting the initial $\mathcal{N}$ to be the non-zero indices of v. Consequently, the optimal v satisfies I for this $\mathcal{N}$.

This homogeneous problem is equivalent to the following *linear program* (LP)

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T w \\ \text{subject to} & Sv = 0 \\ & v_{\mathcal{I}\cap\mathcal{C}} \geq \mathbf{1} \\ & v_{\mathcal{I}\setminus\mathcal{C}} \geq 0 \\ & w \geq v_{\mathcal{R}\setminus\mathcal{C}} \\ & w \geq -v_{\mathcal{R}\setminus\mathcal{C}}, \end{array} \tag{4}$$

by scaling v if necessary, so that the nonzero entries of $v_{\mathcal{I}\cap\mathcal{C}}$ are greater than or equal to one.

We define the set $\mathcal{B} = \mathcal{N} \setminus \mathcal{I}$ to be the reactions in $\mathcal{N}$ which have not been verified to be unblocked yet. Whenever we find a u_k which satisfies the conditions of (3), we update $\mathcal{B}$ by removing the reactions R_j for which $u^k_j \neq 0$, and we also update $\mathcal{N}$ by adding the same reactions to it if they are not already included.

Let $S_{\mathcal{N}}$ denote the matrix consisting of the columns of S which correspond to $\mathcal{N}$. We initialize $\{u^k\}_{k=1}^{K}$ to be a basis for the set of vectors satisfying (3). Note that, this can be obtained by the *singular value decomposition* (SVD) of $S_{\mathcal{N}}$ since it is a basis for the null space of $S_{\mathcal{N}}$ padded by zeros for the indices corresponding to $\mathcal{R} \setminus \mathcal{N}$. This step is not required and can be skipped to decrease the runtime.

In order to generate the next u^k, we consider the solution of

$$\begin{array}{ll} \text{minimize} & x^T u_{\mathcal{B}} + \left\| u_{\mathcal{R} \setminus \mathcal{N}} \right\|_1 \\ \text{subject to} & Su = 0 \\ & \left\| u_{\mathcal{B}} \right\|_\infty \leq \mathbf{1}, \end{array}$$

where x is a random vector generated from the zero-mean normal distribution with variance σ^2. This problem tries to find a u which is sparse in $\mathcal{R} \setminus \mathcal{N}$ by minimizing the l_1 norm, and which is dense in $\mathcal{B}$ by the l_∞ norm constraint [BV04]. The choice of the variance σ^2 manages the trade-off between these two objectives and is set by a simple rule which doubles σ whenever $u_{\mathcal{B}}$ is not dense enough, for instance whenever the size of $\mathcal{B}$ is not reduced by more than half. In addition, we have tried several other heuristics, however, SWIFTCORE is robust across a wide range of σ.

Finally, this problem is equivalent to the following LP

$$\begin{array}{ll} \text{minimize} & x^T u_{\mathcal{B}} + \frac{1}{\sigma} \mathbf{1}^T w \\ \text{subject to} & Su = 0 \\ & u_{\mathcal{B}} \leq \mathbf{1} \\ & -u_{\mathcal{B}} \leq \mathbf{1} \\ & u_{\mathcal{R} \setminus \mathcal{N}} \leq w \\ & -u_{\mathcal{R} \setminus \mathcal{N}} \leq w, \end{array} \tag{5}$$

where x is sampled from the standard normal distribution. More generally, $\frac{1}{\sigma}\mathbf{1}$ can be substituted by any positive weight vector ω to customize the loss corresponding to each reaction. In the current work, we have only experimented with $\omega = \frac{1}{\sigma}\mathbf{1}$ for different values of σ. However, the accompanying package supports any positive weight vector ω.

Subsequently, we keep iterating over k until no reaction remains in $\mathcal{B}$. Therefore, we will eventually arrive at a set $\{u^k\}_{k=1}^K$ which satisfies II for the final $\mathcal{N}$, thus together with the prior v, they imply that $\mathcal{N}$ is a flux consistent subnetwork. Note that, we can also impose lower and upper bound constraints on the feasible flux distributions by adding the corresponding inequalities to (4) and (5) for the general case.

Altogether, this algorithm is based on linear programming, and hence, SWIFTCORE is ultra-fast in comparison to nonlinear methods such as the *model-building algorithm* (MBA) [JSR10] which are orders of magnitude slower. Focusing on computational efficiency, the only real contender is the FASTCORE algorithm which is again based on linear programming but is still several times slower, mainly due to flipping the signs of the columns of the stoichiometric matrix that correspond to the reversible reactions. We refer the interested reader to the associated paper [VPS14] for more details on this technique which we have avoided in SWIFTCORE in the way explained below.

Whether it is a binary variable in a *mixed-integer linear program* (MILP) corresponding to the direction of a reversible reaction or two separate iterations over the forward and reverse directions like in the FASTCORE, a common way of dealing with reversible reactions is to iterate over instances of the metabolic network with a predetermined direction for that reversible reactions. Instead of determining a direction for the reversible reactions, signs of the different entries of x merely encourage one of the two possible directions in a "soft" manner instead of a "hard" constraint. The objective function of (5) is minimized when most of the entries of $u_{\mathcal{B}}$ have the opposite sign to the corresponding entries of x. However, it may not be the case for some entries of the solution, especially entries with small absolute values. In this sense, even though the sampled x assigns random preferences to the direction of every reversible reaction in $\mathcal{B}$, the optimal u might have different signs in a few entries. As a result, the search for a sparse u is conducted over a much larger feasible set, and $\mathcal{B}$ vanishes in fewer iterations.

As a side note, SWIFTCORE also preprocesses the original metabolic network by merging the pairs of reactions which are fully coupled to each other by a metabolite with no other adjacent reaction (*cf.*, Figure 1 in [LDSB12]). However, this optional subroutine can be skipped safely and only affects the runtime. Besides, we can repeat the whole algorithm with the same core reactions $\mathcal{C}$ but replace the original metabolic network by $\mathcal{N}$ to further shrink the subnetwork until its size is no longer reduced.

Implementation

We have implemented both SWIFTCC and SWIFTCORE in an open-source package freely available for academic use on GitHub. It is written in MATLAB® with zero dependencies, though it supports Gurobi™ or CPLEX™ optimizer for improved performance. Additionally, if any other LP solver has been set up for the COBRA toolbox v3.0 [HAP+17], it can use that as well by calling the LP solve function of COBRA.

The partitioning preprocessing step described in the Appendix is also included in this package. Despite the fact that in our simulations it does not improve the performance directly, we suggest exploiting its potential to develop parallel consistency checking methods as the partitioned subnetworks can be analyzed independently. We did not investigate how much this might improve the efficiency of either SWIFTCC or FASTCC none of which can readily be parallelized in an obvious way and this direction is left for future research.

Last but not least, the benchmark codes to reproduce all the figures in this article are also publicly available in the same GitHub repository.

Results and discussion

To assess the performance of SWIFTCORE, we have benchmarked both SWIFTCORE and FASTCORE on the flux consistent part of the Recon3D model [BSZ+18] with randomly selected core sets of varying sizes over the range of 1 to the number of reactions $n = 10600$. All simulations were performed on a desktop PC with AMD Ryzen™7 1800X eight-core processor and 16 GB of memory, and the CPLEX™ optimizer was employed as the LP solver. Furthermore, in the published benchmark code we double check if the reconstructed subnetwork is flux consistent and contains the core reactions and SWIFTCORE always passes these sanity checks.

In Figure 1, the difference between the two versions of the algorithm is that in the SWIFTCORE with reduction version we have included the SVD and full coupling reduction techniques described before in contrast to the other vanilla version which excludes them. In spite of the fact that the vanilla version solves more LP problems, it is actually more efficient nonetheless (see Figure 2).

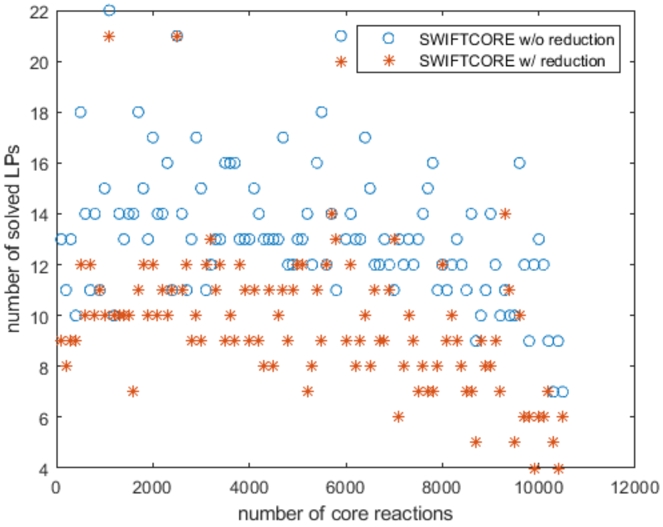

Figure 1: SWIFTCORE requires to solve at most 22 LPs on Recon3D with $n = 10600$

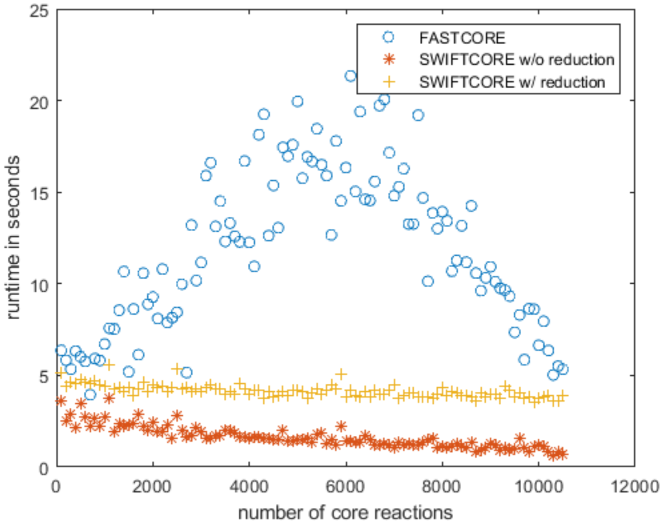

Figure 2: SWIFTCORE is 6× faster than FASTCORE on average over these 105 iterations of varying sizes

Figure 3 shows that the sparsity of the outputs of all three algorithms is nearly identical. Moreover, the intersection of the three solution subnetworks usually accounts for about 95% of the reactions which means the subnetworks are almost the same.

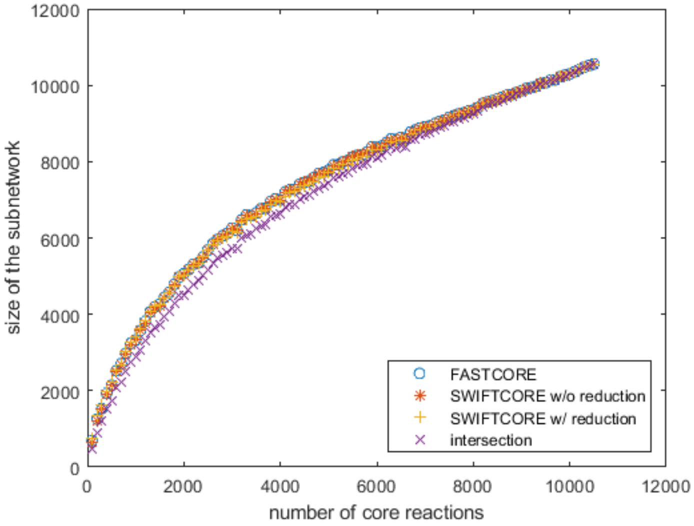

Figure 3: Benchmark of SWIFTCORE against FASTCORE on Recon3D

Regarding the impact of the weights, Figures 4, 5, and 6 demonstrate that the number of solved LPs, the runtime, and the sparsity of the output subnetworks is pretty robust with respect to the variations of weights. In these figures, we randomly sample half of the reactions in the network and run both algorithms with the weight vectors $2^k\mathbf{1}$ for $k = 0, 1, \ldots, 15$. Note that the magnitude of the largest weight vector is 32768 times the magnitude of the smallest one yet again, as claimed before, SWIFTCORE is robust to the choice of σ.

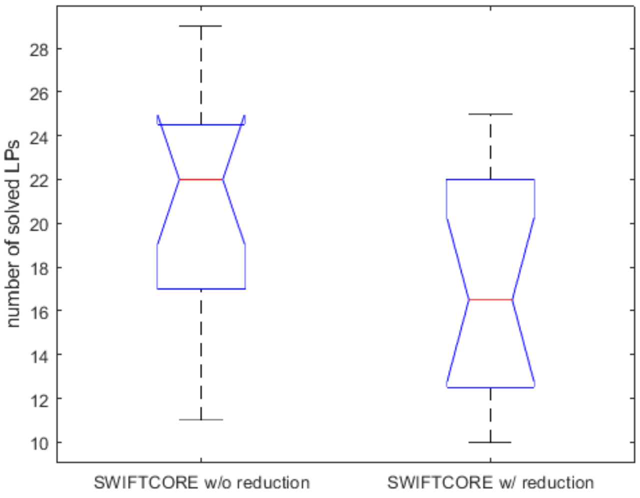

Figure 4: The number of solved LPs changes proportional to the logarithm of the scaling factor

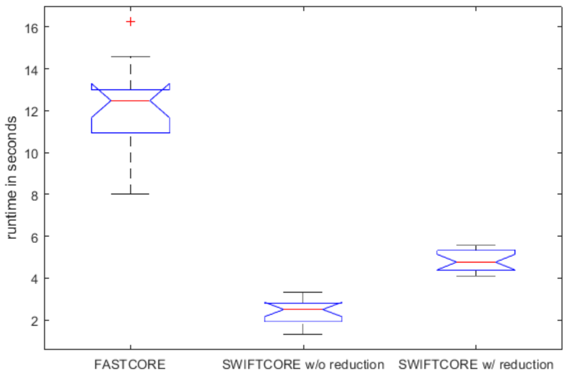

Figure 5: The variation in runtime is relatively smaller for SWIFTCORE compared to FASTCORE

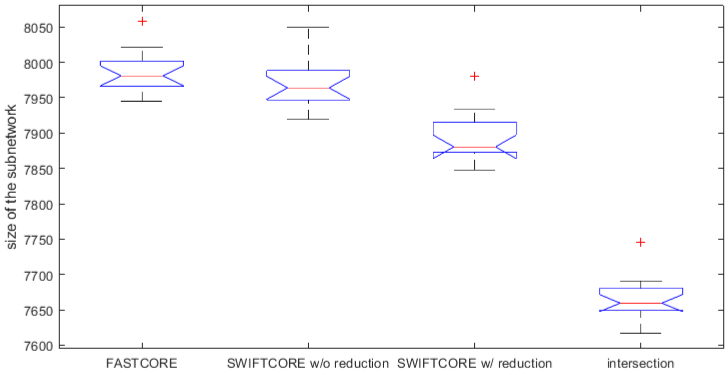

Figure 6: The difference in sparsity is less than 1% for all algorithms

In the end, we have also benchmarked SWIFTCC against FASTCC on the flux inconsistent version of the Recon3D model and its randomly selected subnetworks. In Figures 7 and 8, SWIFTCC's runtime is 8% of FASTCC's runtime on Recon3D model averaged over sampled subnetworks of different sizes from 1 to $n = 13543$.

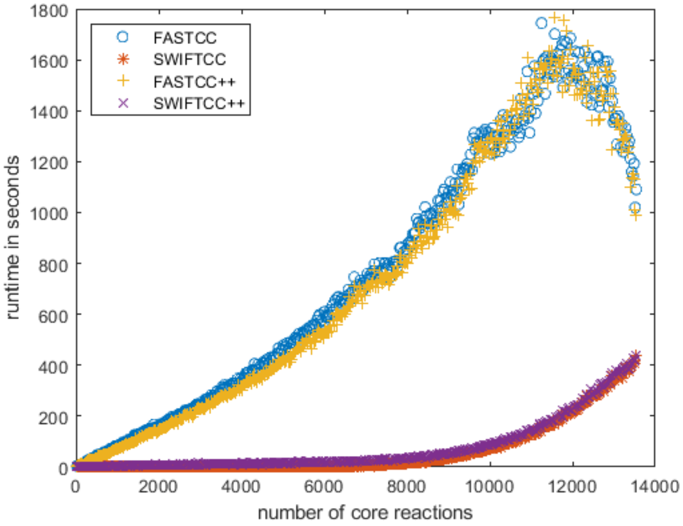

Figure 7: SWIFTCC is more than 12× faster than FASTCC on average over these 467 iterations of varying sizes

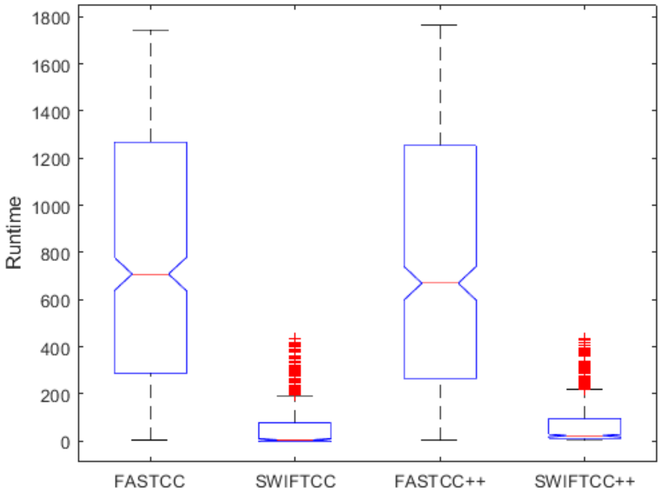

Figure 8: FASTCC++ is only 4% faster than FASTCC on average

In a second setting, we evaluated both algorithms by the reconstruction of a liver model previously studied in the original MBA and FASTCORE articles [JSR10, VPS14]. All simulations were performed on a laptop with Intel© Core™i7-5500U CPU @ 2.40GHz × 2 and 16 GB of memory, and the Gurobi™ optimizer was employed as the LP solver. Figures 9 and 10 are consistent with our previous findings.

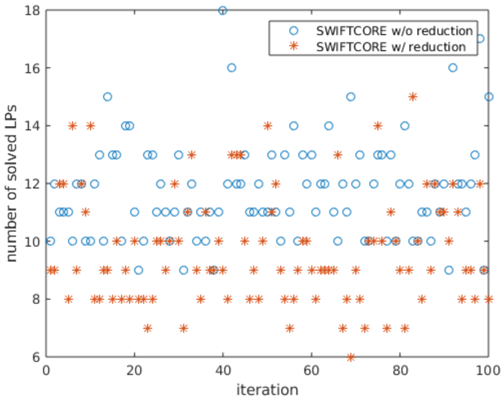

Figure 9: SWIFTCORE requires to solve at most 18 LPs for the hepatocyte-specific reconstruction

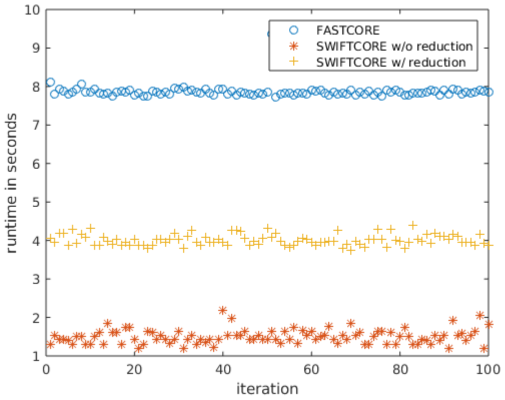

Figure 10: SWIFTCORE is more than 5× faster than FASTCORE on average over these 100 iterations

Since FASTCORE is a deterministic algorithm, we see a corresponding flat line in Figure 11 because of the same hepatocyte-specific core set used in every iteration. On the other hand, the randomly selected x in (5) makes SWIFTCORE a stochastic algorithm which samples the space of alternative optimal subnetworks. It has been proposed that a careful balance between model sparsity and metabolic functionality helps in reducing the ambiguity of context-specific metabolic network predictions [REN16, REN17]. In order to do so, we have tested all reconstructions by the list of data-inferred metabolic tasks recently published in [RCKL19]. As we can see in Figure 12, both versions of the SWIFTCORE algorithm sometimes pass an additional task which can be a guidelines to obtain a more biologically relevant reconstruction from the sampled alternative optima.

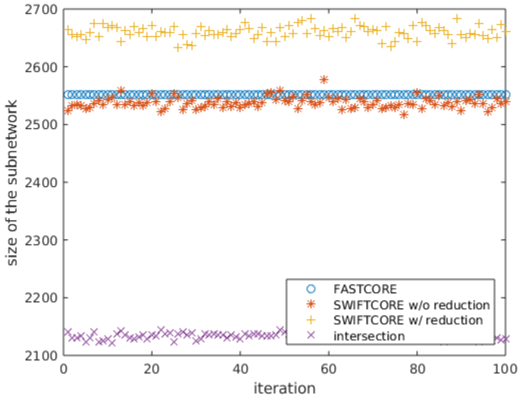

Figure 11: SWIFTCORE with reduction is 4% less sparse and without reduction is 1% sparser than FASTCORE

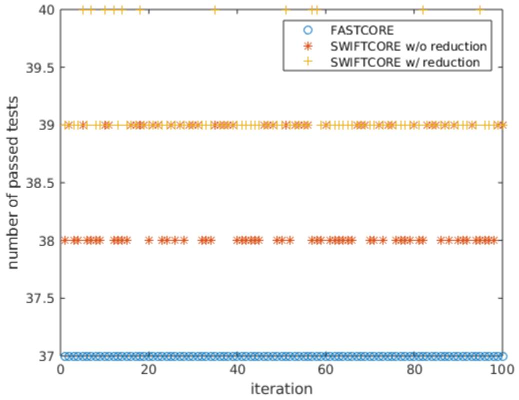

Figure 12: SWIFTCORE with reduction passes 2.12 and without reduction passes 1.44 more tasks than FASTCORE on average

Conclusions

Here, we have presented SWIFTCC and SWIFTCORE and one can easily implement a corresponding SWIFTGAPFILL by substituting the subroutine calls to FASTCC and FASTCORE in the FASTGAPFILL [TVF14]. We have demonstrated through a real-world genome-scale model that the SWIFTCORE family improves the efficiency of metabolic network reconstruction significantly.

References

[ABM$^+$12] Rasmus Agren, Sergio Bordel, Adil Mardinoglu, Natapol Pornputtapong, Intawat Nookaew, and Jens Nielsen. Reconstruction of genome-scale active metabolic networks for 69 human cell types and 16 cancer types using INIT. *PLoS Computational Biology*, 8(5):e1002518, 2012.

[ACL$^+$09] Vicente Acuna, Flavio Chierichetti, Vincent Lacroix, Alberto Marchetti-Spaccamela, Marie-France Sagot, and Leen Stougie. Modes and cuts in metabolic networks: Complexity and algorithms. *Biosystems*, 95(1):51–60, 2009.

[ALS$^+$13] Rasmus Agren, Liming Liu, Saeed Shoaie, Wanwipa Vongsangnak, Intawat Nookaew, and Jens Nielsen. The RAVEN toolbox and its use for generating a genome-scale metabolic model for Penicillium chrysogenum. *PLoS computational biology*, 9(3):e1002980, 2013.

[AMA$^+$14] Rasmus Agren, Adil Mardinoglu, Anna Asplund, Caroline Kampf, Mathias Uhlen, and Jens Nielsen. Identification of anticancer drugs for hepatocellular carcinoma through personalized genome-scale metabolic modeling. *Molecular Systems Biology*, 10(3), 2014.

[BMKP14] Aarash Bordbar, Jonathan M Monk, Zachary A King, and Bernhard Ø Palsson. Constraint-based models predict metabolic and associated cellular functions. *Nature Reviews Genetics*, 15(2):107–120, 2014.

[BMN$^+$12] Aarash Bordbar, Monica L Mo, Ernesto S Nakayasu, Alexandra C Schrimpe-Rutledge, Young-Mo Kim, Thomas O Metz, Marcus B Jones, Bryan C Frank, Richard D Smith, Scott N Peterson, et al. Model-driven multi-omic data analysis elucidates metabolic immunomodulators of macrophage activation. *Molecular systems biology*, 8(1):558, 2012.

[BP08] Scott A Becker and Bernhard Ø Palsson. Context-specific metabolic networks are consistent with experiments. *PLoS Computational Biology*, 4(5):e1000082, 2008.

[BSZ$^+$18] Elizabeth Brunk, Swagatika Sahoo, Daniel C Zielinski, Ali Altunkaya, Andreas Dräger, Nathan Mih, Francesco Gatto, Avlant Nilsson, German Andres Preciat Gonzalez, Maike Kathrin Aurich, et al. Recon3D enables a three-dimensional view of gene variation in human metabolism. *Nature Biotechnology*, 36(3):272, 2018.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization.* Cambridge University Press, New York, NY, USA, 2004.

[Don06] David L Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[EN15] Semidán Robaina Estévez and Zoran Nikoloski. Context-specific metabolic model extraction based on regularized least squares optimization. *PloS One*, 10(7):e0131875, 2015.

[FJF$^+$11a] Ori Folger, Livnat Jerby, Christian Frezza, Eyal Gottlieb, Eytan Ruppin, and Tomer Shlomi. Predicting selective drug targets in cancer through metabolic networks. *Molecular Systems Biology*, 7(1), 2011.

[FJF$^+$11b] Ori Folger, Livnat Jerby, Christian Frezza, Eyal Gottlieb, Eytan Ruppin, and Tomer Shlomi. Predicting selective drug targets in cancer through metabolic networks. *Molecular systems biology*, 7(1):501, 2011.

[FM17] Hamideh Fouladiha and Sayed-Amir Marashi. Biomedical applications of cell-and tissue-specific metabolic network models. *Journal of biomedical informatics*, 68:35–49, 2017.

[GPF08] Albert Gevorgyan, Mark G Poolman, and David A Fell. Detection of stoichiometric inconsistencies in biomolecular models. *Bioinformatics*, 24(19):2245–2251, 2008.

[HAP$^+$17] Laurent Heirendt, Sylvain Arreckx, Thomas Pfau, Sebastian N Mendoza, Anne Richelle, Almut Heinken, Hulda S Haraldsdottir, Sarah M Keating, Vanja Vlasov, Jacek Wachowiak, et al. Creation and analysis of biochemical constraint-based models: the COBRA toolbox v3.0. *arXiv preprint arXiv:1710.04038*, 2017.

[JSR10] Livnat Jerby, Tomer Shlomi, and Eytan Ruppin. Computational reconstruction of tissue-specific metabolic models: application to human liver metabolism. *Molecular Systems Biology*, 6(1):401, 2010.

[LDSB12] Abdelhalim Larhlimi, Laszlo David, Joachim Selbig, and Alexander Bockmayr. F2C2: a fast tool for the computation of flux coupling in genome-scale metabolic networks. *BMC Bioinformatics*, 13(1):57, 2012.

[OMP15] Edward J. O'Brien, Jonathan M. Monk, and Bernhard Ø Palsson. Using genome-scale models to predict biological capabilities. *Cell*, 161(5):971 – 987, 2015.

[RCKL19] Anne Richelle, Austin WT Chiang, Chih-Chung Kuo, and Nathan E Lewis. Increasing consensus of context-specific metabolic models by integrating data-inferred cell functions. *PLoS Computational Biology*, 15(4):e1006867, 2019.

[REN16] Semidán Robaina-Estévez and Zoran Nikoloski. On the effects of alternative optima in context-specific metabolic model predictions, 2016.

[REN17] Semidán Robaina-Estévez and Zoran Nikoloski. On the effects of alternative optima in context-specific metabolic model predictions. *PLoS Computational Biology*, 13(5):e1005568, 2017.

[SCH$^+$08] Tomer Shlomi, Moran N Cabili, Markus J Herrgård, Bernhard Ø Palsson, and Eytan Ruppin. Network-based prediction of human tissue-specific metabolism. *Nature biotechnology*, 26(9):1003, 2008.

[SH94] Stefan Schuster and Claus Hilgetag. On elementary flux modes in biochemical reaction systems at steady state. *Journal of Biological Systems*, 2(02):165–182, 1994.

[SOA$^+$16] Itay Shaked, Matthew A. Oberhardt, Nir Atias, Roded Sharan, and Eytan Ruppin. Metabolic network prediction of drug side effects. *Cell Systems*, 2(3):209 – 213, 2016.

[SQ16] André Schultz and Amina A Qutub. Reconstruction of tissue-specific metabolic networks using CORDA. *PLoS Computational Biology*, 12(3):e1004808, 2016.

[TB19] Mojtaba Tefagh and Stephen P Boyd. Quantitative flux coupling analysis. *Journal of mathematical biology*, 78(5):1459–1484, 2019.

[TBP$^+$15] Luis Tobalina, Rafael Bargiela, Jon Pey, Florian-Alexander Herbst, Iván Lores, David Rojo, Coral Barbas, Ana I Peláez, Jesús Sánchez, Martin von Bergen, et al. Context-specific metabolic network reconstruction of a naphthalene-degrading bacterial community guided by metaproteomic data. *Bioinformatics*, 31(11):1771–1779, 2015.

[TVF14] Ines Thiele, Nikos Vlassis, and Ronan MT Fleming. fastGapFill: efficient gap filling in metabolic networks. *Bioinformatics*, 30(17):2529–2531, 2014.

[VPS14] Nikos Vlassis, Maria Pires Pacheco, and Thomas Sauter. Fast reconstruction of compact context-specific metabolic network models. *PLoS Computational Biology*, 10(1):e1003424, 2014.

[WEP12] Yuliang Wang, James A Eddy, and Nathan D Price. Reconstruction of genome-scale metabolic models for 126 human tissues using mCADRE. *BMC Systems Biology*, 6(1):153, 2012.

[ZRS10] Hadas Zur, Eytan Ruppin, and Tomer Shlomi. iMAT: an integrative metabolic analysis tool. *Bioinformatics*, 26(24):3140–3142, 2010.

Appendix

To the end of this Appendix, without loss of generality we assume that all the reactions are irreversible, *i.e.*, $\mathcal{I} = \mathcal{R}$. If this is not the case, we can replace each reversible reaction by a pair of irreversible reactions corresponding to its forward and reverse directions.

We define the skeleton digraph of a metabolic network as $G = (\mathcal{M}, \mathcal{A})$ where $\mathcal{M}$ is the set of metabolites and $\mathcal{A}$ is the set of ordered pairs of metabolites (M_i, M_j) for which there exists a reaction $R_k \in \mathcal{R}$ whose reactants and products include M_i and M_j, respectively.

We define the mapping $f : \mathcal{R} \longrightarrow \mathcal{P}(\mathcal{A})$ by

$$f(R_k) = \{(M_i, M_j) \mid S_{ik} < 0, S_{jk} > 0\},$$

where $\mathcal{P}(\mathcal{A})$ denotes the power set (the set of all the subsets) of $\mathcal{A}$. Moreover, we define its extension $\tilde{f} : \mathcal{P}(\mathcal{R}) \longrightarrow \mathcal{P}(\mathcal{A})$ by

$$\tilde{f}(\tilde{\mathcal{R}}) = \bigcup_{R_k \in \tilde{\mathcal{R}}} f(R_k).$$

Note that, $\tilde{f}(\mathcal{R}) = \mathcal{A}$. Intuitively, $\tilde{f}$ of a subset of reactions $\tilde{\mathcal{R}} \subseteq \mathcal{R}$ is the corresponding directed arcs in $\mathcal{A}$ by breaking any hyperarcs in $\tilde{\mathcal{R}}$ into directed arcs.

Let $\mathcal{A} = \{A_1, A_2, \ldots, A_a\}$, and S_k denote the kth column of S. Suppose that $\mathbf{1}^T S = 0$. Then for any $R_k \in \mathcal{R}$ we can define

$$C_k = -\sum_{S_{ik}<0} S_{ik} = \sum_{S_{jk}>0} S_{jk} = \frac{1}{2}\sum_{l=1}^{m} |S_{lk}|,$$

and we have that

$$\begin{aligned} S_k &= \sum_{S_{ik}<0} S_{ik} e_i + \sum_{S_{jk}>0} S_{jk} e_j \\ &= \sum_{S_{ik}<0} \sum_{S_{jk}>0} \frac{S_{ik}S_{jk}}{C_k} e_i - \sum_{S_{jk}>0} \sum_{S_{ik}<0} \frac{S_{ik}S_{jk}}{C_k} e_j \\ &= \sum_{S_{ik}<0, S_{jk}>0} \frac{-S_{ik}S_{jk}}{C_k} (e_j - e_i). \end{aligned}$$

Therefore,

$$S = \partial F,$$

where ∂ is the incidence matrix of G and F is the nonnegative $l \times n$ matrix whose entries are

$$f_{lk} = \frac{-S_{ik}S_{jk}}{C_k},$$

if $A_l = (M_i, M_j) \in f(R_k)$ and $f_{lk} = 0$ if $A_l \notin f(R_k)$.

The support of a vector $v \in \mathbf{R}^n$, denoted by $\operatorname{supp}(v)$, is defined to be the set of its nonzero indices. With a slight abuse of notation, this is also used to show the set of reactions in $\mathcal{R}$ which are active in a flux distribution, *i.e.*,

$$\operatorname{supp}(v) = \{R_i \in \mathcal{R} \mid v_i \neq 0\}.$$

Suppose that $\mathbf{1}^T S = 0$. We claim that $\tilde{f}(\operatorname{supp}(v))$ is a strongly connected subgraph of G for any steady-state flux distribution v. The sketch of the proof is to first note that

$$\tilde{f}(\operatorname{supp}(v)) = \bigcup_{v_k \neq 0} f(R_k) = \bigcup_{v_k \neq 0} \operatorname{supp}(F_k) = \operatorname{supp}(Fv)$$

but then $\partial(Fv) = Sv = 0$ and hence $\operatorname{supp}(Fv)$ is strongly connected and so is $\tilde{f}(\operatorname{supp}(v))$.

This observation gives rise to a preprocessing step for any flux consistency checking algorithm by partitioning G into its strongly connected components, and any reaction which is not inside a strongly connected component would be blocked. Then the flux consistency checking algorithm is called on each one of them to get the rest of blocked reactions. It only remains to show that the assumption $\mathbf{1}^T S = 0$ holds for a broad class of metabolic networks with a slight modification to internalize their boundary reactions but without changing their steady-state flux distributions.

The boundary reactions $\mathcal{R}_B \subsetneq \mathcal{R}$, as the name suggests, lie on the boundary of a given metabolic network, *e.g.*, exchange reactions with extracellular metabolites, and all the reactants utilized or all the products formed by these reactions are external, *i.e.*, their corresponding stoichiometric vectors are either nonnegative or nonpositive. Consequently,

$$\tilde{f}(\mathcal{R}_B) = \emptyset.$$

On the other hand, internal reactions $\mathcal{R}_I$ are the subset of $\mathcal{R}$ which only comprise internal metabolites in $\mathcal{M}$, hence the name internal. Next, we will see that this distinction is necessary for the stoichiometric consistency analysis where we restrict our attention to the internal reactions. Otherwise, the missing information on extracellular metabolites can be misinterpreted as stoichiometric inconsistency in the metabolic network.

Let S_I denote the submatrix of S restricted to the columns indexed by $\mathcal{R}_I$, and $w \in \mathbf{R}^m$ denote the positive vector of the molecular masses of $\mathcal{M}$. By the law of mass conservation in a stoichiometrically consistent metabolic network, the sum of molecular masses of $\mathcal{M}$ weighted by their associated stoichiometric coefficients in any arbitrary $R_i \in \mathcal{R}_I$ must be equal to zero. Therefore, any $w > 0$ which fulfils the mass conservation law is assumed to satisfy $w^T S_I = 0$ by our earlier convention. A metabolic network is called stoichiometrically consistent if there exists at least one molecular mass vector w such that

$$\begin{aligned} &S_I^T w = 0 \\ &w > 0, \end{aligned} \tag{6}$$

and stoichiometrically inconsistent otherwise [GPF08]. We note that, determining whether a metabolic network is stoichiometrically consistent or not by (6) is the same feasibility problem as (1) by replacing S and $\mathcal{I}$ with S_I^T and $\mathcal{M}$.

Consider the following stoichiometric matrix

$$S' = \begin{bmatrix} WS \\ -w^T S \end{bmatrix}$$

where $w > 0$ is an arbitrary molecular mass vector and W is the diagonal matrix whose diagonal entries are w. We associate the additional row with a fictitious extracellular metabolite M_{m+1}, and because of $w^T S_I = 0$, the stoichiometry of the internal reactions do not involve this newly added metabolite. Since the last row of S' is a linear combination of the other rows, we have $S'v = 0 \Leftrightarrow Sv = 0$. Therefore, we can replace S by S' and the set of the steady-state flux distributions does not change. Furthermore, if S_i is either nonnegative or nonpositive, then the ith entry of $-w^T S$ has the other sign and hence with respect to S' all reactions are internal, *i.e.*, $S'_I = S'$. The boundary reactions which were previously defined to have either nonnegative or nonpositive stoichiometric vectors correspond to the reactions which involve the fictitious metabolite M_{m+1} in this new setting. Eventually,

$$\mathbf{1}^T S' = \mathbf{1}^T \begin{bmatrix} WS \\ -w^T S \end{bmatrix} = \mathbf{1}^T WS - w^T S = 0.$$

As a final remark, we recall that in order to construct G and f we only need to know the signs of the elements of S', thus this can be done without computing w explicitly. Even when we do not know whether the metabolic network is stoichiometrically consistent or not, we can still construct G, and if it is not strongly connected, we conclude that the metabolic network has either stoichiometric or flux inconsistencies.