Bioinformatics, 36(5), 2020, 1644–1646
doi: 10.1093/bioinformatics/btz761
Advance Access Publication Date: 9 October 2019
Applications Note

Systems biology

ErrorTracer: an algorithm for identifying the origins of inconsistencies in genome-scale metabolic models

Nikolay Martyushenko[1,*] and Eivind Almaas[1,2,*]

[1]Department of Biotechnology and [2]Department of Public Health and General Practice, K.G. Jebsen Center for Genetic Epidemiology, NTNU – Norwegian University of Science and Technology, Trondheim N-7491, Norway

*To whom correspondence should be addressed.

Associate Editor: Jonathan Wren

Received on February 25, 2019; revised on September 25, 2019; editorial decision on September 28, 2019; accepted on October 3, 2019

Abstract

Motivation: The number and complexity of genome-scale metabolic models is steadily increasing, empowered by automated model-generation algorithms. The quality control of the models, however, has always remained a significant challenge, the most fundamental being reactions incapable of carrying flux. Numerous automated gap-filling algorithms try to address this problem, but can rarely resolve all of a model's inconsistencies. The need for fast inconsistency checking algorithms has also been emphasized with the recent community push for automated model-validation before model publication. Previously, we wrote a graphical software to allow the modeller to solve the remaining errors manually. Nevertheless, model size and complexity remained a hindrance to efficiently tracking origins of inconsistency.

Results: We developed the ErrorTracer algorithm in order to address the shortcomings of existing approaches: ErrorTracer searches for inconsistencies, classifies them and identifies their origins. The algorithm is ~2 orders of magnitude faster than current community standard methods, using only seconds even for large-scale models. This allows for interactive exploration in direct combination with model visualization, markedly simplifying the whole error-identification and correction work flow.

Availability and implementation: Windows and Linux executables and source code are available under the EPL 2.0 Licence at https://github.com/TheAngryFox/ModelExplorer and https://www.ntnu.edu/almaaslab/downloads.

Contact: nikolay.martyushenko@ntnu.no or eivind.almaas@ntnu.no

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

The existence of multiple software platforms for the automated reconstruction and curation of genome-scale metabolic networks (e.g. Arkin *et al.*, 2018; Karp *et al.*, 2016; Wang *et al.*, 2018) has made such models commonplace. However, despite the existence of a variety of automated gap-filling algorithms incorporated in these tools (Faria *et al.*, 2018), a large number of existing models contain significant errors, such as energy-generating cycles (Fritzemeier *et al.*, 2017) and blocked reactions that leave parts of the metabolic network unable to carry flux. Typical gap-filling tools are narrowly focused on ensuring that a metabolic network produces biomass, and these tools draw upon the same reaction databases as were used for making the model in the first place. Additionally, model consistency-checking is at the center of the current community-push for standardized model testing and quality assessment (Lieven *et al.*, 2018), and it is necessary with fast algorithms for consistency checking and error identification. To address these challenges, we have developed a novel set of algorithms, called ErrorTracer, which are implemented in a published graphical model-correction framework (Martyushenko and Almaas, 2019). We demonstrate that ErrorTracer is not only orders of magnitude faster than existing algorithms (Dreyfuss *et al.*, 2013; Martyushenko and Almaas, 2019; Vlassis *et al.*, 2014) at finding inconsistent reactions, but can also identify non-trivial model elements causing the inconsistencies.

2 Approach

The ErrorTracer algorithm is a hybrid between logical inference and linear optimization (see Fig. 1a for an overview, and Section 2 of Supplementary Note S1 for a detailed description of the algorithm). In the first part, the logical inference steps simplify the model, identifying local metabolic network errors in the process. This model-reduction phase is based on three principles: (i) fusion of duplicate reactions, (ii) concatenation of reaction pairs that share a common metabolite not shared with any other reaction and (iii) conditional removal of metabolites interfacing import/export reactions (see Supplementary Fig. S5 for an illustration of the rules of local error spreading). Model errors determined at this stage are classified

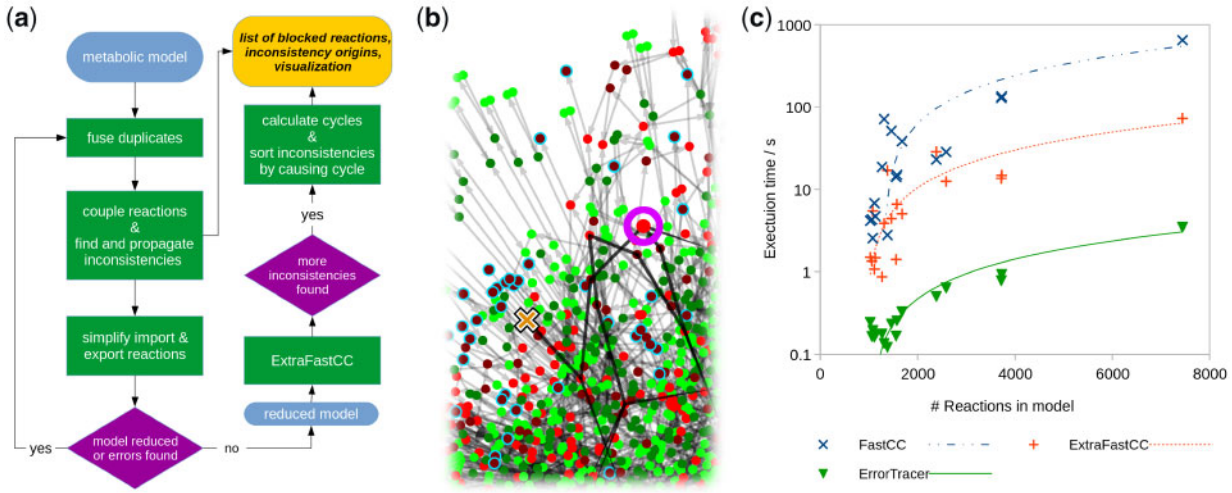

Fig. 1. Overview of ErrorTracer: (**a**) A flow diagram of the ErrorTracer algorithm. In the main loop (left side), logical simplifications are performed on a model. This process identifies local inconsistencies, traces which reactions they influence and reduces the model size. When no further reductions are possible (right side), the algorithm ExtraFastCC is executed on the reduced model for consistency checking and identified inconsistencies are attributed to different constrained reaction cycles. (**b**) An example visualization of the iTO977 *S.cerevisiae* model in ModelExplorer (Martyushenko and Almaas, 2019) showing the lack of export of chitosan (M_m607) as an origin of inconsistency (orange cross). This origin causes a halt in chitin synthesis (R_CHS1), visualized as a red node inside an added purple circle. The extent of influence of the inconsistency origin is highlighted using black lines. (**c**) Execution times of the three state-of-the-art algorithms FastCC (Vlassis *et al.*, 2014), ExtraFastCC (Martyushenko and Almaas, 2019), and ErrorTracer with corresponding linear trend lines

as: *source*, *reversibility* or *stoichiometry* (for a detailed description of the error types, see Section 1 of Supplementary Note S1). Based on the analysis of 17 commonly used metabolic models (see Supplementary Table S1) these errors on average amount to ~85% of the total error count in a genome-scale metabolic model.

In the second part of ErrorTracer (Fig. 1a, second column), we first identify remaining errors using our previous algorithm, ExtraFastCC (Martyushenko and Almaas, 2019). Subsequently, ErrorTracer determines stoichiometrically constrained cycles within the model which could cause these inconsistencies, assigning each inconsistency to its respective cycle. These errors are termed *cycle* errors. It is theoretically possible that the metabolic model would have errors being neither local nor cycle-related, and the algorithm warns the user if such inconsistencies are discovered. However, testing a large number of genome-scale models, we have not observed such errors: Chemical equations with integer stoichiometries very much reduce the scope of errors that a modeller could possibly encounter. The final step is to visually present the errors and dependencies within the interactive ModelExplorer framework (see Fig. 1b).

3 Results

In order to assess the speed of our algorithm, we tested ErrorTracer on a range of 17 genome-scale reconstructed models (see Supplementary Table S1) from the OpenCOBRA repository previously used by Ebrahim *et al.*, 2015, the models ranging in size from about 1000 to 7500 reactions. We compared the execution time of ErrorTracer on these models with our previous algorithm—ExtraFastCC (Martyushenko and Almaas, 2019) and with its predecessor FastCC (Vlassis *et al.*, 2014) (Fig. 1c). ErrorTracer is one order of magnitude faster than the others on smaller models, with the difference increasing to more than two orders of magnitude against FastCC for the largest model, RECON2 (Thiele *et al.*, 2013). The execution speed difference is even greater if we compare ErrorTracer with the modern cycle-free flux variability algorithms Fast-SNP (Saa and Nielsen, 2016) and LLC-NS (Chan *et al.*, 2018). These are up to three orders of magnitude slower (see Supplementary Fig. S4), probably due to the additional constrain of only being allowed to attain non-cyclic flux distributions.

ErrorTracer also demonstrates relatively homogeneous execution times, with all of the values falling between 0.12 and 3.5 s on an Intel Core i5-5300U CPU. This gave a longest to shortest time ratio of 28 as compared to 84 for ExtraFastCC and 250 for FastCC. The execution time of FastCC was found to be proportional to the product of total reaction number with the number of reversible blocked reactions in the model. The other algorithms scaled with the square of the total reaction number, but with much smaller proportionality coefficients (Supplementary Fig. S3).

Assessing the complexity of the different subroutines of ErrorTracer, we found that the initial logical reduction and error tracing scales linearly with model size (Supplementary Fig. S1a). The size of the resulting reduced model also showed a clear linear dependence on the size of the original model (Supplementary Fig. S1b). The subsequent ExtraFastCC-based subroutine showed a quadratic dependence on the size of the reduced model (Supplementary Fig. S2a), with the values showing significantly less spread than those obtained with same approach run on the full model (Supplementary Fig. S2b). Additionally, model reduction with ErrorTracer allowed ExtraFastCC in the second part of the algorithm to use the faster but less stable reduced-gradient method instead of the slower barrier optimization used in previous versions. This indicates that the ErrorTracer logical-reduction algorithm can make models more numerically tractable for LP solvers in addition to reducing their size.

4 Discussion

ErrorTracer provides a significant improvement to the time-consuming process of correcting metabolic reconstructions by identifying model inconsistencies and pin-pointing the causes of errors. Additionally, the fast algorithms of ErrorTracer is a much needed addition in the community push for standardized consistency checking of models of any size.

Funding

M.N. and E.A. would like to thank The Research Council of Norway grant 245160 (ERASysAPP: WineSys) and 271585 (ERA-IB2: PolyBugs) for funding.

Conflict of Interest: none declared.

References

Arkin,A.P. *et al.* (2018) Kbase: the united states department of energy systems biology knowledgebase. *Nat. Biotechnol.*, **36**, 566.

Chan,S.H.J. *et al*. (2018) Accelerating flux balance calculations in genome-scale metabolic models by localizing the application of loopless constraints. *Bioinformatics*, **34**, 4248–4255.

Dreyfuss,J.M. *et al*. (2013) Reconstruction and validation of a genome-scale metabolic model for the filamentous fungus Neurospora crassa using FARM. *PLoS Comput. Biol*., **9**, e1003126.

Ebrahim,A. *et al*. (2015) Do genome-scale models need exact solvers or clearer standards? *Mol. Syst. Biol*., **11**, 831.

Faria,J.P. *et al*. (2018) Methods for automated genome-scale metabolic model reconstruction. *Biochem. Soc. Trans*., **46**, 931–936.

Fritzemeier,C.J. *et al*. (2017) Erroneous energy-generating cycles in published genome scale metabolic networks: identification and removal. *PLoS Comput. Biol*., **13**, e1005494.

Karp,P.D. *et al*. (2016) Pathway tools version 19.0 update: software for pathway/genome informatics and systems biology. *Brief. Bioinform*., **17**, 877–890.

Lieven,C. *et al*. (2018) Memote: a community-driven effort towards a standardized genome-scale metabolic model test suite. *bioRxiv*, 350991. doi: 10.1101/350991.

Martyushenko,N. and Almaas,E. (2019) Modelexplorer – software for visual inspection and inconsistency correction of genome-scale metabolic reconstructions. *BMC Bioinformatics*, **20**, 56.

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.

Thiele,I. *et al*. (2013) A community-driven global reconstruction of human metabolism. *Nat. Biotechnol*., **31**, 419–425.

Vlassis,N. *et al*. (2014) Fast reconstruction of compact context-specific metabolic network models. *PLoS Comp. Biol*., **10**, e1003424.

Wang,H. *et al*. (2018) Raven 2.0: a versatile toolbox for metabolic network reconstruction and a case study on *Streptomyces coelicolor*. *PLoS Comput. Biol*., **14**, e1006541.