PAPER

dingo: a Python package for metabolic flux sampling

Apostolos Chalkis,[1,*] Vissarion Fisikopoulos,[1,2] Elias Tsigaridas[1,3] and Haris Zafeiropoulos[1,4,*]

[1] GeomScale org. , [2] Department of Informatics & Telecommunications, National & Kapodistrian University of Athens, Panepistimioupolis, Ilisia, 16122, Athens, Greece , [3]Inria Paris and IMJ-PRG, Sorbonne Université and Paris Université, France and [4]Department of Microbiology, Immunology and Transplantation, Rega Institute for Medical Research, Laboratory of Molecular Bacteriology, KU Leuven, Street, 3000, Leuven, Belgium

[*] Apostolos Chalkis tolis.chal@gmail.com Haris Zafeiropoulos haris.zafeiropoulos@kuleuven.be

FOR PUBLISHER ONLY Received on Date Month Year; revised on Date Month Year; accepted on Date Month Year

Abstract

We present `dingo`, a Python package that supports a variety of methods to sample from the flux space of metabolic models, based on state-of-the-art random walks and rounding methods. For uniform sampling `dingo`'s sampling methods provide significant speed-ups and outperforms existing software. Indicatively, `dingo` can sample from the flux space of the largest metabolic model up to now (Recon3D) in less than a day using a personal computer, under several statistical guarantees; this computation is out of reach for other similar software. In addition, `dingo` supports common analysis methods, such as Flux Balance Analysis (FBA) and Flux Variability Analysis (FVA), and visualization components. `dingo` contributes to the arsenal of tools in metabolic modeling by enabling flux sampling in high dimensions (in the order of thousands).

Key words: metabolic modelling, random sampling, MCMC, polytope

Introduction

Metabolic models enhance the study of the relationship between genotype and phenotype in an attempt to elucidate the mechanisms that govern the physiology and the growth of a species and/or a community (13). By optimizing a linear objective function over a polytope, Flux Balance Analysis (FBA) identifies a single optimal flux distribution (14). Flux Variability Analysis (FVA) reveals the limits of the solution space (7). Contrary to FBA and FVA, flux sampling is an unbiased method, as it does not depend on the selection of the objective function. It allows us to cover all the possible flux values by estimating a probability distribution for the flux value of a certain reaction (16).

The ability to sample (efficiently) points from the convex polytope corresponding to (the steady states of) a metabolic model allows us to investigate its whole solution space. This way, we can obtain a more detailed insight of a system at steady state; where the production rate of each metabolite equals its consumption rate. Alternatively, we can perform flux sampling after optimizing an objective function, and approximate the flux distributions in optimal scenarios.

Even though flux sampling has proved itself by delivering great insights in a range of applications (9), high dimensionality- and anisotropy-oriented limitations (16) force the current implementations to struggle or even to fail in several cases (4). A range of Markov Chain Monte Carlo (MCMC) algorithms and implementations have been developed to address this obstacle (4) (see Supplementary File–Section 1). In this setting, we present `dingo`, a Python package that supports efficient flux sampling, based on a variety of state-of-the-art MCMC sampling algorithms; it also provides classical FBA and FVA methods and advanced visualisations.

Implementation

`dingo` is an open-source Python package that exploits the efficiency of `volesti`, an open-source `C++` software library that implements high-dimensional MCMC sampling and volume approximation algorithms.

`dingo` supports a variety of MCMC algorithms for uniform sampling. Among them, the Multiphase Monte Carlo Sampling (MMCS) algorithm (2) has been reported as the most efficient algorithm in practice. MMCS unifies rounding and sampling of a convex polytope in one pass, obtaining both upon termination. In this study, we show that combining the rounding of `PolyRound` with the optimized Billiard walk implementation of `dingo`, i.e., Billiard Walk with Rounding (BWR), yields the fastest sampling for the networks we test (up to dimension 5 335). Interestingly, we show that as the networks' dimension increases, MMCS will overrule. An example of an application that creates polytopes in higher dimensions is sampling from the solution space of community models where several metabolic networks are combined.

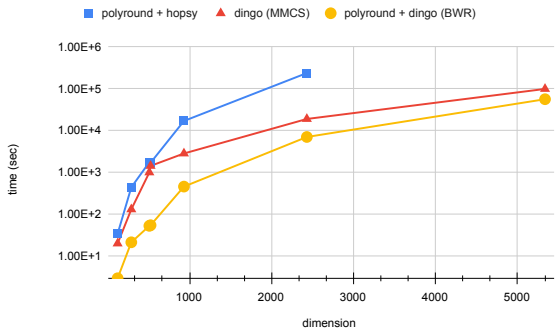

Fig. 1. Comparison of three sampling methods (`PolyRound` with `hopsy`, `dingo`'s MMCS and `PolyRound` with `dingo`'s BWR) when sampling from the flux space of 7 GEMs corresponding to polytopes of dimension ranging from 122 to 5335, under the same statistical guarantees. `PolyRound` was used for rounding with `hopsy` and `dingo`'s Billiard Walk used to sample from the rounded polytope. `dingo`'s MMCS run-time corresponds to both rounding and sampling, starting from the non-rounded polytope (i.e. same as the input of `PolyRound`).

`dingo` enables the performance of the MMCS algorithm in parallel threads and uses the state-of-the-art linear programming solvers of `Gurobi` [8]. It ensures the quality of the output samples using two widely used diagnostics, the Effective Sample Size (ESS) (6) and the Potential Scale Reduction Factor (PSRF) (5); `dingo` guarantees bounded values for both diagnostics for the returned sample. In addition to the MMCS algorithm and the optimized Billiard walk, it also supports the Random Directions Hit-and-Run (RDHR) (17), the Coordinate Directions Hit-and-Run (CDHR) (17), the Dikin (11), the John and Vaidya (3) and the Ball Walk (12) sampling algorithms.

We ensure the correctness of `dingo`'s functionality using a set of unit tests running on a continuous integration platform. All three main formats for metabolic models (.xml, .json and .mat) are supported. A tutorial is available as a Google Colab notebook.

Performance comparison and illustrations

Currently, the most efficient way to perform flux sampling, to the best of our knowledge, is to combine the `PolyRound` Python package (18) (for rounding the polytope) with CDHR sampling algorithm as implemented in the the `HOPS` C++ library (10) to sample from the rounded polytope. We compare two sampling methods implemented in `dingo` against the combination of `PolyRound` and `hopsy` (the Python interface of `HOPS`), over a set of models having dimension from around 100 to more than 13,000. In all cases, we perform a pre-processing step using `PolyRound`. `dingo` performs rounding and sampling in one step using the MMCS algorithm or using an optimized Billiard walk to sample from the rounded polytope obtained by `PolyRound` (see also Supplementary File–Section 2), i.e., BWR. To ensure that the quality of the sample provides an accurate approximation of the target distribution, we require an ESS of 1.000 and a PSRF of at most 1.1, in all cases. To our knowledge `dingo` is the only software that provides this combined statistical guarantee. For all the tested models `dingo` is faster than `PolyRound` / `hopsy`. Moreover, `dingo`'s added value highlights as the model's dimensions increases (see Fig. 1). Indicatively, `dingo` can sample the latest version of the human metabolic network, the Recon3D model (1), in less than 16 hours, using modest hardware; while after 10 days, `hopsy` did not converge.

To demonstrate `dingo`'s flux sampling and illustrations tools in a real-world scenario, we use the integrated human alveolar macrophage model with the virus biomass objective function (VBOF) of Sars-Cov-2 (15) (see Supplementary File–Section 3). Notably, our findings confirm the authors' indicating Guanylate Kinase 1 as a potential therapeutic target.

Conclusions

`dingo` is a Python package that employs efficient C++ MCMC implementations from `volesti` library. It supports a variety of MCMC algorithms and classical methods as FBA and FVA. `dingo` unlocks the fastest implementation for sampling current metabolic networks namely BWR. Additionally, `dingo` provides an implementation of the MMCS algorithm that is also more efficient that the current state-of-the-art but also our experiments denote that it would be faster than BWR for higher dimensions. It also offers statistical and illustration tools, like copula estimation, that can help the user to extract useful information about the model (see Supplementary File–Section 3). `dingo` facilitates the survey of the largest models available for the time being assuring for the first time high quality of the samples returned. Moreover, it requires minimum computational resources requirements and aims to support a broad spectrum of research and application needs via a user-friendly design.

Competing interests

No competing interest is declared.

Author contributions statement

A.C., V.F., E.T. and H.Z. contributed to the conceptualization, methodology, and software development, and they all wrote and reviewed the manuscript.

Acknowledgments

The authors thank the anonymous reviewers for their valuable suggestions. This work is supported in part by Google Summer of Code. We also thank Area 52 lab in University College Dublin for allowing us to use the *bob* server to run the comparison experiments.

References

1. 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.
2. Apostolos Chalkis, Vissarion Fisikopoulos, Elias Tsigaridas, and Haris Zafeiropoulos. Geometric Algorithms for Sampling the Flux Space of Metabolic Networks. In Kevin Buchin and Éric Colin de Verdière, editors, *37th International Symposium on Computational Geometry (SoCG 2021)*, volume 189 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 21:1–21:16,

Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

3. Yuansi Chen, Raaz Dwivedi, Martin J. Wainwright, and Bin Yu. Fast MCMC Sampling Algorithms on Polytopes. *Journal of Machine Learning Research*, 19(55):1–86, 2018.
4. Shirin Fallahi, Hans J Skaug, and Guttorm Alendal. A comparison of Monte Carlo sampling methods for metabolic network models. *PLOS One*, 15(7):e0235393, 2020.
5. Andrew Gelman and Donald B. Rubin. Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science*, 7(4):457–472, 1992. Publisher: Institute of Mathematical Statistics.
6. Charles J. Geyer. Practical Markov Chain Monte Carlo. *Statist. Sci.*, 7(4):473–483, 11 1992.
7. Steinn Gudmundsson and Ines Thiele. Computationally efficient flux variability analysis. *BMC bioinformatics*, 11(1):1–3, 2010.
8. Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023.
9. Helena A Herrmann, Beth C Dyson, Lucy Vass, Giles N Johnson, and Jean-Marc Schwartz. Flux sampling is a powerful tool to study metabolism under changing environmental conditions. *NPJ systems biology and applications*, 5(1):1–8, 2019.
10. Johann F Jadebeck, Axel Theorell, Samuel Leweke, and Katharina Noh. Hops: high-performance library for non uniform sampling of convex constrained models. *Bioinformatics*, 37:1776–1777, 2021.
11. Ravindran Kannan and Hariharan Narayanan. Random walks on polytopes and an affine interior point method for linear programming. *Mathematics of Operations Research*, 37(1):1–20, 2012.
12. László Lovász, Ravi Kannan, and Miklós Simonovits. Random walks and an $O^*(n^5)$ volume algorithm for convex bodies. *Random Structures and Algorithms*, 11:1–50, 1997.
13. Andrew Morris, Kyle Meyer, and Brendan Bohannan. Linking microbial communities to ecosystem functions: what we can learn from genotype–phenotype mapping in organisms. *Philosophical Transactions of the Royal Society B*, 375(1798):20190244, 2020.
14. Jeffrey D Orth, Ines Thiele, and Bernhard Ø. Palsson. What is flux balance analysis? *Nature biotechnology*, 28(3):245–248, 2010.
15. Alina Renz, Lina Widerspick, and Andreas Dräger. FBA reveals guanylate kinase as a potential target for antiviral therapies against SARS-CoV-2. *Bioinformatics*, 36(Supplement_2):i813–i821, December 2020.
16. Jan Schellenberger and Bernhard Ø. Palsson. Use of randomized sampling for analysis of metabolic networks. *Journal of biological chemistry*, 284(9):5457–5461, 2009.
17. Robert L. Smith. Efficient Monte Carlo procedures for generating points uniformly distributed over bounded regions. *Operations Research*, 32(6):1296–1308, 1984.
18. Axel Theorell, Johann F Jadebeck, Katharina Nöh, and Jörg Stelling. Polyround: Polytope rounding for random sampling in metabolic networks. *Bioinformatics*, 07 2021.