gapsplit: Efficient random sampling for non-convex constraint-based models

Thomas C. Keaty and Paul A. Jensen*

Department of Bioengineering and Carl R. Woese Institute for Genomic Biology, University of Illinois at Urbana-Champaign, Urbana, IL, USA

*Correspondence to pjens@illinois.edu

Abstract

Summary: GAPSPLIT generates random samples from convex and non-convex constraint-based models. GAPSPLIT targets under-sampled regions of the solution space for uniform coverage.

Availability and Implementation: Python and Matlab source code are freely available at `http://jensenlab.net/tools`.

Contact: pjens@illinois.edu

Introduction

Constraint-based models allow systems-level interrogation of biochemical networks with minimal kinetic data. The number of variables in constraint-based models far exceeds the number of kinetic parameters, resulting in underdetermined systems of equations that produce an infinite number of solutions. Rather than focus on any single solution, modelers can use ensembles of randomly sampled solutions to analyze network properties (Schellenberger and Palsson, 2009).

The leading algorithms for sampling constraint-based models use the "hit-and-run" (HR) framework (Smith, 1984). HR sampling walks through a model's solution space by randomly selecting directions based on a set of warmup points. HR's efficiency is tied to the convexity of the solution space. Since a convex combination of any number of existing solutions is also a solution, HR algorithms can quickly generate new solutions without resolving the model. When applied to constraint-based models, current HR samplers (ACHR (Kaufman and Smith, 1998), OptGP (Megchelenbrink et al., 2014), and CHRR (Haraldsdóttir et al., 2017)) quickly generate a series of samples that converge to a stable distribution. However, models containing reactions with fixed bounds can drastically reduce the fraction of the total sample space covered by the HR samplers (see Binns et al. (2015) and data below). One random sampler with improved coverage uses a "poling" method to push the random walk of the HR sampler away from previous samples (Binns et al., 2015). While the poling method improved coverage, the resulting optimization problems are nonlinear and require orders of magnitude more computation time.

Adding transcriptional regulation or enzymatic complexes to a model requires discrete variables, making the model non-convex. HR samplers cannot directly sample non-convex models. As a workaround, the ll-ACHR sampler uses a boxing approximation to sample models with (non-convex) loopless flux constraints (Saa and Nielsen, 2016). The box constraints enclose the non-convex solution space with a convex hull, and the convex hull, not the original solution space, is sampled. Care must be taken to reject any infeasible samples that lie outside the solution space but inside the convex hull. The efficiency of boxed models also decreases with additional discrete variables or non-convex constraints (Kiatsupaibul et al., 2011).

We present a new class of random sampler for constraint-based models. Our algorithm – called GAPSPLIT – uses mathematical programming to find solutions in the underexplored areas of the model's solution space. Unlike HR algorithms, GAPSPLIT samples convex and non-convex models directly. Samples identified by GAPSPLIT uniformly cover a model's solution space. GAPSPLIT yields better coverage than HR samplers for tightly constrained and non-convex models.

The GAPSPLIT sampler

GAPSPLIT is designed to find sample points that uniformly cover the entire solution space. GAPSPLIT's objective is to minimize the size of each variable's *max gap*, the largest interval between two adjacent sample points (Figure 1A). Given a set of samples, GAPSPLIT selects

a single variable and identifies its max gap. GAPSPLIT adds a constraint requiring the next solution be in the center of the max gap (the *target*; see Figure 1A). The model is solved to find such a solution, the constraint is removed, and the process is repeated with a different variable. To speed up sampling, GAPSPLIT also attempts to simultaneously split the max gaps of k randomly selected variables. GAPSPLIT uses a quadratic objective function to minimize the distance between the next solution and the centers of the max gaps for the k other reactions. (A complete description of the GAPSPLIT algorithm is presented in the Supplementary Methods.) The GAPSPLIT algorithm can be applied to any mathematical program including models with binary or integer constraints.

Results

One metric to assess the quality of a set of random samples is coverage, defined as

$$\text{coverage} = 1 - \text{mean}\left\{\text{relative max gap}(x_i)\right\}$$

where x_i is a variable in the model and the *relative max gap* is the max gap of x_i divided by the feasible range of x_i. Coverage ranges from 0 (all points at the edges of the feasible range) to 1 (uniform coverage by an infinite number of points). A coverage of 0.9 indicates that the model's variables, on average, have a maximum relative gap of 10%.

GAPSPLIT achieves better overall coverage with fewer samples than HR family algorithms. We generated samples from three genome-scale metabolic models using GAPSPLIT ACHR, and CHRR (Figure 1B). Both ACHR and CHRR plateaued within a few hundred samples at a coverage of 0.2, meaning each variable, on average, had an unsampled gap that covered 80% of the feasible space. GAPSPLIT samples quickly covered the solution space, reaching a coverage of 0.8 within 500 samples and plateauing with coverage over 0.9 after 2000 samples. The models were sampled as published with default bounds corresponding to glucose minimal media. We hypothesized that the fixed bounds prevented the HR samples from covering a larger fraction of the sample space. Indeed, opening all bounds to arbitrarily large values improved the coverage of the ACHR and CHRR samplers (Figure 1C). GAPSPLIT generated the best coverage for all models, but the HR samples also achieved coverage of at least 0.8. Thus GAPSPLIT gives better coverage of metabolic models especially when some of the variable bounds are fixed.

For models with fixed bounds, GAPSPLIT is slower than CHRR and faster than ACHR on a per sample basis (Figure 1D). However, GAPSPLIT is more efficient than either HR algorithm in the time required to reach a specific level of coverage since it yields better coverage per sample. GAPSPLIT is slower than either HR sampler for models with arbitrarily open bounds. However, we note that such models are not physiologically realistic since flux balance analysis requires at least one fixed constraint to limit nutrient uptake (Orth et al., 2010).

An advantage of GAPSPLIT is sampling non-convex models including those with discrete variables. We tested GAPSPLIT on two non-convex models of the yeast *Saccharomyces cerevisiae*: a metabolic model (Duarte et al., 2004) with gene-protein-reaction rules encoded as Boolean constraints (Shlomi et al., 2007; Jensen et al., 2011) and a combined

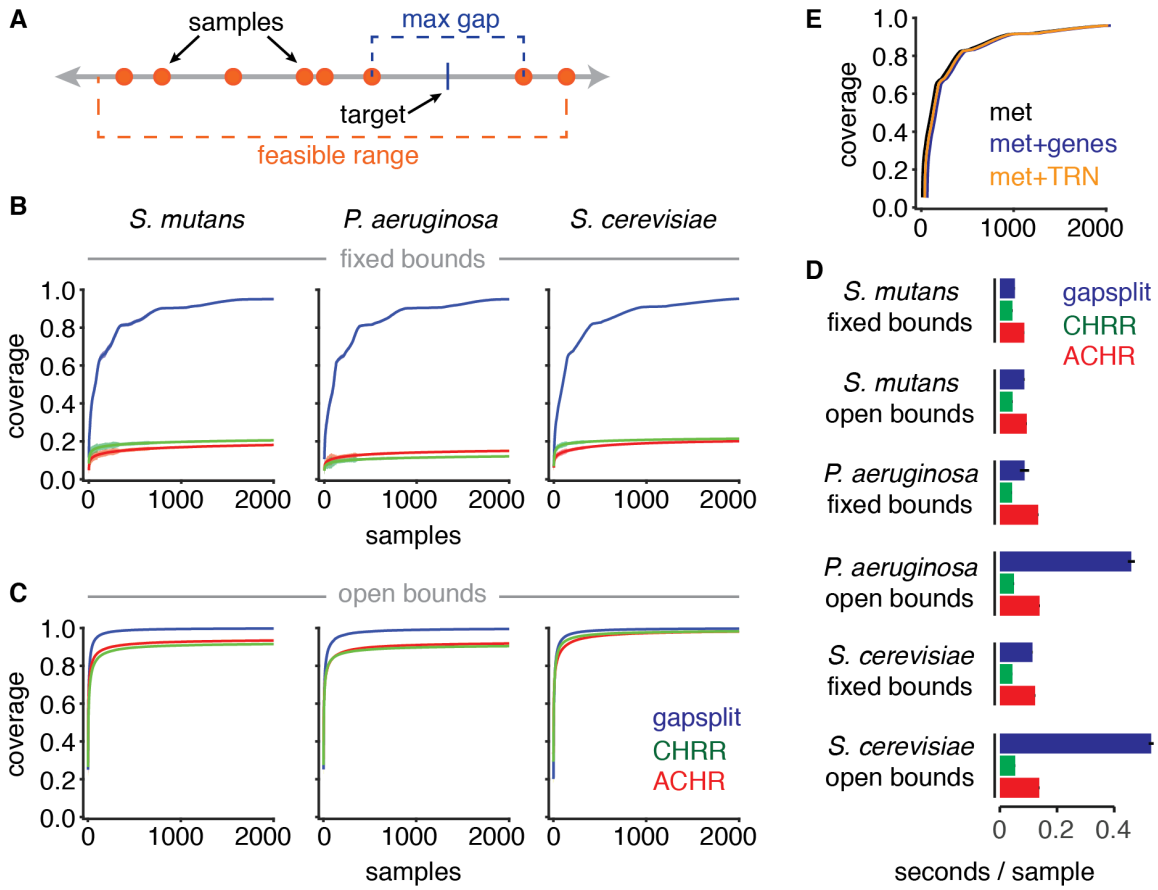

Figure 1. A. GAPSPLIT finds random samples for each variable (orange circles) within the variable's feasible range. If a variable is selected for targeting, the next solution will be at a target that splits the maximum remaining gap in half. **B.** GAPSPLIT yields better coverage with fewer samples than HR family samplers. The mean (solid line) and 95% confidence intervals (color shading) are shown for 100 runs of each sampler on three metabolic models: bacteria *Streptococcus mutans* (iSMU v1.0, Jijakli and Jensen (2018)) and *Pseudomonas aeruginosa* (iMO1056, Oberhardt et al. (2008)); and the yeast *Saccharomyces cerevisiae* (iND750, Duarte et al. (2004)). Bounds were fixed to glucose minimal media as specified in the original publications. **C.** The coverage of HR family samplers improves if the bounds on exchange reactions are relaxed to arbitrarily large values. However, such conditions are not physiologically reasonable. **D.** GAPSPLIT is slower than the CHRR but faster than the ACHR samplers when models have fixed bounds. Mean time per 1000 samples is shown for 25 independent runs of each algorithm. Error bars show the standard deviation. Opening the bounds on the model slows all three algorithms. **E.** GAPSPLIT can sample non-convex models. Adding binary constraints for gene associations (met+genes, blue) or logical constraints for transcriptional regulation (met+TRN, orange) does not affect sampling of the yeast metabolic model (met, black). Each line represents the mean coverage for 50 independent simulations.

metabolic/regulatory model (Herrgård et al., 2006). Adding hundreds of binary variables to the models did not affect the coverage during sampling (Figure 1E).

GAPSPLITś performance is tuned by changing only a single parameter: the number of secondary variables to target at each iteration. Changing this parameter (expressed as a fraction of the model's total number of variables) can affect GAPSPLIT's performance (Supplementary Figure S1). However, a single value (5%) was chosen as the default setting and worked well for all the experiments in this study. We do not expect users will need to tune this parameter for other models although they can change the parameter if needed.

Conclusions

GAPSPLIT is a new class of random sampler for constraint-based models. It samples convex and non-convex models and outperforms HR family samplers on models with fixed bounds. GAPSPLIT is available in Matlab and Python and is compatible with models from the COBRA Toolbox (Heirendt et al., 2019), TIGER (Jensen et al., 2011), and cobrapy (Ebrahim et al., 2013). We believe GAPSPLIT opens new possibilities for exploring non-convex models including models with transcriptional regulation. Using GAPSPLIT, researchers can develop mixed-integer algorithms that incorporate random sampling.

Acknowledgements

This work was supported by the National Institutes of Health grant EB027396. The authors declare no financial or commercial conflict of interest.

References

Michael Binns, Pedro de Atauri, Anestis Vlysidis, Marta Cascante, and Constantinos Theodoropoulos. Sampling with poling-based flux balance analysis: Optimal versus sub-optimal flux space analysis of Actinobacillus succinogenes. *BMC Bioinformatics*, 16 (1):49, February 2015. ISSN 1471-2105. doi: 10.1186/s12859-015-0476-5.

Natalie C. Duarte, Markus J. Herrgård, and Bernhard Ø Palsson. Reconstruction and validation of Saccharomyces cerevisiae iND750, a fully compartmentalized genome-scale metabolic model. *Genome research*, 14(7):1298–1309, July 2004. doi: 10.1101/gr.2250904.

Ali Ebrahim, Joshua A. Lerman, Bernhard Ø Palsson, and Daniel R. Hyduke. COBRApy: COnstraints-Based Reconstruction and Analysis for Python. *BMC systems biology*, 7(1): 74, August 2013. doi: 10.1186/1752-0509-7-74.

Hulda S. Haraldsdóttir, Ben Cousins, Ines Thiele, Ronan M. T. Fleming, and Santosh Vempala. CHRR: Coordinate hit-and-run with rounding for uniform sampling of constraint-based models. *Bioinformatics*, 33(11):1741–1743, June 2017. ISSN 1367-4803. doi: 10.1093/bioinformatics/btx052.

Laurent Heirendt, Sylvain Arreckx, Thomas Pfau, Sebastián N. Mendoza, Anne Richelle, Almut Heinken, Hulda S. Haraldsdóttir, Jacek Wachowiak, Sarah M. Keating, Vanja Vlasov, Stefania Magnusdóttir, Chiam Yu Ng, German Preciat, Alise Žagare, Siu H. J. Chan, Maike K. Aurich, Catherine M. Clancy, Jennifer Modamio, John T. Sauls, Alberto Noronha, Aarash Bordbar, Benjamin Cousins, Diana C. El Assal, Luis V. Valcarcel, Iñigo Apaolaza, Susan Ghaderi, Masoud Ahookhosh, Marouen Ben Guebila, Andrejs Kostromins, Nicolas Sompairac, Hoai M. Le, Ding Ma, Yuekai Sun, Lin Wang, James T. Yurkovich, Miguel A. P. Oliveira, Phan T. Vuong, Lemmer P. El Assal, Inna Kuperstein, Andrei Zinovyev, H. Scott Hinton, William A. Bryant, Francisco J. Aragón Artacho, Francisco J. Planes, Egils Stalidzans, Alejandro Maass, Santosh Vempala, Michael Hucka, Michael A. Saunders, Costas D. Maranas, Nathan E. Lewis, Thomas Sauter, Bernhard Ø Palsson, Ines Thiele, and Ronan M. T. Fleming. Creation and analysis of biochemical constraint-based models using the COBRA Toolbox v.3.0. *Nature Protocols*, 14(3):639, March 2019. ISSN 1750-2799. doi: 10.1038/s41596-018-0098-2.

Markus J. Herrgård, Baek-Seok Lee, Vasiliy Portnoy, and Bernhard Ø Palsson. Integrated analysis of regulatory and metabolic networks reveals novel regulatory mechanisms in Saccharomyces cerevisiae. *Genome research*, 16(5):627–635, May 2006. doi: 10.1101/gr.4083206.

Paul A. Jensen, Kyla A. Lutz, and Jason A. Papin. TIGER: Toolbox for integrating genome-scale metabolic models, expression data, and transcriptional regulatory networks. *BMC systems biology*, 5(1):147, 2011. doi: 10.1186/1752-0509-5-147.

Kenan Jijakli and Paul Anthony Jensen. Metabolic modeling of Streptococcus mutansreveals complex nutrient requirements of an oral pathogen. *bioRxiv*, page 419507, 2018. doi: 10.1101/419507.

David E. Kaufman and Robert L. Smith. Direction Choice for Accelerated Convergence in Hit-and-Run Sampling. *Operations Research*, 46(1):84–95, February 1998. ISSN 0030-364X. doi: 10.1287/opre.46.1.84.

Seksan Kiatsupaibul, Robert L. Smith, and Zelda B. Zabinsky. An Analysis of a Variation of Hit-and-run for Uniform Sampling from General Regions. *ACM Trans. Model. Comput. Simul.*, 21(3):16:1–16:11, February 2011. ISSN 1049-3301. doi: 10.1145/1921598.1921600.

Wout Megchelenbrink, Martijn Huynen, and Elena Marchiori. optGpSampler: An Improved Tool for Uniformly Sampling the Solution-Space of Genome-Scale Metabolic Networks. *PloS one*, 9(2):e86587, February 2014. doi: 10.1371/journal.pone.0086587.

Matthew A. Oberhardt, Jacek Puchałka, Kimberly E. Fryer, Vítor A. P. Martins dos Santos, and Jason A. Papin. Genome-Scale Metabolic Network Analysis of the Opportunistic Pathogen Pseudomonas aeruginosa PAO1. *Journal of Bacteriology*, 190(8):2790–2803, April 2008. ISSN 0021-9193, 1098-5530. doi: 10.1128/JB.01583-07.

Jeffrey D. Orth, Ines Thiele, and Bernhard Ø Palsson. What is flux balance analysis? *Nature biotechnology*, 28(3):245–248, March 2010. doi: 10.1038/nbt.1614.

Pedro A. Saa and Lars K. Nielsen. Ll-ACHRB: A scalable algorithm for sampling the feasible solution space of metabolic networks. *Bioinformatics (Oxford, England)*, 32(15): 2330–2337, July 2016. doi: 10.1093/bioinformatics/btw132.

J. Schellenberger and B. O. Palsson. Use of Randomized Sampling for Analysis of Metabolic Networks. *The Journal of biological chemistry*, 284(9):5457–5461, February 2009. doi: 10.1074/jbc.R800048200.

Tomer Shlomi, Yariv Eisenberg, Roded Sharan, and Eytan Ruppin. A genome-scale computational study of the interplay between transcriptional regulation and metabolism., 2007.

Robert L. Smith. Efficient Monte Carlo Procedures for Generating Points Uniformly Distributed Over Bounded Regions. *Operations Research*, 32(6):1296–1308, 1984. ISSN 0030-364X.