COBREXA.jl: constraint-based reconstruction and exascale analysis

Miroslav Kratochvíl[1, †], Laurent Heirendt[1,4, †], St. Elmo Wilken[2], Taneli Pusa[1,3], Sylvain Arreckx[1], Alberto Noronha[3], Marvin van Aalst[2], Venkata P Satagopam[1,4], Oliver Ebenhöh[2], Reinhard Schneider[1,4], Christophe Trefois[1,4, *], and Wei Gu[1,4, *]

[1]Luxembourg Centre for Systems Biomedicine, University of Luxembourg, Campus Belval, Belvaux, Luxembourg
[2]Institute of Quantitative and Theoretical Biology, Heinrich-Heine-Universität Düsseldorf, Universitätsstraße 1, 40225 Düsseldorf, Germany.
[3]Nium S.à.r.l., 6 Avenue des hauts fourneaux, L-4362 Esch-sur-Alzette, Luxembourg
[4]ELIXIR Luxembourg, 6, avenue du Swing, Campus Belval, L-4367 Belvaux, Luxembourg.
[†]Equal contributors.
[*]To whom correspondence should be addressed.

Summary: COBREXA.jl is a Julia package for scalable, high-performance constraint-based reconstruction and analysis of very large-scale biological models. Its primary purpose is to facilitate the integration of modern high performance computing environments with the processing and analysis of large-scale metabolic models of challenging complexity. We report the architecture of the package, and demonstrate its scalability by benchmarking the task distribution overhead incurred when performing analyses of many variants of multi-organism community models simultaneously.
Availability and implementation: `https://doi.org/10.17881/ZKCR-BT30`.
Contact: christophe.trefois@uni.lu, wei.gu@uni.lu

1 Introduction

Understanding metabolic interactions in cells is a crucial step to investigate disease mechanisms and to discover new therapeutics (Cook et al., 2017; Apaolaza et al., 2018; Brunk et al., 2018). Constraint-Based Reconstruction and Analysis (COBRA) is a promising methodology for analyzing various metabolic processes at the organism- and community- levels (Fang et al., 2020). The main idea behind COBRA is to represent an organism as a constrained set of interconnected reactions and metabolites based on genomic sequencing data. This leads to a straightforward interpretation of metabolism as a constrained linear system, which enables the utilization of a wide range of well-developed analysis methods (Orth et al., 2010).

The increasing ubiquity of genomic sequencing has led to a rapid expansion

in the number and complexity of genome-scale metabolic models, e.g. the human metabolic model that has more than 80,000 reactions (Thiele et al., 2020). Recent automated reconstruction tools can generate models spanning the entire primary metabolism of both pro- and eukaryotes (Machado et al., 2018). Consequently, modern metabolic models are considerably larger in scale than their predecessors, which is further compounded by the construction of multi-member community models. The increasing size of models (see Figure S1) drives the need to develop analysis algorithms that can accommodate this complexity. While the underlying optimization problems are hard to accelerate and parallelize, many analysis types can be computed concurrently. However, despite continued efforts (Heirendt, Thiele, et al., 2017), this remains challenging due to the inherent scalability limits of existing software implementations.

Here, we present COBREXA.jl, a package for running exascale COBRA workflows. The package is implemented in the Julia programming language (Bezanson et al., 2017), which allows for facile extension with user-defined numeric-computing routines, and interoperability with other high performance computing packages. It provides a 'batteries-included' solution for scaling analyses to high-performance computing (HPC) facilities, giving researchers a powerful toolkit for executing complicated high-volume workflows, such as the creation of digital metabolic twins in personalized medicine (Björnsson et al., 2020), and analysis of extensive communities in ecology and biotechnology. We report the implementation architecture, and substantiate how the design accommodates future extensions and scaling of common analysis tasks.

2 Implementation and results

COBREXA.jl is designed to benefit from modern programming paradigms, increasing the composability and extensibility of the implemented functions. The data structures used for model storage are replaceable by custom representations, resulting in high usage flexibility. Analysis methods support massive parallelization by default, using a previously developed framework (Kratochvíl et al., 2020). Efficient automation is provided for common tasks, e.g. creating and scrutinizing many variants of existing models, reducing the requirement for user expertise in parallel programming. A detailed architecture overview is provided in the Supplementary.

COBREXA.jl also supports the most widely used analysis methods in the field, including flux balance analysis, flux variability analysis, and flux sampling

(Gudmundsson et al., 2010). A variety of model formats are supported, including sparse-matrix-based models, as used e.g. by Heirendt, Arreckx, et al., 2019; object-oriented models, as used in COBRApy by Ebrahim et al., 2013; as well as models stored in SBML format (Keating et al., 2020). Multiple functions are provided for modifying the models, e.g. simulating the gene knockouts, and for orchestrating parallel analysis of many modified models on a cluster of computing nodes.

We benchmarked the efficiency of the distributed analysis as follows: a community model based on the human gut microbiome (Magnúsdóttir et al., 2017) was iteratively modified to simulate the impact of introducing engineered microbes into the microbiome. All constructed variants of the model were evaluated with a simple analysis routine (see Supplementary for details). The optimizations improved scalability of the constructed workflows by minimizing the communication overhead, mainly by removing the need to transfer bulk model data. We show the impact in Figure 1, where we compare the performance with naive vs. optimized workload distribution. The results suggest that the optimized approach will be able to support exascale workloads (see Supplementary), as well as massive application of existing methodologies.

3 Conclusion

COBREXA.jl is a new package developed for exascale-ready processing of constraint-based biological models. It differs from the other implementations of COBRA methods (Heirendt, Arreckx, et al., 2019; Ebrahim et al., 2013; Gelius-Dietrich et al., 2013), by focusing on computational efficiency. This is required for performing analyses of extremely large models and model variants, future-proof extensibility, and workload distribution that minimizes the computation time and resource usage on HPC infrastructures. This enables processing and analysis of datasets that pose challenges for the currently available tools, such as the comprehensive human gut microbiome models.

Funding

The research leading to these results has received funding from the European Union's Horizon 2020 Programme under the PerMedCoE Project (www.permedcoe.eu), grant agreement n° 951773. This work was also partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)

under Germany's Excellence Strategy–EXC-2048/1–project ID 390686111 and EUs Horizon 2020 research and innovation program under the Grant Agreement 862087. The presented experiments were carried out using the HPC facilities of the University of Luxembourg (see `https://hpc.uni.lu`).

References

Apaolaza, I. et al. (2018). COBRA methods and metabolic drug targets in cancer. *Mol & Cell Oncol* 5.1, e1389672.

Bezanson, J. et al. (2017). Julia: A fresh approach to numerical computing. *SIAM Rev* 59.1, pp. 65–98.

Björnsson, B. et al. (2020). Digital twins to personalize medicine. *Genome Med* 12.1, pp. 1–4.

Brunk, E. et al. (2018). Recon3D enables a three-dimensional view of gene variation in human metabolism. *Nat Biotechnol* 36.3, p. 272.

Cook, D. J. and J. Nielsen (2017). Genome-scale metabolic models applied to human health and disease. *Wires Syst Biol Med* 9.6, e1393.

Ebrahim, A. et al. (2013). COBRApy: COnstraints-Based Reconstruction and Analysis for Python. *BMC Syst Biol* 7.1, pp. 1–6.

Fang, X. et al. (2020). Reconstructing organisms in silico: genome-scale models and their emerging applications. *Nat Rev Microbiol* 18.12, pp. 731–743.

Gelius-Dietrich, G. et al. (2013). Sybil: efficient constraint-based modelling in R. *BMC Syst biol* 7.1, pp. 1–8.

Gudmundsson, S. and I. Thiele (2010). Computationally efficient flux variability analysis. *BMC Bioinformatics* 11.1, pp. 1–3.

Heirendt, L., S. Arreckx, et al. (2019). Creation and analysis of biochemical constraint-based models using the COBRA Toolbox v.3.0. *Nat Protoc* 14.3, pp. 639–702.

Heirendt, L., I. Thiele, et al. (2017). DistributedFBA.jl: high-level, high-performance flux balance analysis in Julia. *Bioinformatics* 33.9, pp. 1421–1423.

Keating, S. M. et al. (2020). SBML Level 3: an extensible format for the exchange and reuse of biological models. *Mol Syst Biol* 16.8, e9110.

Kratochvíl, M. et al. (2020). GigaSOM.jl: High-performance clustering and visualization of huge cytometry datasets. *GigaScience* 9.11, giaa127.

Machado, D. et al. (2018). Fast automated reconstruction of genome-scale metabolic models for microbial species and communities. *Nucleic Acids Res* 46.15, pp. 7542–7553.

Magnúsdóttir, S. et al. (2017). Generation of genome-scale metabolic reconstructions for 773 members of the human gut microbiota. *Nat Biotechnol* 35.1, pp. 81–89.

Orth, J. D. et al. (2010). What is flux balance analysis? *Nat Biotechnol* 28.3, pp. 245–248.

Thiele, I. et al. (2020). Personalized whole-body models integrate metabolism, physiology, and the gut microbiome. *Mol Syst Biol* 16.5, e8982.

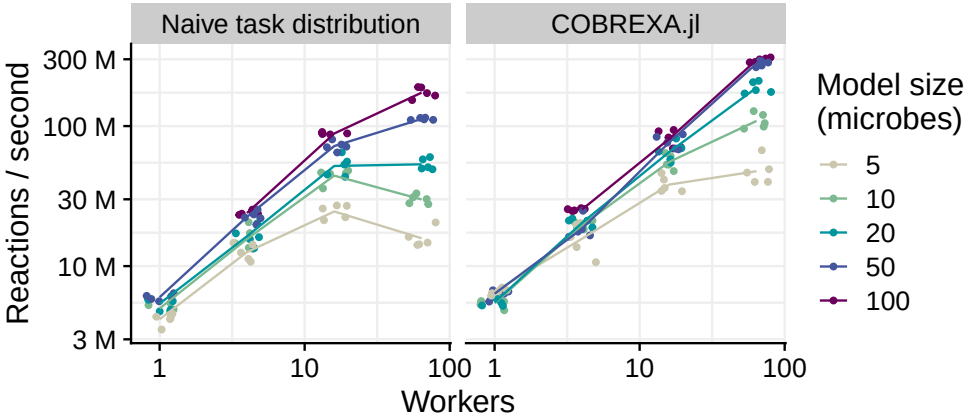

Figure 1: Performance of a simple analysis executed over a large number of variants of a community model of AGORA microbes shows efficiency of the workload distribution; each dot represents a benchmark run, lines connect mean values for benchmark groups. The improvements in COBREXA.jl (right) increase the speed and scalability.