

SOFTWARE

Open Access



CoGAPS 3: Bayesian non-negative matrix factorization for single-cell analysis with asynchronous updates and sparse data structures

Thomas D. Sherman¹, Tiger Gao² and Elana J. Fertig^{1,3,4*} 

*Correspondence: ejfertig@jhmi.edu
¹ Department of Oncology, Johns Hopkins University School of Medicine, Baltimore, MD, USA
Full list of author information is available at the end of the article

Abstract

Background: Bayesian factorization methods, including Coordinated Gene Activity in Pattern Sets (CoGAPS), are emerging as powerful analysis tools for single cell data. However, these methods have greater computational costs than their gradient-based counterparts. These costs are often prohibitive for analysis of large single-cell datasets. Many such methods can be run in parallel which enables this limitation to be overcome by running on more powerful hardware. However, the constraints imposed by the prior distributions in CoGAPS limit the applicability of parallelization methods to enhance computational efficiency for single-cell analysis.

Results: We developed a new software framework for parallel matrix factorization in Version 3 of the CoGAPS R/Bioconductor package to overcome the computational limitations of Bayesian matrix factorization for single cell data analysis. This parallelization framework provides asynchronous updates for sequential updating steps of the algorithm to enhance computational efficiency. These algorithmic advances were coupled with new software architecture and sparse data structures to reduce the memory overhead for single-cell data.

Conclusions: Altogether our new software enhance the efficiency of the CoGAPS Bayesian matrix factorization algorithm so that it can analyze 1000 times more cells, enabling factorization of large single-cell data sets.

Keywords: Single cell, Matrix factorization, Pattern detection, Unsupervised learning

Background

Non-negative matrix factorization (NMF) techniques have emerged as powerful tools to identify the cellular and molecular features that are associated with distinct biological processes from single cell data [3–5, 7, 14, 16]. Bayesian factorization approaches can mitigate local optima and leverage prior distributions to encode biological structure in the features [9, 12]. However, the computational cost of implementing these approaches may be prohibitive for large single cell datasets. Many NMF methods can be run in



© The Author(s) 2020. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>. The Creative Commons Public Domain Dedication waiver (<http://creativecommons.org/publicdomain/zero/1.0/>) applies to the data made available in this article, unless otherwise stated in a credit line to the data.

parallel, and thereby leverage the increasing availability of suitable hardware to scale for analysis of large single cell datasets [1, 8, 10].

Previously, we developed CoGAPS as a sparse, Bayesian NMF approach for bulk [6, 9, 13] and single-cell genomics analysis [3, 11]. Comparison studies to gradient-based NMF [9, 11] and autoencoders [11] demonstrated the unique robustness of this approach to initialization and its inference of dynamic biological processes in bulk and single cell datasets. Further comparison of this approach to principal component analysis and independent component analysis demonstrated the unique ability of this approach to infer transcriptional signatures unique to specific individuals and tissues in GTEx [12]. CoGAPS was designed to perform Gibbs sampling for a unique prior distribution that adapts the level of sparsity to the distribution of expression values in each gene and cell. While this design allows CoGAPS to adapt to different types of data, it also imposes a constraint on the algorithm that requires the update steps to be proposed sequentially. While the sequential updates of CoGAPS limit implementation of embarrassingly parallel computational approaches, we present a new method for isolating the sequential portion of CoGAPS so that the majority of the algorithm can be run in parallel. Additionally, we derive an optimization for sparse data in order to take advantage of the nature of many single-cell data sets. In combination, these new features in CoGAPS version 3.2 allows for efficient Bayesian NMF analysis of large single cell data sets.

Implementation

The CoGAPS algorithm

The input for CoGAPS is a data matrix of single-cell data with N measures (e.g., genes, genomic coordinates, proteins) and M cells, $D \in \mathbb{R}^{N \times M}$, and a number of patterns (features) to learn, K . It factors D into two lower dimensional matrices, $A \in \mathbb{R}^{N \times K}$ and $P \in \mathbb{R}^{K \times M}$. The columns of the A matrix contains relative weights of each measurement for the learned features and the corresponding rows of the P matrix contains the relative expression of those features in each cell [12]. CoGAPS assumes the elements of D are normally distributed with mean AP and variance proportional to D . The algorithm has a Gamma prior on each element of A and P whose shape hyper parameter has a Poisson prior. This model encodes a sparsity constraint that adapts to the relative sparsity of each gene or cell in the data [11]. In version 3.2 of CoGAPS, the input matrix D can now be passed as either a data matrix or a Single Cell Experiment object. The output for A and P is stored in a Linear Embedding Matrix to enable compatibility with emerging single-cell workflows Bioconductor.

Asynchronous updates

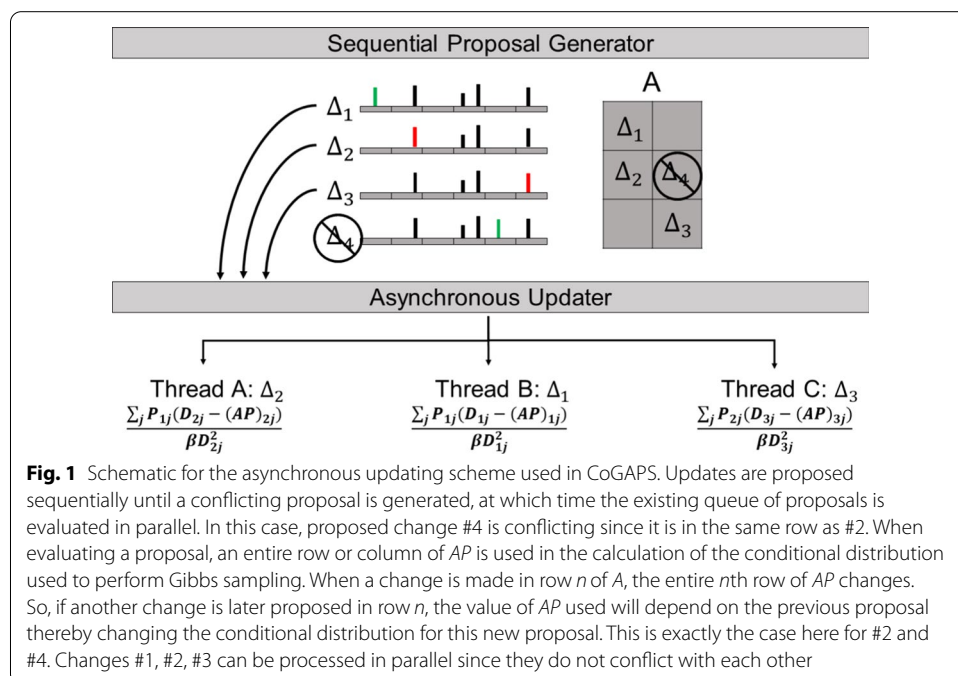
Although the algorithm that determines the order of the matrix updates in CoGAPS must be run sequentially, the large number of measurements in genomics data provide feasibility for running the most computationally intensive portion of the algorithm in parallel. Notably, the proposal for which matrix element to update can be made efficiently, whereas evaluating the new value for that element requires an expensive calculation across a row or column of the data. We take advantage of this fact with an asynchronous updating scheme that yields a Markov chain that is equivalent to the one obtained from the standard sequential algorithm [6]. In order to do

this, we build up a queue of proposals using the sequential algorithm until a conflicting proposal is generated, at which point we evaluate the entire queue of proposals in parallel.

The asynchronous updating scheme heavily relies on the designation of conflicting, or dependent, proposals. Specifically, if two proposals are independent, they must be able to be evaluated in any order without one impacting the sampling distribution of the other. This allows a queue of independent proposals to be evaluated in parallel and still produce a deterministic result. One example of dependent proposals is given here (Fig. 1) and a full accounting of all possible conflicts can be found in Additional file 1.

Sparse data structures

Single-cell data tends to be sparse. Therefore, the natural solution for reducing memory overhead is to use sparse data structures to represent the data D in the analysis. While the data, D , may be very sparse, the weights in A and P correct for dropout and therefore have a product that may be largely non-zero. Traditionally, CoGAPS caches this product to reduce the number of calculations at each step. However, caching AP introduces an unacceptable memory overhead when the data is stored in a sparse format. To address this, we separate the matrix calculations into terms that can be efficiently calculated using only the non-zero entries of D and terms that can be precomputed before each batch of updates as described in detail in Additional file 1. By doing this, we can make the computation time proportional to the sparsity of the data. However, since storing the data in a sparse format requires the calculation of AP during the update steps, there is a performance trade-off that needs to be considered. Typically, when the data is more than 80% sparse it is more efficient to use the sparse optimization, even though it requires calculating AP .



Results

We simulated three sparse single-cell datasets with the R package Splatter [15]. We varied the level of sparsity in each data set and tested the amount of memory used with and without the sparse optimization enabled. We also measured the run time in both the single-threaded and multi-threaded case. Table 1 gives a high-level overview of the performance differences. For example, with 2000 genes and 2000 cells when the data is 90% zeroes, using the sparse optimization and 4 threads will give identical results to the standard algorithm in 1/5 of the time while using 1/25 of the memory.

We also tested the performance on a single-cell 10X data set generated at the Broad Institute [2]. We ran a benchmark on a subset of 37,000 human immune cells (umbilical cord blood) and only kept the 3000 highest variance genes. In this case, enabling the sparse optimization reduced memory overhead by 82% and cut the run time by 74%. When using 4 threads instead of 1, the run time was cut by an additional 36%. We further ran this benchmark on random subsets of this data to benchmark performance as a function of number of genes and number of cells (Fig. 2). We observe that the running time in these simulations increases linearly with both the number of cells and the number of genes as expected.

Conclusions

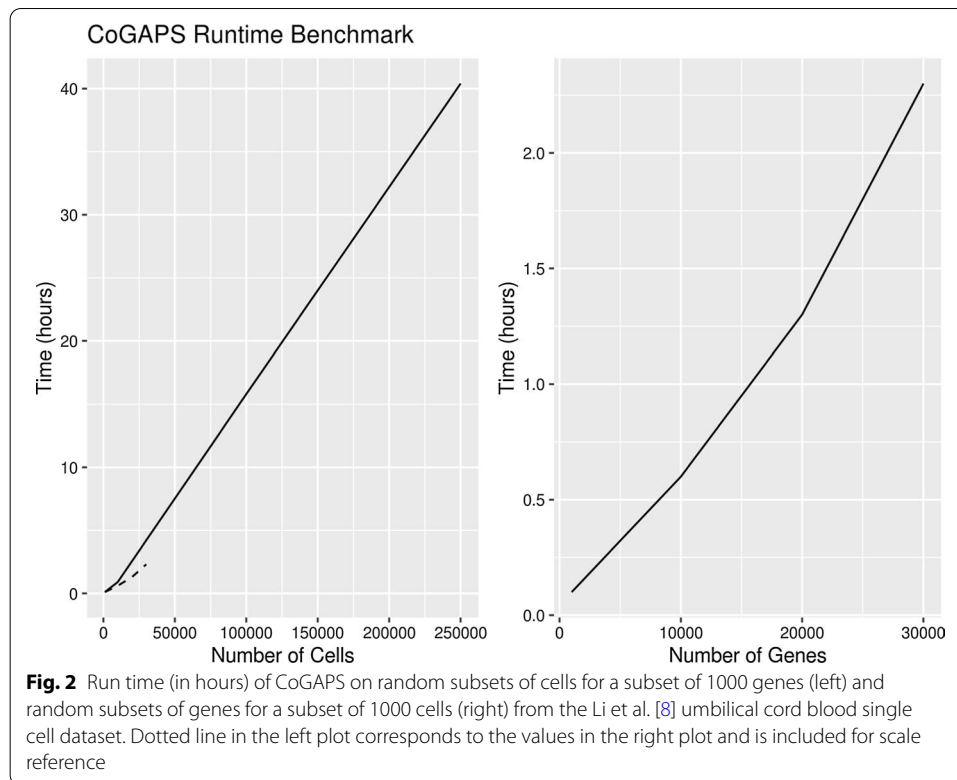
In this paper, we present an algorithm and software to enable parallelization of CoGAPS to enable analysis of large single cell datasets. This parallelization was done by combining existing methods for Gibbs sampling [1, 8, 10] with a new infrastructure for the updating steps in CoGAPS. Prior to the implementation of an asynchronous updating scheme, CoGAPS was applied to large data sets by using a distributed version of the algorithm, GWCoGAPS, that performed analysis across random sets of genes [13] or random sets of cells [11]. This distributed version leveraged the observation that the learned values of *A* and *P* are robust across these random sets. Future work combining the asynchronous and distributed parallelization methods will be critical to further enhance performance by utilizing all CPU cores efficiently.

Availability and requirements

- Project name: CoGAPS
- Project home page: <https://doi.org/doi:10.18129/B9.bioc.CoGAPS>
- Operating systems: Platform independent
- Programming languages: R and C++

Table 1 Relative performance of the sparse optimization on 2000 genes and 2000 cells, baseline is the standard algorithm with 1 thread and no sparse optimization

Data sparsity (%)	Memory (MB)	Runtime (1 thread)	Runtime (4 threads)
70	0.14	1.92	0.62
80	0.09	1.24	0.42
90	0.04	0.42	0.20



Other requirements: R version 3.6 or higher

Any restrictions to use by non-academics: None

Supplementary information

Supplementary information accompanies this paper at <https://doi.org/10.1186/s12859-020-03796-9>.

Additional file 1: Extended methods describing the asynchronous update algorithm to enhance parallelization and efficiency for the Bayesian NMF in Version 3.0.

Abbreviations

CoGAPS: Coordinated Gene Activity in Pattern Sets; CPU: Central processing unit; NMF: Non-negative matrix factorization.

Acknowledgements

The authors thank Genevieve Stein-O'Brien, Emily Davis-Marcisak, Loyal Goff, and Ted Liefeld for helpful discussions and benchmarking algorithm performance.

Authors' contributions

TDS and EJF designed the study and algorithms. TDS and TG implemented all algorithms and developed all software. EJF oversaw the study. TDS, TG, and EJF wrote and edited the manuscript. All authors have read and approved the manuscript.

Funding

This work was supported by grants from the NIH (NCI R01CA177669, U01CA196390, U01 CA212007, P30 CA006973, and a Pilot Project from P50 CA062924; NIDCR R01 DE027809), the Chan-Zuckerberg Initiative DAF (2018-183444) an advised fund of the Silicon Valley Community Foundation, the Johns Hopkins University Catalyst and Discovery awards, the Johns Hopkins University School of Medicine Synergy Award, the Lustgarten Foundation, and the Allegheny Health Network-Johns Hopkins Cancer Research Fund. The funders provided salary support and compute costs for the experiments, and did not influence the design or analyses performed in this study.

Availability of data and materials

The datasets analyzed during the current study are available from Bo Li et al. Census of Immune Cells. Broad Inst. Mass. Inst. Technol. Howard Hughes Med. Inst. Retrieved from <https://data.humancellatlas.org/explore/projects/cc95ff89-2e68-4a08-a234-480eca21ce79>.

Ethics approval and consent to participate

This research involved neither human nor animal subjects.

Consent for publication

Not applicable.

Competing interests

The authors declare they have no competing interests.

Author details

¹ Department of Oncology, Johns Hopkins University School of Medicine, Baltimore, MD, USA. ² Department of Computer Science, Johns Hopkins University, Baltimore, MD, USA. ³ Department of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, MD, USA. ⁴ Department of Biomedical Engineering, Johns Hopkins University School of Medicine, Baltimore, MD, USA.

Received: 30 November 2019 Accepted: 1 October 2020

Published online: 14 October 2020

References

1. Ahn S, et al. Large-scale distributed Bayesian matrix factorization using stochastic gradient MCMC. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining—KDD'15. Sydney: ACM Press; 2015. p. 9–18.
2. Bo Li, et al. Census of immune cells. Broad Inst. Mass. Inst. Technol. Howard Hughes Med. Inst. <https://data.human-cellatlas.org/explore/projects/cc95ff89-2e68-4a08-a234-480eca21ce79>. Accessed 2019
3. Clark BS, et al. Single-cell RNA-seq analysis of retinal development identifies NFI factors as regulating mitotic exit and late-born cell specification. *Neuron*. 2019;102:1111–1126.e5.
4. Cleary B, et al. Efficient generation of transcriptomic profiles by random composite measurements. *Cell*. 2017;171:1424–1436.e18.
5. Duren Z, et al. Integrative analysis of single-cell genomics data by coupled nonnegative matrix factorizations. *Proc Natl Acad Sci*. 2018;115:7723–8.
6. Fertig EJ, et al. CoGAPS: an R/C++ package to identify patterns and biological process activity in transcriptomic data. *Bioinform Oxf Engl*. 2010;26:2792–3.
7. Kotliar D, et al. Identifying gene expression programs of cell-type identity and cellular activity with single-cell RNA-Seq. *eLife*. 2019;8:e43803.
8. Li F, et al. A fast distributed stochastic gradient descent algorithm for matrix factorization. In: *JMLR: workshop and conference proceedings*. 2014;36:77–87.
9. Ochs MF, Fertig EJ. Matrix factorization for transcriptional regulatory network inference. In: *2012 IEEE symposium on computational intelligence in bioinformatics and computational biology*; 2012. p. 387–96.
10. Schmidt MN, et al. Bayesian non-negative matrix factorization. In: Adali T, et al., editors. *Independent component analysis and signal separation*. Lecture notes in computer science. Berlin: Springer; 2009. p. 540–7.
11. Stein-O'Brien GL, et al. Decomposing cell identity for transfer learning across cellular measurements, platforms, tissues, and species. *Cell Syst*. 2019;8:395–411.e8.
12. Stein-O'Brien GL, et al. Enter the matrix: factorization uncovers knowledge from omics. *Trends Genet*. 2018;34:790–805.
13. Stein-O'Brien GL, et al. PatternMarkers & GWCoGAPS for novel data-driven biomarkers via whole transcriptome NMF. *Bioinform Oxf Engl*. 2017;33:1892–4.
14. Welch JD, et al. Single-cell multi-omic integration compares and contrasts features of brain cell identity. *Cell*. 2019;177:1873–1887.e17.
15. Zappia L, et al. Splatter: simulation of single-cell RNA sequencing data. *Genome Biol*. 2017;18:174.
16. Zhu X, et al. Detecting heterogeneity in single-cell RNA-Seq data by non-negative matrix factorization. *PeerJ*. 2017;5:e2888.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.