RESEARCH

BioNetGMMFit: a Parameter Estimation Tool for BioNetGen using Single-Cell Snapshot Data from Cell Populations Evolving over Time

John Wu[1,4*], William CL Stewart[2], Ciriyam Jayaprakash[3] and Jayajit Das[4,5]

[*]Correspondence: wu.4427@osu.edu
[1]Department of Computer Science, The Ohio State University, Columbus, US
Full list of author information is available at the end of the article

Abstract

Background: Mechanistic models are commonly employed to describe signaling and gene regulatory kinetics in single cells and cell populations. Recent advances in single-cell technologies have produced multidimensional datasets where snapshots of copy numbers (or abundances) of a large number of proteins and mRNA are measured across time in single cells. The availability of such datasets presents an attractive scenario where mechanistic models are validated against experiments, and estimated model parameters enable quantitative predictions of signaling or gene regulatory kinetics. To empower the systems biology community to easily estimate parameters accurately from multidimensional single-cell data, we have merged a widely used rule-based modeling software package BioNetGen, which provides a user-friendly way to code for mechanistic models describing biochemical reactions, and the recently introduced CyGMM, that uses cell-to-cell differences to improve parameter estimation for such networks, into a single software package: BioNetGMMFit.

Results: BioNetGMMFit provides parameter estimates of the model, supplied by the user in the BioNetGen markup language (BNGL), which yield the best fit for the observed single-cell, timestamped data of cellular components. Furthermore, for more precise estimates, our software generates confidence intervals around each model parameter. BioNetGMMFit is capable of fitting datasets of increasing cell population sizes for any mechanistic model specified in the BioNetGen markup language.

Conclusion: By streamlining the process of developing mechanistic models for large single-cell datasets, BioNetGMMFit provides an easily-accessible modeling framework designed for scale and the broader biochemical signaling community.

Keywords: Mechanistic Modeling; BioNetGen; Parameter Estimation; Rule Based Modeling; Generalized Method of Moments

Background

Development of rule-based modeling approaches such as BioNetGen (BNGL) [1] and libRoadRunner [2] provide a user-friendly way to address combinatorial complexity in modeling biochemical reactions in signaling and gene regulatory networks [3, 4]. Estimation of model parameters (e.g., reaction rates) in rule-based models using experimental measurements is important for improving model predictions and quantifying underlying mechanisms described by the model. Software packages such as PyBioNetFit [5] have made it possible to estimate parameters from rule-based

models using bulk measurements for selected proteins (e.g. average protein abundances observed over time).

Recent advances in new single-cell technologies capture cell-to-cell differences [6, 7, 8, 9, 10] in protein/mRNA abundances that allow for the evaluation of higher-order moments which can be utilized to improve the estimation of model parameters. Generalized Method of Moments (GMM), a method used widely in econometrics [11, 12], provides a systematic approach to include means and higher moments in the parameter estimation problem [13, 14]. The recently proposed method CyGMM [14] uses GMM in conjunction with Particle Swarm Optimization (PSO) [15] to estimate model parameters from single-cell cytometry data. Specifically, CyGMM finds the reaction rates that minimize the difference between the predicted and observed moments. This partnership of methodology proved effective in producing unbiased parameter estimates in mechanistic or ODE models [14].

Here we present a software tool BioNetGMMFit which enables parameter estimation in BioNetGen models using single-cell snapshot data with GMM. To use the tool, users must supply a BioNetGen model file, time-stamped snapshot protein abundance data files, and run configuration files. Based on the analysis of time-stamped protein abundances, BioNetGMMFit estimates parameters of the BioNetGen model, provides confidence intervals, predicts moments at future times, and reports a goodness of fit statistic (i.e the cost function), which is the distance between the observed moments in the data and the predicted moments, computed by the model. We include the ability to tune the built-in PSO hyperparameters for optimal performance and precision of estimates. Furthermore, being parallelized and written in C++, BioNetGMMFit offers ease of scalability with increasing data sizes on high-performance computing clusters. As C++ compilation is nontrivial across different operating systems, BioNetGMMFit is readily available through Docker.

Implementation

Currently, we have made BioNetGMMFit available as both a web user interface (via Docker), and as a collection of C++ source files. In Figure 1, BioNetGMMFit reads specifically (A) a directory containing a CSV file with initial protein abundances (X) and a directory containing CSV files with observed protein abundances at different time snapshots (Y); (B) a BNGL file describing the mechanistic model capable of executing deterministic or stochastic kinetics; (C) a time steps CSV file that describes the time steps of the intervals of evolution in the mechanistic model; (D) and a BioNetGMMFit hyperparameter configuration file that allows for various features from simulating evolution data to holding an unknown parameter constant in the mechanistic model. A general outline of BioNetGMMFit is shown in Figure 1. More documentation on how to use BioNetGMMFit can be found on the GitHub Repo page (https://github.com/jhnwu3/BioNetGMMFit).

In the case of directories of data files, only one X CSV data file is read from directory X for the initial set of protein abundances while the Y directory can contain multiple CSV files for each time step being analyzed. Data files are formatted such that each measured cell is a row and each column is a type of protein abundance measured from the cell. As BioNetGMMFit assumes protein abundances or concentrations are non-negative, any cell data with negative protein values for a given time is deleted before parameter estimation is performed.

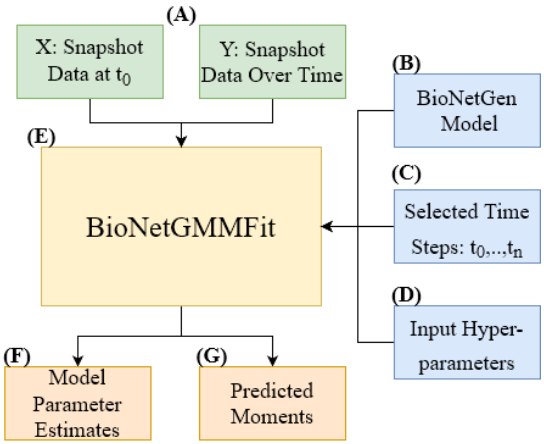

Figure 1: **BioNetGMMFit Work Flow:** (A) is the snapshot data. (B) is the defined BioNetGen model file. (C) are the selected time steps of the Snapshot data in (A). (D) is the hyperparameters configuration file that contains all hyperparameters used by the particle swarm optimizer for parameter estimation. (F) Parameter estimates (i.e values and confidence intervals) and (G) predicted moments from the model are the respective outputs of BioNetGMMFit.

As an example, we will define a signaling kinetic model involving four proteins to analyze a CD8+ T cell cytometry dataset. For each protein, its three trajectories: means, variances, and covariances, are shown across time (see Figure 2). To make the presentation simple, we will only analyze trajectories at two time points: 1 minute and 2 minute. Again, as shown in Figure 2B, mechanistic model reaction rules are defined in the BNGL where they are then converted into an SBML file using BioNetGen's writeSBML() function. Using libRoadRunner's C++ API [2], BioNetGMMFit converts this SBML file into a roadrunner object that is capable of simulating the defined model to user-specified time intervals.

Once the model is defined, the hyperparameters configuration file supplies the PSO hyperparameters used for parameter estimation. For more information on the configuration file, a table of parameters is provided on the documentation within the GitHub page. To estimate parameters within the BioNetGen model, PSO minimizes the GMM-weighted square euclidean distances of lower and higher order moments between the observed and model-simulated data [14]. Upon PSO convergence, estimates are provided and confidence intervals are computed.

BioNetGMMFit also plots confidence intervals as well as the observed and estimated moments as shown in Figure 2. The underlying datasets used for the generation of the CD8+ T Cells are from (https://dpeerlab.github.io/dpeerlab-website/dremi-data.html) [16].

Results

By combining the best aspects of CyGMM and BioNetGen, our new software: BioNetGMMFit provides several notable features for the analysis of single-cell timestamped cytometry data, especially when compared to existing software such as

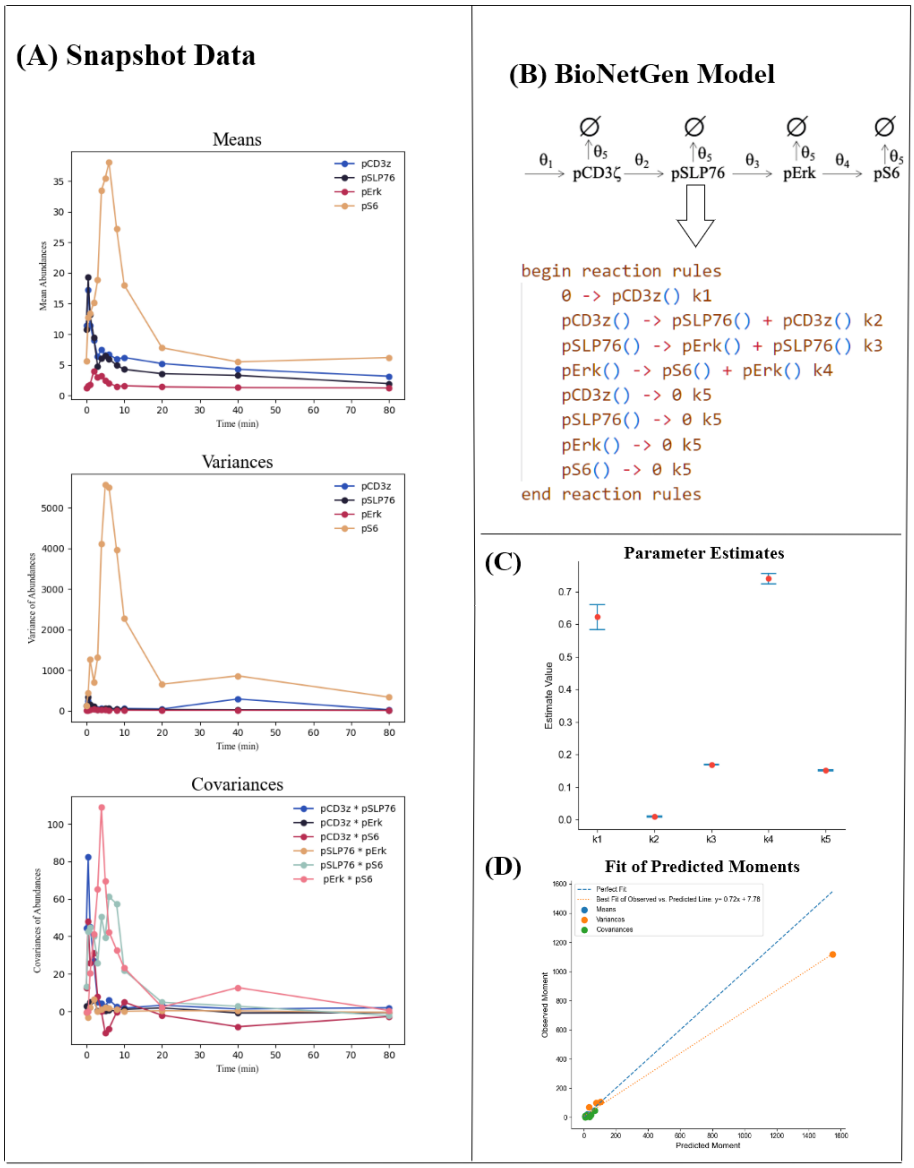

Figure 2: **Key Inputs and Outputs of BioNetGMMFit:** (A), (B), (C), and (D) are directly related to Figure 1. Panel A in Figure 2 shows the sample means, variances, and covariances of CD8+ proteins for the observed time points that can be inputted into BioNetGMMFit as indicated in Box (C) in Figure 1. Panel B in Figure 2 shows the ruled-based model and its corresponding BioNetGen representation, which is indicated in Figure 1 Box (B). Panels C and D in Figure 2, shows the parameter estimates and predicted moments, which correspond to Figure 1's Boxes F and G, respectively.

PyBioNetFit [5] and Statistical Model Checking (SMC) [17]. In particular, BioNetGMMFit offers the ability to analyze single-cell trajectories for a large number (> 1000) of cells, predict moments at future times, and simulate data with ground

truth parameters. We detail the full capabilities and limitations of BioNetGMMFit below.

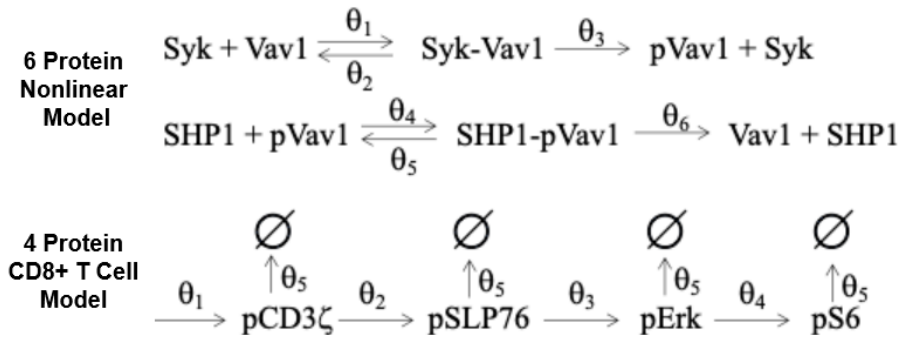

Figure 3: **Schematic Diagrams of CD8+ T Cell and Nonlinear 6 Protein Reaction Models** The CD8+ T cell model was used in the implementation section. The 6 protein nonlinear model schematic was used to simulate the moments seen in the results section.

Capabilities

Accurately Fitting Across a Large Number of Cells

BioNetGMMFit accurately and efficiently estimates BNGL model parameters from cytometry data observed in single-cell trajectories. Furthermore, it uses the same sophisticated minimization procedure as CyGMM [14] to estimate parameters from the analysis of higher-order moments. By contrast, alternative methods may have difficulty estimating parameters accurately and efficiently due to their limitations.

For example, when the estimation depends on the analysis of data from a large number of single cells (as opposed to bulk measurements that have been summarized across cells), existing software is either not designed to handle such data [5, 17], or is less user-friendly, or is computationally inefficient [1]. In many cases, when attempting to perform parameter estimation across a combination of moment statistics, inefficiencies related to file input and output occur. For instance, in using BioNetGen's simulator, each cell has its own BNGL file containing a bulk measurement such as average protein abundances across time. Therefore, to perform parameter estimation across a combination of different measurements (i.e means, variances, and covariances) with BioNetGen only [1], the user must repeatedly read from multiple data files in its parameter estimation. In PyBioNetFit [5], only the mean abundances at each time point are used for parameter estimation. Our method allows users to simply input several time snapshot files containing abundance data from thousands of cells, and stores them directly in memory for quick simulations. See Table 1 below for each dataset's cell count used for parameter estimation. While the comparison is not exact as the models and objective function (i.e BioNetGMMFit's incorporation of higher-order moments) analyzed were different, we show that we can attain run time performance comparable to that of PyBioNetFit's [5]. In the analysis of the 6 protein nonlinear model in Figure 3 with 5,000 single cells, run time was approximately 1.9 hours on a 16-core processor for 30 repeated parameter estimates, which falls into the range of run time using the SBML simulator in

PyBioNetFit for a similar number of cores [5]. In comparison to SMC [17], while again the comparisons are not exact as the methods and models are different, our run times are comparable to their 4.2 hours for parameter estimation. Due to C++ parallelization with OpenMP, our software is capable of linearly scaling with increasing cell count, thereby reducing run times for large numbers of cells. Table 3 provides run-time details concerning the number of cells and model parameters.

Estimating with Higher-order Moments

Due to their emphasis on bulk measurements, existing software such as PyBioNetFit for rule-based parameter estimation are unable to fit higher moments. By contrast, BioNetGMMFit supports three levels of analysis, where each level involves an increasing number of moments. First, one can choose to estimate parameters using only the means (i.e. first moments). Second, one can choose to estimate parameters using both means and variances. Third, one can use means, variances, and covariances. Results for moments with different levels of fitting are shown with different colors in Figure 2.

BioNetGen Model Analysis and Hyperparameter Tuning Using In Silico Models

Another feature of BioNetGMMFit is the ability to simulate deterministic and stochastic kinetics in BNGL models using a user-defined set of ground truth parameters. This feature is enabled in the hyperparameters file. More documentation can be found on the GitHub page. Referring back to the workflow diagram in Figure 1, our program uses the time steps file to compute newly evolved conditions (Y) based on a set of ground truth parameters and corresponding initial conditions. Once the protein abundances have been evolved, BioNetGMMFit proceeds with parameter estimation (i.e. the selected rule-based model is fit to the simulated cytometry data). For instance, one can choose to simulate a nonlinear six-protein and six-parameter reaction model, which was previously analyzed in [14] and shown in Figure 3, and get parameter estimations and predicted moments shown in Figure 4. Immediately following parameter estimation, the predicted first, second, and cross moments are plotted against the simulated observed moments at different time points as shown in Figure 4. The three slopes shown in Figure 4, each near unity, indicate good agreement between the predicted moments (i.e. means, variances, and covariances) and the observed moments at each time point. Furthermore, using the model parameters estimated by BioNetGMMFit, we can *forecast* moments (i.e. predict moments) at times well outside of the given observational range (Figure 4).

Another important utility of being able to simulate models with known ground truth parameters is hyperparameter tuning. In the case of PSO, there are two main hyperparameters to tune, the number of particles and the number of steps. By having a simulated reference point of the "optimal" or ground truth parameters for your BioNetGen model, we can provide insights into the performance of a hyperparameter configuration. For instance, in Figure 4, parameter estimates are improved by increasing the number of particles and steps in the hyperparameter configuration.

Limitations

BioNetGMMFit currently has several limitations pertaining to our GMM based method and software implementation.

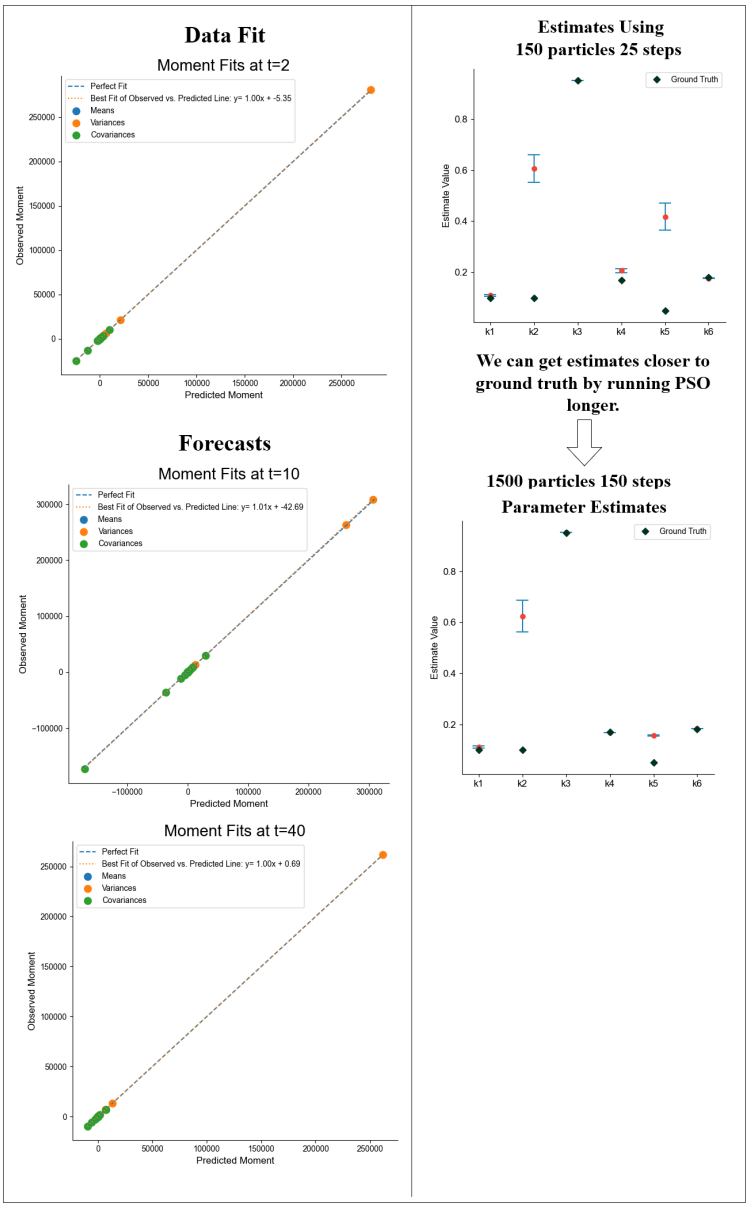

Figure 4: **Simulated 6 Protein Model Prediction:** The moment fits on the left are the predicted moments plotted against the observed moments at various time points using only the parameter estimates computed from data observed up to t=2 in the simulated nonlinear 6 protein model. More specifically, parameter estimates were derived from configuration 6 Protein A. Please see Table 2 for more information on the PSO hyperparameter configurations used. The bottom two graphs are the forecasted moment fits at t=10 and t=40. On the right, we have parameter estimates for two different hyperparameter configurations. Note that the time units for the simulated dataset are arbitrary.

Not All Mechanistic Models Have Convex Cost Functions

GMM cost functions for ODE models can possess multiple local minima or extremely flat curvatures [14] that make the estimation of optimal parameter values

challenging. However, this method does not fully address the difficulties of such irregular cost functions. Although PSO mitigates these optimization problems, depending on the computational constraints, BioNetGMMFit can still fall within local minima. One such example is shown in Figure 4 where only 150 particles and 25 steps were used for parameter estimation of the simulated 6 protein model to represent a computationally constrained environment. However, note that despite the parameter estimates being unable to capture the ground truth parameters, estimates were still able to effectively predict the observed first, second, and cross moments as shown in Figure 4. Furthermore, should one care about the best possible parameter estimates, as shown with k_4 in Figure 4, by using ten times the number of particles and steps in PSO, parameter estimates become much closer to the ground truth parameters. However, this slight improvement in estimation demands higher computational costs as shown in Table 3 where there is a drastic increase in run time from configuration "6 Protein Time Points Set A" to "6 Protein Time Points Set A Extreme" in Table 3. Despite improvements in parameter estimates, there is no guarantee that using more particles to search the model parameter space will necessarily lead to the discovery of the ground truth parameters in mechanistic models.

Limitations With Software

BioNetGMMFit is currently limited to parameter estimation for well-mixed mechanistic models describing deterministic and stochastic kinetics. Unlike PyBioNetFit [5], other forms of simulation such as NFsim [18] and spatial modeling [19, 20] are not supported. Furthermore, BioNetGMMFit does not support any optimization algorithms outside of PSO. Additionally, the lack of Python support may make it difficult to integrate it with pre-existing workflows that rely heavily on Python. As compilation with libRoadRunner C++ libraries is a nontrivial process, BioNetGMMFit is most readily accessible through Docker, which although accessible across all major operating systems, still requires the use of third-party virtualization software. These software weaknesses will need to be addressed in future iterations of the software.

Computational Costs

Run times may vary drastically depending on the time points, models, and numbers of single cells being analyzed. To provide some idea of computational costs, we profiled the CD8+ T cell signaling model and the nonlinear signaling model shown in Figure 3. The number of model parameters, PSO hyperparameters, and dataset characteristics are shown in Tables 2 and 1 while their respective run times are shown in Table 3. Please note that each run is considered a PSO estimate. To generate confidence intervals, more than one run is required while generally 30 or more runs are preferred. All run times are rounded to the nearest second and all runs were performed using a 16-Core AMD EPYC 7302 processor with 32 threads and a maximum clock speed of 3.3 GHz. Only 30 threads were used for testing.

Although their asymptotic relationship may differ depending on the BioNetGen model, increasing the particle swarm's number of particles and step count generally has a linear effect on run time where a 10× in particles tends to lead to a 10× in run

Hyperparameter	CD8+ T Cell Model	6 Protein Time Points Set A	6 Protein Time Points Set B	6 Protein Time Points Set C	6 Protein Time Set A Extreme
# of single cells Used for Estimation	719	5,000	5,000	5,000	5000
# of single cells Observed	653	5,000	5,000	5,000	5000
Time Steps	1.0, 2.0	0.0, 0.5, 2.0	0.0, 0.5, 2.0, 7.0	0.0, 0.5, 2.0, 10.0, 20.0, 30.0	0.0, 0.5, 2.0

Table 1: **Datasets Used For Estimation:** The first row contain the labels of the model as well as its respective PSO hyperparameters and dataset contents used in BioNetGMMFit. Only two datasets are used, the 4 Protein CD8+ T Cell and the simulated nonlinear 6 protein dataset. Each letter "A", "B", and "C" in the "6 Protein Time Points Set" columns indicate one unique set of time points of snapshot data used in the parameter estimation for the simulated 6 protein model. The "Extreme" term in the last column represents a ten-fold multiplicative increase of particles and steps in PSO. For instance "6 Protein Time Points Set A Extreme" is the PSO hyperparameter configuration that improves the parameter estimates from "6 Protein Time Points Set A" as shown in Figure 4. Table 2 provides more information regarding the PSO hyperparameters used for parameter estimation. The following row is the number of single cells in the initial conditions (i.e first time point) that are used for simulation by the BNGL model for estimation. The next row is the number of cells observed in the following time points. Cell counts remained constant across all time points in the simulated 6 protein dataset.

Hyperparameter	CD8+ T Cell Model	6 Protein Time Points Set A	6 Protein Time Points Set B	6 Protein Time Points Set C	6 Protein Time Set A Extreme
Model Parameters	5	6	6	6	6
Particles	150	150	150	150	1500
Steps	25	25	25	25	150

Table 2: **PSO and Model Parameters:** Again, the first row indicates the model as well as its respective PSO hyperparameters and dataset contents. In this case, only two BNGL models were used, the 4 Protein CD8+ T Cell model and the simulated 6 protein nonlinear model. The "Extreme" term in the last column represents a ten-fold multiplicative increase of particles and steps in PSO. For instance "6 Protein Time Points Set A Extreme" is the PSO hyperparameter configuration that improves the parameter estimates from "6 Protein Time Points Set A" as shown in Figure 4.

Number of Runs	CD8+ T Cell Model (s)	6 Protein Time Points Set A (s)	6 Protein Time Points Set B (s)	6 Protein Time Points Set C (s)	6 Protein Time Set A Extreme (s)
1	6	230	455	1,165	10,689
5	32	1,152	2,273	5,827	53,445
30	190	6,911	13,635	34,959	320,670

Table 3: **Run Times for Different Model Configurations:** The 6 Protein Time Points Set A, B, and C columns show the change in run time with respect to the change in time steps used in Table 1. 6 Protein Extreme provides some insight into how long the run time of an extreme number of particles and steps used in PSO. One run is considered to be one PSO estimate. More than one PSO estimate is required for confidence intervals.

time. In terms of data inputs, increasing the number of cells has a linear relationship with run time. More complex models have a dramatic effect on simulation time where increasing the number of parameters and species in a system increases run time in a nonlinear fashion. Similarly, as ode-solvers must approximate solutions, increasing the time intervals tend to have a nonlinear relationship with run time. Fortunately, as BioNetGMMFit is parallelized through OpenMP, users can reduce run time by increasing the number of cores accessed by BioNetGMMFit.

Conclusions

We present a new software: BioNetGMMFit, that greatly enhances a researcher's ability to explore a wide range of mechanistic models while using higher-order moments to leverage the information in single-cell data. In particular, BioNetGMMFit provides accurate estimation of model parameters, and reports valid confidence intervals for each estimated parameter as well. Furthermore, BioNetGMMFit can easily handle data observed across multiple time points, and it can predict protein abundances at future times as well.

Appendix

Acknowledgements
Special thanks to Dr. Jim Faeder, Dr. Ali Sinan, and Dr. Lucian Smith for their advice in the development of this software.

Funding
This work is supported by the NIH awards R01-AI 143740 and R01-AI 146581 to JD, and by the Research Institute at the Nationwide Children's Hospital.

Availability of data and materials
The dataset(s) supporting the conclusions of this article is(are) available in the BioNetGMMFit/example repository, [https://github.com/jhnwu3/BioNetGMMFit/tree/main/example].

- Project name: BioNetGMMFit
- Project home page:https://github.com/jhnwu3/BioNetGMMFit
- Archived version: N/A
- Operating system(s): Platform independent through Docker
- Programming language: C/C++
- Other requirements: Docker, if Compiling see GitHub
- License: MIT License
- Any restrictions to use by non-academics: N/A

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
Every author contributed to the writing and approval of this manuscript. J.W and W.CL.S contributed to the software development and design of BioNetGMMFit.

Author details
[1]Department of Computer Science, The Ohio State University, Columbus, US. [2], GIG Statistical Consulting LLC, Columbus, US. [3]Department of Physics, The Ohio State University, Columbus, US. [4]Steve and Cindy Rasmussen Institute for Genomics, The Abigail Wexner Research Institute, Nationwide Children's Hospital, Columbus, US. [5]Departments of Pediatrics, Biomedical Informatics, Pelotonia Institute of Immuno-Oncology, College of Medicine, and Biophysics Program, The Ohio State University, Columbus, US.

References

1. Harris, L.A., Hogg, J.S., Tapia, J.-J., Sekar, J.A.P., Gupta, S., Korsunsky, I., Arora, A., Barua, D., Sheehan, R.P., Faeder, J.R.: BioNetGen 2.2: advances in rule-based modeling. Bioinformatics **32**(21), 3366–3368 (2016). doi:10.1093/bioinformatics/btw469. https://academic.oup.com/bioinformatics/article-pdf/32/21/3366/7889722/btw469.pdf
2. Somogyi, E.T., Bouteiller, J.-M., Glazier, J.A., König, M., Medley, J.K., Swat, M.H., Sauro, H.M.: libRoadRunner: a high performance SBML simulation and analysis library. Bioinformatics **31**(20), 3315–3321 (2015). doi:10.1093/bioinformatics/btv363. https://academic.oup.com/bioinformatics/article-pdf/31/20/3315/17087875/btv363.pdf
3. Goldstein, B., Faeder, J.R., Hlavacek, W.S.: Mathematical and computational models of immune-receptor signalling. Nature Reviews Immunology **4**(6), 445–456 (2004). doi:10.1038/nri1374
4. Loskot, P., Atitey, K., Mihaylova, L.: Comprehensive review of models and methods for inferences in Bio-Chemical reaction networks. Front Genet **10**, 549 (2019)
5. Mitra, E.D., Suderman, R., Colvin, J., Ionkov, A., Hu, A., Sauro, H.M., Posner, R.G., Hlavacek, W.S.: Pybionetfit and the biological property specification language. iScience **19**, 1012–1036 (2019). doi:10.1016/j.isci.2019.08.045
6. Hui, E., Cheung, J., Zhu, J., Su, X., Taylor, M.J., Wallweber, H.A., Sasmal, D.K., Huang, J., Kim, J.M., Mellman, I., *et al.*: T cell costimulatory receptor cd28 is a primary target for pd-1–mediated inhibition. Science **355**(6332), 1428–1433 (2017)
7. Milo, R., Phillips, R.: Cell Biology by the Numbers. Garland Science, ??? (2015)

8. Hsueh, R.C., Natarajan, M., Fraser, I., Pond, B., Liu, J., Mumby, S., Han, H., Jiang, L.I., Simon, M.I., Taussig, R., Sternweis, P.C.: Deciphering signaling outcomes from a system of complex networks. Science Signaling **2**(71), 22–22 (2009). doi:10.1126/scisignal.2000054. https://www.science.org/doi/pdf/10.1126/scisignal.2000054
9. Papalexi, E., Satija, R.: Single-cell rna sequencing to explore immune cell heterogeneity. Nature Reviews Immunology **18**(1), 35–45 (2018). doi:10.1038/nri.2017.76
10. Saeys, Y., Van Gassen, S., Lambrecht, B.N.: Computational flow cytometry: helping to make sense of high-dimensional immunology data. Nature Reviews Immunology **16**(7), 449–462 (2016). doi:10.1038/nri.2016.56
11. Hansen, L.P.: Large sample properties of generalized method of moments estimators. Econometrica **50**, 1029–1054 (1982)
12. Hall, A.R.: Generalized Method of Moments. OUP Oxford, ??? (2004)
13. Lück, A., Wolf, V.: Generalized method of moments for estimating parameters of stochastic reaction networks. BMC Systems Biology **10**(1), 98 (2016). doi:10.1186/s12918-016-0342-8
14. Wu, J., Stewart, W., Jayaprakash, C., Das, J.: Generalized method of moments improves parameter estimation in biochemical signaling models of time-stamped single-cell snapshot data. bioRxiv (2022). doi:10.1101/2022.03.17.484491. https://www.biorxiv.org/content/early/2022/03/19/2022.03.17.484491.full.pdf
15. Poli, R., Kennedy, J., Blackwell, T.: Particle swarm optimization. Swarm intelligence **1**(1), 33–57 (2007)
16. Krishnaswamy, S., Spitzer, M.H., Mingueneau, M., Bendall, S.C., Litvin, O., Stone, E., Pe'er, D., Nolan, G.P.: Conditional density-based analysis of t cell signaling in single-cell data. Science **346**(6213), 1250689 (2014)
17. Liu, B., Faeder, J.R.: Parameter estimation of rule-based models using statistical model checking. In: 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 1453–1459 (2016). doi:10.1109/BIBM.2016.7822738
18. Sneddon, M.W., Faeder, J.R., Emonet, T.: Efficient modeling, simulation and coarse-graining of biological complexity with nfsim. Nature Methods **8**, 177–183 (2011)
19. Grewal, R.K., Das, J.: Spatially resolved in silico modeling of nkg2d signaling kinetics suggests a key role of nkg2d and vav1 co-clustering in generating natural killer cell activation. PLOS Computational Biology **18**(5), 1–33 (2022). doi:10.1371/journal.pcbi.1010114
20. Konstorum, A., Vella, A.T., Adler, A.J., Laubenbacher, R.C.: Addressing current challenges in cancer immunotherapy with mathematical and computational modelling. Journal of The Royal Society Interface **14**(131), 20170150 (2017). doi:10.1098/rsif.2017.0150. https://royalsocietypublishing.org/doi/pdf/10.1098/rsif.2017.0150