bioRxiv preprint doi: https://doi.org/10.1101/2021.01.07.425782; this version posted January 8, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

dynUGENE: an R package for uncertainty-aware gene regulatory network inference, simulation, and visualization

Tianyu Lu[1,2]✉ **and Anjali Silva**[2,3,4]

[1]Department of Computer Science, University of Toronto, Toronto, Canada
[2]Department of Cell and Systems Biology, University of Toronto, Toronto, Canada
[3]Princess Margaret Cancer Centre, University Health Network, Toronto, Canada
[4]Vector Institute, Toronto, Canada

Methods for gene regulatory network inference focus on network architecture identification but neglect model selection and simulation. We implement an extension to the dynGENIE3 algorithm that accounts for model uncertainty as an R package, providing users with an easy to use interface for model selection and gene expression profile simulation. Source code is available at https://github.com/tianyu-lu/dynUGENE with a detailed user guide. A webserver with interactive controls is available at https://tianyulu.shinyapps.io/dynUGENE/.

Gene regulatory network | network inference

Correspondence: ***tianyu.lu @mail.utoronto.ca***

Introduction

Complex phenomena such as cell development and apoptosis emerge from coordinated dynamics of gene regulatory networks (GRN). Inferring network structure from data can be used for hypothesis generation, revealing mechanisms in cell development and disease (Huang et al., 2009), and modelling network evolution (Crombach and Hogeweg, 2008). Accurate dynamical models allow us to predict the effects of network perturbations on biological function, for example to push cells out of a disease state (Karlebach and Shamir, 2010), or to design synthetic GRNs given the desired dynamics of a network (Hiscock, 2019). The ideal model should be flexible enough to capture highly nonlinear interactions while not sacrificing model interpretability and computation time.

We present dynUGENE (dynamical Uncertainty-aware GEne NEwork inference), an R package that extends the functionality of dynGENIE3, a state-of-the-art method for GRN inference (Geurts et al., 2018). We build on dynGENIE3 because it satisfies all three of our model desiderata. Existing extensions include TIMEOR and BENIN which both incorporate heterogeneous data to improve network inference accuracy (Wonkap and Butler, 2020; Conard et al., 2020). Here, we take a different approach and instead account for uncertainty in dynGENIE3, allowing for stochastic gene expression simulations and parsimonious model selection. Our extension is available as an easy to use R package and also as an interactive web server.

Package Design

dynGENIE3 Background. dynGENIE3 poses GRN inference as a feature selection problem. It first trains random forests to predict the change in concentration of each species given the current concentrations of all species. Each interaction from species x_i to species x_j is associated with an importance score, calculated by the reduction in variance from using x_i to predict the change in x_j. The importance score for an interaction, when normalized, is interpreted as the probability of that interaction to exist. For a detailed treatment, see the vignette and (Geurts et al., 2018).

Model Selection. The inferred network can be visualized as a $p \times p$ matrix where the entry $[x_i, x_j]$ is the importance score of x_i for inferring x_j (Fig. 1). However, real GRNs are often not fully connected and the presence of an interaction is binary (Mangan et al., 2016). To address this, dynUGENE includes a function for model selection based on visualizing the Pareto front (Mangan et al., 2016). However, we note that the model at the sharp drop in the Pareto front is not always the best model (Supplementary Fig. S1). We include an additional function on the web server where users can choose which interactions to mask. The masked networks can then be simulated, allowing for application-specific tuning of model complexity.

Model Simulation. The inferred networks and masked networks can be used to simulate gene expression profiles by numerically solving the system of ordinary differential equations learned by the random forests. In addition to deterministic simulations, we provide an option that accounts for the uncertainty in the random forests predictions for stochastic simulations. For stochastic simulations, instead of only taking the mean of a random forest's predictions, we sample from the Gaussian $\mathcal{N}(\mu, \sigma^2)$ where μ is the mean and σ^2 is the variance of the random forest's predictions.

Provided Datasets. The dynUGENE package provides four example time-series datasets: repressilator, stochastic repressilator, Hodgkin-Huxley, and stochastic Hodgkin-Huxley (Elowitz and Leibler, 2000; Hodgkin and Huxley, 1952). These datasets were generated from systems of ordinary or

bioRxiv preprint doi: https://doi.org/10.1101/2021.01.07.425782; this version posted January 8, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

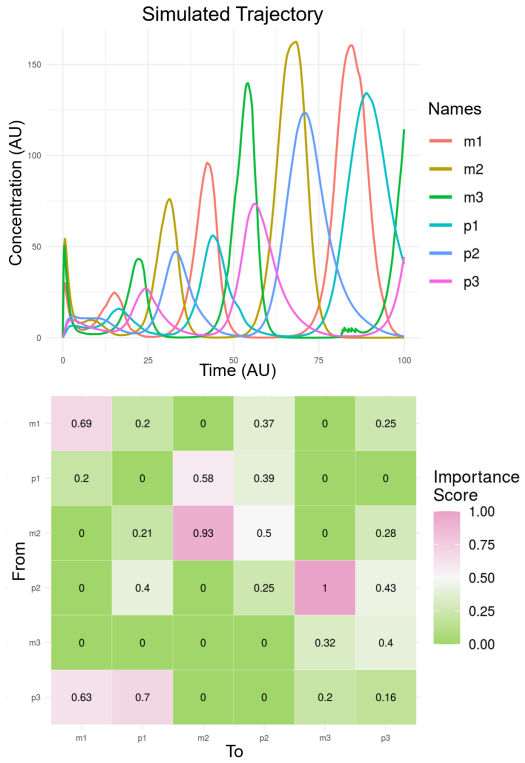

Fig. 1: Bottom: inferred importance scores on the repressilator dataset for the 16th network in the step-wise column masks plot (Supplementary Fig. S2). Top: Simulated trajectory using the inferred network.

stochastic differential equations. Details are provided in the vignette. The package also includes one steady state dataset, SynTReN300, taken from GRNdata (Bellot et al., 2020). Users can provide their own data as input following the format specified in `?inferNetwork`.

Discussion

A requirement for dynGENIE3 and dynUGENE is that all species must be tracked through time. This requirement is difficult to satisfy in practice as there are often unknown species in a biological process of interest. Methods that can identify or approximate latent structure in partially-observed systems are more appropriate here (Hiscock, 2019). An omics treatment such as RNA-seq can cover breadth but current sequencing techniques require cells to be destroyed, thus making time series data collection difficult. Non-destructive sequencing techniques could address this issue.

The implementation of an inferred network as a gene circuit will require more thought. Even for networks with sparse interactions, the likelihood of finding a set of genes and proteins that satisfy the interaction strengths and activation or inhibitory effects is unknown. In fact, whether a species is an activator or inhibitor is not explicitly given in the interaction matrix. We can address this by posing dynUGENE as a constrained optimization problem where it is limited to using only a given set of parts (genes, promoters, ribosome binding sites, proteins, etc.) thus relating the importance scores with biological interaction strengths. We leave this for future work.

Data and code availability

Source code is available at https://github.com/tianyu-lu/dynUGENE with a detailed user guide. A webserver with interactive controls is available at https://tianyulu.shinyapps.io/dynUGENE/.

ACKNOWLEDGEMENTS
The authors thank the authors of dynGENIE3 for their work and Alan Moses for guidance.

FUNDING
This work was supported by a Postdoctoral Fellowship from Canadian Institutes of Health Research.

Bibliography

Sui Huang, Ingemar Ernberg, and Stuart Kauffman. Cancer attractors: a systems view of tumors from a gene network dynamics and developmental perspective. In *Seminars in cell & developmental biology*, volume 20, pages 869–876. Elsevier, 2009.

Anton Crombach and Paulien Hogeweg. Evolution of evolvability in gene regulatory networks. *PLoS Computational Biology*, 4(7):e1000112, 2008.

Guy Karlebach and Ron Shamir. Minimally perturbing a gene regulatory network to avoid a disease phenotype: the glioma network as a test case. *BMC Systems Biology*, 4(1):15, 2010.

Tom W Hiscock. Adapting machine-learning algorithms to design gene circuits. *BMC Bioinformatics*, 20(1):1–13, 2019.

Pierre Geurts et al. dyngenie3: dynamical genie3 for the inference of gene networks from time series expression data. *Scientific Reports*, 8(1):1–12, 2018.

Stephanie Kamgnia Wonkap and Gregory Butler. Benin: Biologically enhanced network inference. *Journal of Bioinformatics and Computational Biology*, 18(03):2040007, 2020.

Ashley Mae Conard, Nathaniel Goodman, Yanhui Hu, Norbert Perrimon, Ritambhara Singh, Charles Lawrence, and Erica Larschan. Timeor: a web-based tool to uncover temporal regulatory mechanisms from multi-omics data. *bioRxiv*, 2020.

Niall M Mangan, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Inferring biological networks by sparse identification of nonlinear dynamics. *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, 2(1):52–63, 2016.

Michael B Elowitz and Stanislas Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(6767):335–338, 2000.

Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4):500, 1952.

Pau Bellot, Catharina Olsen, and Patrick E Meyer. *grndata: Synthetic Expression Data for Gene Regulatory Network Inference*, 2020. R package version 1.20.0.

Carl Ganz. rintrojs: A wrapper for the intro. js library. *Journal of Open Source Software*, 1(6):63, 2016.

Gregory R. Warnes, Ben Bolker, Lodewijk Bonebakker, Robert Gentleman, Wolfgang Huber, Andy Liaw, Thomas Lumley, Martin Maechler, Arni Magnusson, Steffen Moeller, Marc Schwartz, and Bill Venables. *gplots: Various R Programming Tools for Plotting Data*, 2020. R package version 3.1.0.

Hadley Wickham. *ggplot2: elegant graphics for data analysis*. springer, 2016.

Christopher Rackauckas and Qing Nie. Adaptive methods for stochastic differential equations via natural embeddings and rejection sampling with memory. *Discrete and Continuous Dynamical Systems. Series B*, 22(7):2731, 2017a.

Christopher Rackauckas and Qing Nie. Differentialequations. jl–a performant and feature-rich ecosystem for solving differential equations in julia. *Journal of Open Research Software*, 5 (1), 2017b.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2020.