Generative Models Validation via Manifold Recapitulation Analysis

Nicolo' Lazzaro*[1], Gianluca Leonardi[1], Raffaele Marchesi[1], Massimiliano Datres[1], Anna Saiani[1], Jacopo Tessadori[1], Alejandro Granados[2], Johan Henriksson[3], Marco Chierici[1], Giuseppe Jurman[1], Toma Tebaldi†[4], and Gabriele Sales[5]

[1]Fondazione Bruno Kessler, Trento, Italy
[2]Calico Life Sciences, San Francisco, United States
[3]Umeå universitet, Umeå, Sweden
[4]Università degli Studi di Trento, Trento, Italy
[5]Università degli Studi di Padova, Padova, Italy

October 2024

Abstract

Summary: Single-cell transcriptomics increasingly relies on nonlinear models to harness the dimensionality and growing volume of data. However, most model validation focuses on local manifold fidelity (e.g., Mean Squared Error and other data likelihood metrics), with little attention to the global manifold topology these models should ideally be learning. To address this limitation, we have implemented a robust scoring pipeline aimed at validating a model's ability to reproduce the entire reference manifold. The Python library Cytobench demonstrates this approach, along with Jupyter Notebooks and an example dataset to help users get started with the workflow. Manifold recapitulation analysis can be used to develop and assess models intended to learn the full network of cellular dynamics, as well as to validate their performance on external datasets.

Availability: A Python library implementing the scoring pipeline has been made available via pip and can be inspected at GitHub alongside some Jupyter Notebooks demonstrating its application.

Contact: nlazzaro@fbk.eu or toma.tebaldi@unitn.it

1 Introduction

As single-cell transcriptomics datasets expand in size and complexity, generative models are increasingly employed to understand and interpret their vast and highly nonlinear underlying dynamics [7, 1]. Most of the validation approaches currently available focus primarily on data likelihood and objective functions, which measure the model's ability to represent space only near the observed data points [8, 4]. This narrow focus limits our understanding of how well these models represent the broader biological processes they are intended to learn [12, 2]. The growth of transcriptomics datasets, on the other hand, offers the opportunity to train generative algorithms more capable of interpolating sparse regions of the manifold, advancing personalized medicine and drug development, and enhancing our understanding of processes that are opaque to traditional modeling approaches [6, 9]. In order to facilitate the implementation of such models, the framework proposed here is intended to serve as a baseline validation approach.

Our approach is fully unsupervised and evaluates the manifold-recapitulating capabilities of generative models in single-cell transcriptomics. It is designed to complement current estimates of local manifold fidelity (e.g., reconstruction error and sample likelihood) with global metrics on empirical distribution similarity. This task is complicated by the high dimensionality and sparsity of the underlying data, which exhibit varied landscapes in terms of space density and gene activation patterns. Our validation scheme makes no assumptions regarding the underlying sequencing technology: it directly measures how well a model reconstructs the empirical distribution of a target dataset from a subset of its samples. Therefore, this approach can be used to assess the biological understanding of a model and its applicability to external datasets (e.g., as an ideal null model).

2 Pointwise Empirical Distance

A simple approach to validate how closely a generative model captures the structure of a target population is to evaluate how well it can reconstruct the empirical distribution from a small subset of representative points. This process can be done either iteratively (e.g., via Markov Chain Monte Carlo simula-

*Corresponding author: nlazzaro@fbk.eu
†Corresponding author: toma.tebaldi@unitn.it

tion) or through one-shot approaches (e.g., leveraging Variational Autoencoders or Stable Diffusion models) [7, 1, 3, 5]. In both cases, the metric is represented by the score calculated as $1 - \frac{D_f}{D_i}$, where D_f is the distance between the model-expanded subset and the empirical distribution, and D_i is the distance of the initial subset of points (with replicates) with respect to the empirical distribution. This score ranges from 1 to negative infinity, with 1 indicating a perfect score, minus infinity indicating the worst possible case, and 0 corresponding to the distance of the initial point configuration. For practical reasons our implementation applies the Exponential Linear Unit (ELU) function to bound this quantity between 1 and -1, but the raw score can be used as well if needed.

In order to compare two empirical distributions, one corresponding to the original data and the other generated by a model, we could use measures like the energy distance, but the curse of dimensionality impacts significantly its sensitivity [10, 13]. Other options, like the Wasserstein distance, preserve more information, but its computation in high dimensions is onerous. As a tradeoff, real-world applications frequently resort to dimensionality reduction as a preprocessing step. Reducing the original space by orders of magnitudes forces a hard information tradeoff, which can make different distributions appear close (or even identical) in the compressed space. In light of these difficulties, we developed a new distance measure between empirical distributions that first computes the pairwise distances in the original space and then evaluates the distance between 1D distributions. This approach preserves all properties of real distances whilst bounding the computational complexity to the square of sample sizes.

More specifically, this distance - which we refer to as pointwise empirical distance (PED) - is computed by comparing the pairwise distance of every point in the first distribution with the pairwise distance against all points in the second distribution (and vice versa). Comparing the two distance distributions can be done with any nonparametric distance (e.g. Wasserstein distance). Averaging the resulting distance computed for each point in the two datasets will yield a valid distance measure, which we can leverage to evaluate a generative model on large datasets with a reasonable computational footprint. Mathematically, we define the PED as follows:

$$\begin{aligned}\mathrm{PED}(X,Y) =& \frac{1}{2}\mathbb{E}\left[\inf_{\pi}\left(\|d(X,X') - d(Y,X')_{\pi}\|_p\right)\right] \\ &+ \frac{1}{2}\mathbb{E}\left[\inf_{\pi}\left(\|d(Y,Y') - d(X,Y')_{\pi}\|_p\right)\right]\end{aligned}$$

The PED score can be interpreted as a measure of similarity between the target and generated distributions, with lower PED values indicating better correspondence. To evaluate statistical significance, we can generate a null distribution of PED values by repeatedly evaluating the PED on different subsamples of the initial distribution. This approach yields a distribution of PED values that can then be used to estimate a likelihood that an external empirical distribution was generated from the same ground distribution.

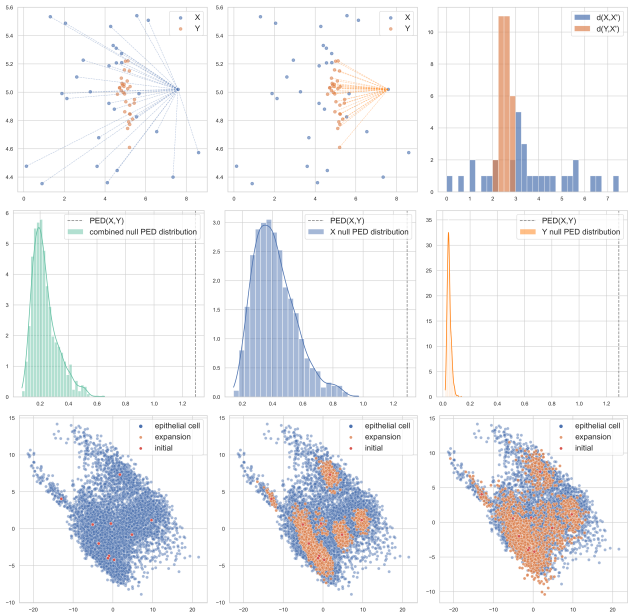

Figure 1. Top row: visual representation of PED contribution from a single point. The final PED is computed by repeating the same process for all X', then for all Y', and finally averaging all individual distributions distances. Middle row: comparison of PED(X,Y) against the null PED distribution of X+Y (green), X (blue), and Y (orange). These distributions can be used - each with distinct applications and significance - to assess the likelihood that empirical distributions originate from the same ground distribution. Bottom row: Cytobench scoring pipeline: For a single-cell transcriptomic dataset of epithelial cells, the process begins by selecting initial points for expansion, followed by scoring the generative algorithm at each expansion round on a scale from -1 to $+1$. In this case the computed scoring is 0.36 and 0.54 (example from the Jupyter Notebooks examples).

3 Scoring Pipeline

Our scoring pipeline requires two inputs from the user: (1) a dataset of samples, and (2) a custom Python function that takes an empirical distribution of points and returns samples from the same biological space. Optionally, a data validator can be provided to evaluate individual generated samples. This validator will then be used to add an independent penalty proportional to the ratio of invalid samples, enhancing the scoring process.

The pipeline begins by clustering the input dataset to select a small subset of representative points (the cluster centroids), duplicating each one proportionally to the size of its originating cluster. This initial set of points is then passed to the user-defined Python function to generate new samples within the same bi-

ological space. Sample generation can be performed either iteratively over a user-defined number of epochs or in a one-shot manner, depending on the generative approach employed.

The generated points are evaluated based on the relative decrease (or increase) in distance with respect to the original empirical distribution. A well-performing model will distribute the initial set of points (with duplicates) evenly across the underlying manifold topology, thereby decreasing their distance from the empirical distribution. Conversely, a poor model will produce samples biased toward external attractors (such as better known cell types from the training set) reducing the coverage of the target manifold compared to the initial set of points. By tracking the distance progression, we obtain a robust scoring metric that naturally penalizes sampling bias while rewarding accurate manifold recapitulation.

For large datasets encompassing multiple cell types, it is generally beneficial to evaluate generative models locally. This approach allows for tracking model performance across different regions of the manifold, rather than expecting uniform performance across the entire dataset which, while ideal, is often nontrivial. The library automatically handles this by first clustering the samples with the Leiden algorithm and then tracking local cluster scores alongside the global one. Various scoring approaches (e.g., global, local, balanced) are implemented and presented in the companion Jupyter Notebooks.

An important consideration is that since the library is meant to evaluate overall empirical distributions it will lack the likelihood evaluation of single samples typically found in traditional models. To address this, the scoring function accepts an optional user-defined custom Python function: the data validator. This function should return a boolean flag for a set of input cells, assessing their biological validity. This information is leveraged to further penalize points that have moved outside the target biological space, with an adjustable penalty factor that, by default, scales quadratically with the number of invalid points.

4 Case Study

In the GitHub repository, we present a detailed explanation of the pointwise empirical distance (and divergence). We demonstrate how the benchmarking workflow can be employed to validate a generative neural network. The associated Jupyter Notebooks can be run directly on Google Colab, eliminating the need for a local setup.

To enable interactive execution of the notebooks, the repository includes a reduced version of the Tabula Muris dataset[11], selected on the basis of its diverse range of cell types. By focusing on the 128 genes with the highest variance, we reduced the memory footprint to a level manageable by Colab while retaining sufficient expressiveness for demonstration purposes. The Jupyter Notebooks are designed to be accessible to non-bioinformaticians, providing a broad understanding of the material we discuss. They include modular code hands-on experimentation with the library and dataset.

Then we demonstrate how local null distributions can be used to account for different manifold volumes across biological cell types and configurations. This application specifically showcases how Cytobench can be leveraged to set up a series of cell-type-specific datasets to test foundational generative models against. By fixing the experiment's initial conditions, we can measure performance and identify pitfalls in a quantitative and precise manner, thereby offering a comprehensive assessment tool for generative models in single-cell transcriptomics.

Overall we expect our approach to facilitate significant advancements in the understanding and application of generative models in biological research.

Acknowledgments

This work was supported by by the European Union through the 3DSecret project under the HORIZON-EIC-2022-PATHFINDER-OPEN-01-01 programme [grant number 101099066].

References

[1] Mohanad Abukmeil, Stefano Ferrari, Angelo Genovese, Vincenzo Piuri, and Fabio Scotti. A survey of unsupervised generative models for exploratory data analysis and representation learning. *ACM Comput. Surv.*, 54(5), July 2021.

[2] Gunnar Carlsson. Topology and data. *Bulletin of The American Mathematical Society - BULL AMER MATH SOC*, 46:255–308, 04 2009.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680. Curran Associates, Inc., 2014.

[4] Koichi Handa, Morgan C Thomas, Michiharu Kageyama, Takeshi Iijima, and Andreas Bender. On the difficulty of validating molecular generative models realistically: a case study on public and proprietary data. *J. Cheminform.*, 15(1):112, November 2023.

[5] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022.

[6] Bingyu Liu, Shunfeng Hu, and Xin Wang. Applications of single-cell technologies in drug discovery for tumor treatment. *iScience*, 27(8):110486, August 2024.

[7] Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nat. Methods*, 15(12):1053–1058, December 2018.

[8] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study, 2018.

[9] Jianzhu Ma, Michael Ku Yu, Samson Fong, Keiichiro Ono, Eric Sage, Barry Demchak, Roded Sharan, and Trey Ideker. Using deep learning to model the hierarchical structure and function of a cell. *Nat. Methods*, 15(4):290–298, April 2018.

[10] Maria L Rizzo and Gábor J Székely. Energy distance. *Wiley Interdiscip. Rev. Comput. Stat.*, 8(1):27–38, January 2016.

[11] Nicholas Schaum, Jim Karkanias, Norma F. Neff, Andrew P. May, Stephen R. Quake, Tony Wyss-Coray, Spyros Darmanis, Joshua Batson, Olga Botvinnik, Michelle B. Chen, Steven Chen, Foad Green, Robert C. Jones, Ashley Maynard, Lolita Penland, Angela Oliveira Pisco, Rene V. Sit, Geoffrey M. Stanley, James T. Webber, Fabio Zanini, Ankit S. Baghel, Isaac Bakerman, Ishita Bansal, Daniela Berdnik, Biter Bilen, Douglas Brownfield, Corey Cain, Min Cho, Giana Cirolia, Stephanie D. Conley, Aaron Demers, Kubilay Demir, Antoine de Morree, Tessa Divita, Haley du Bois, Laughing Bear Torrez Dulgeroff, Hamid Ebadi, F. Hernán Espinoza, Matt Fish, Qiang Gan, Benson M. George, Astrid Gillich, Geraldine Genetiano, Xueying Gu, Gunsagar S. Gulati, Yan Hang, Shayan Hosseinzadeh, Albin Huang, Tal Iram, Taichi Isobe, Feather Ives, Kevin S. Kao, Guruswamy Karnam, Aaron M. Kershner, Bernhard M. Kiss, William Kong, Maya E. Kumar, Jonathan Y. Lam, Davis P. Lee, Song E. Lee, Guang Li, Qingyun Li, Ling Liu, Annie Lo, Wan Jin Lu, Anoop Manjunath, Kaia L. May, Oliver L. May, Marina McKay, Ross J. Metzger, Marco Mignardi, Dullei Min, Ahmad N. Nabhan, Katharine M. Ng, Joseph Noh, Rasika Patkar, Weng Chuan Peng, Robert Puccinelli, Eric J. Rulifson, Shaheen S. Sikandar, Rahul Sinha, Krzysztof Szade, Weilun Tan, Cristina Tato, Krissie Tellez, Kyle J. Travaglini, Carolina Tropini, Lucas Waldburger, Linda J. van Weele, Michael N. Wosczyna, Jinyi Xiang, Soso Xue, Justin Youngyunpipatkul, Macy E. Zardeneta, Fan Zhang, Lu Zhou, Paola Castro, Derek Croote, Joseph L. DeRisi, Christin S. Kuo, Benoit Lehallier, Patricia K. Nguyen, Serena Y. Tan, Bruce M. Wang, Hanadie Yousef, Philip A. Beachy, Charles K.F. Chan, Kerwyn Casey Huang, Kenneth Weinberg, Sean M. Wu, Ben A. Barres, Michael F. Clarke, Seung K. Kim, Mark A. Krasnow, Roel Nusse, Thomas A. Rando, Justin Sonnenburg, and Irving L. Weissman. Single-cell transcriptomics of 20 mouse organs creates a tabula muris. *Nature*, 562:367–372, 10 2018.

[12] Laurens van der Maaten, Eric Postma, and H. Herik. Dimensionality reduction: A comparative review. *Journal of Machine Learning Research - JMLR*, 10, 01 2007.

[13] Cedric Villani. *Optimal Transport*. Grundlehren der mathematischen Wissenschaften. Springer, Berlin, Germany, December 2009.