Machine Learning in Computational Biology

23/5/2024

Assignment #3

The goal of this assignment is to get familiar with high-dimensional datasets, dimensionality reduction methods and unsupervised learning for single-cell state identification. You will be working with a pre-processed single-cell RNA-seq dataset containing 137 cells and 54,675 genes, available in the "Assignments" folder of eclass as "Assignment_3_2024_dataset.zip"**.** Your task involves creating a data analysis pipeline within a Python or R file. This pipeline should take any single-cell RNA-seq dataset as input and execute the following stages of data analysis, which you will present in a Jupyter or R-markdown notebook:

1. **Dimensionality reduction [33 points]** of the expression data using at least three alternative methods which should include, but not be limited to, Principal Components Analysis (PCA), t-distributed stochastic neighbour embedding (well known as "t-SNE"), and Uniform Manifold Approximation and Projection (UMAP). The latter two are most often used to visualize high-dimensionality datasets in 2 or 3 dimensions that humans can perceive.
   It is crucial to:
   1. Select the best number of dimensions for PCA based on some criterion.
   2. Choose the optimal parameters for t-SNE & UMAP.

   Hint: There are multiple methods for 1. On the other hand, 2. may require some trial and error. Your choices should be justified and reflect a good understanding of each technique.

2. **Clustering [33 points]** of the dimensionality-reduced data (resulting from each of the three methods) into the optimal number of cell groups (cell "states") using Gaussian Mixture Modeling (GMM), as probabilistic clustering enables the calculation of probabilities for each cell belonging to each cluster. Ensure your solution automatically identifies the optimal GMM model by evaluating the number of components and their covariance matrix structure using established model selection techniques such as the Bayesian Information Criterion (BIC). Don't forget to produce cell posterior probabilities as well as evaluating the performance of each clustering solution. By the conclusion of this stage, each cell should possess a posterior distribution for every state (cluster). Justify all your actions.

3. **Visualization [34 points]** of your results (clusters inferred, cell posteriors, cell joint distributions, the Gaussian Distribution of each cluster component, etc.) in the most intuitive manner for a human investigator. It is your job to select the best methods to present your results and unfold a convincing story to the evaluator (as if you are submitting a paper for review). The output should include a "labels.csv" file. This file should feature cells as indexes and a single column named "labels," which will indicate, using an integer, the cluster label to which each cell primarily belongs (e.g. if we have 5 clusters use labels: 0,1,2,3,4).

Important Notes:

- You are allowed to use any available package in R or python that you deem suitable for this assignment.
- Ensure your submission presentation notebook includes clearly labeled graphs with markdown explanations for all three tasks of the pipeline. Your explanations should cover what you did, why you did it, and how you did it (not only the code but also the methodology and thought process behind it). **Your grade will heavily depend on the clarity of these explanations and graphs.**
- Provide insight into which parameters of your pipeline might be ideal for user tuning (if any). Back up your reasoning with comprehensive explanations.
- Avoid writing new functions in your presentation notebook file. It should be used to present the pipeline, which should cover all the analysis needs described above.
- As always, the sci-kit library documentation provides valuable information for all the tasks described above.

Bonus parts:

- **(for up to 100% bonus points):** Suggest a drastically different computational pipeline for the same data analysis workflow (e.g., not using GMMs for producing posterior probability estimates, or BIC to select "best" models) and compare systematically your original to the new pipeline. Also, make both pipelines parametric so that they can work with a dataset of any size in terms of the number of cells and genes provided. Test both pipelines using realistic synthetic data that you may generate yourselves using GMMs with different numbers of states and state "topologies" (covariance matrices).
- **(for up to 100% bonus points):** We've kept the true class labels (cell types) for each cell in a secret file. We will use them to with your clustering outcomes ("labels.csv" file) using the Adjusted Rand Index (ARI), a metric that compares cluster assignment with ground truth labels. The student achieving the highest ARI score will receive this bonus!

Bonus parts of assignments are optional. Not doing them will not affect your final course grade. However, if you consistently accumulate "Bonus points" in the class assignments (i.e., you demonstrate consistent extra effort), your course grade will be boosted by as much as a whole mark (e.g., a 7.2 may become 8, a 4.2 may become 5, etc.) and if you ask for a reference letter in the future this extra effort will be noted.

An important note on LLM usage for coding:

Learning programming/Data Science is not about mastering a single thing. It is mostly about gathering bits and pieces of knowledge and fitting them together like a puzzle. Once you have built "muscle memory" from practicing with assignments like this one, you can easily use your knowledge to tackle similar but harder problems in the future. However, if you rely on LLMs without 100% understanding of the code they provide, you enter an infinite cycle where you will keep returning to it for the same tasks over and over again, and each time it is varied, often error-prone solutions will only lead to further confusion. It is much better to use LLMs as a teaching assistant if needed, although the sci-kit learn package provides great documentation with in-depth explanations for all of the tools it provides, and it is a great idea to familiarize yourself with its use as early as possible. After all, the assignment solutions that utilize copy/pasted LLM code are most often distinguishable, and we have the tools and experience to spot them!

If you do use LLM assistance you should disclose it, as it is now becoming a standard practice when submitting publications. In that case, please add a small section at the end of your report explaining which LLM you used and for what purposes (tasks). Feel free to discuss your experience.

Deadline: You should upload your code to e-class in a well-organized .zip file with your name and ID number by **June 14, 2024.**

Each student should work independently, this is not a team project.