Binary Clustering: Evaluating runtime

Single-cell RNA sequencing(scRNAseq) is a new techinque to measure expression levels of indiviudal cells. Analysis of this data has lead to valuable biological insights. However, as datasets include more cells, this analysis becomes more and more computationally intense. This research looks into time-efficiency effects of a novel idea for data analysis: binary clustering.

Introduction

- scRNAseq datasets have been increasing in both number of cells measured and sparsity
- Clustering of these datasets costs a lot of time and memory
- Previous research [1] has shown that storing the data in a binary format (Figure 1) can reduce storage costs, while retaining most of the biological information.
- This suggests that a specialized algorithm for binary data could be more time-efficient
- We introduce two versions of a binary clustering algorithm:
  - Exact, where every cell is compared to every other cell
  - Approximated, cells are only compared to cells that are likely to be similar
- The binary approaches are experimentally compared to an existing library (Seurat)

	Cell 1	Cell 2	Cell 3	Cell 4
Gene 1	0	0	3	0
Gene 2	2	0	4	1
Gene 3	0	0	1	0
Gene 4	10	2	3	0

	Cell 1	Cell 2	Cell 3	Cell 4
Gene 1	0	0	1	0
Gene 2	1	0	1	1
Gene 3	0	0	1	0
Gene 4	1	1	1	0

Fig 1: Converting scRNAseq data

Algorithms

General workflow:

1. Load data into memory
2. Pre-process data
3. Create nearest neighbour (kNN) graph (most time-consuming step)
4. Assign clusters

Unique properties:

Seurat:
- Works on numerical data
- Applies several dimensionality reduction methods
- Approximates kNN graph

Binary Exact:
- Works on binary data.
- Compares every cell to every other cell for the kNN graph

Binary Approximated:
- Works on binary data
- Approximates kNN graph (using the same approximation library as Seurat)

Methodology

Experimentally evaluate runtime of:
- Seurat
- Exact binary algorithm
- Approximated binary algorithm

Measure runtime of only steps 2 & 3

Datasets

10 datasets were used:
- 1 dataset with 500.000 cells and 500 genes
- 9 datasets with 1.000, 10.000, and 100.000 cells, and 500, 1.000 and 2.000 genes

Conclusion

- Binary clustering can be faster than Seurat in some cases
- When the number of dimensions is equal, binary clustering is much faster
- Approximation of kNN graph is required to be competitive on datasets with more **cells**
- Dimensionality reduction is required to be competetive on datasets with more **genes**

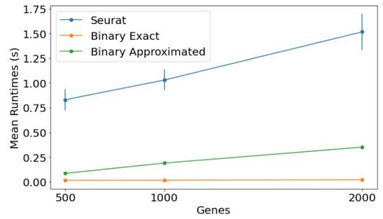
Overall, binary clustering shows promise, since it could be faster than Seurat with the right additions and modifications. Proper approximation and dimensionality reduction techniques are essential to achieve this.

Related literature

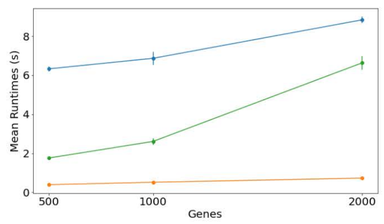
[1] G. A. Bouland, A. Mahfouz, and M. J. Reinders, "The rise of sparser single-cell rnaseq datasets; consequences and opportunities," bioRxiv, pp. 2022–05, 2022.

Results

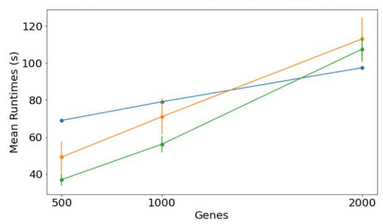
- Binary clustering is significantly faster than Seurat on smaller datasets. (Figures 2a and 2b)
  - This ranges from 2 to 50 times faster.
- Approximated methods (Seurat and approximated binary) scale better when **cell** counts increase. (Figure 3)
  - This is because approximated methods have a better time complexity than the exact method. (O(n log(n)) vs. O(n^2))
- Seurat scales much better then both binary methods when **gene** counts increase in large datasets. (Figure 2c)
  - Seurat uses dimensionality reduction methods before doing the most time-consuming operations. This allows the rest of the clustering procedure to **not** be affected when more genes are used.
- Performing the most time consuming part of the clustering procedure (k-NN graph creation) is orders of magnitude faster with binary data than with continous data.
  - Because of the dimensionality reduction, this stage can still be faster with continuous data than with binary data, because way less dimensions are considered.
  - However, when normalizing for the amount of dimensions, the binary algorithm is 118 times faster on average.



(a) Mean runtimes for 1.000 cells.

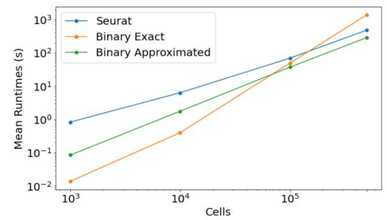


(b) Mean runtimes for 10.000 cells.

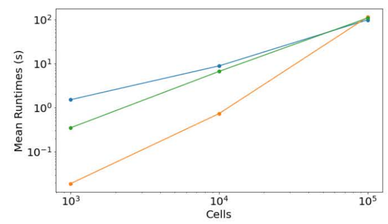


(c) Mean runtimes for 100.000 cells.

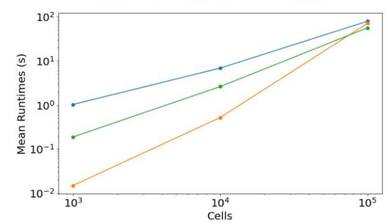
Fig 2: Mean runtimes for datasets with different cell counts.



(a) Mean runtimes for 500 genes.



(b) Mean runtimes for 1.000 genes.



(c) Mean runtimes for 2.000 genes.

Fig 3: Mean runtimes for datasets with different gene counts.

Authors
Milan de Koning
milandekoning@tudelft.nl

Supervisors
Gerard Bouland
G.A.Bouland@tudelft.nl

Marcel Reinders
M.J.T.Reinders@tudelft.nl

Affiliations
Delft University of Technology
Faculty of Electrical engineering, Mathematics and Computer Science