MS BIOSTATISTICS CAPSTONE – Group 1:

Benchmark Analysis of InSituType Cell Typing Model for Spatial Transcriptomics

Sponsor: NanoString Technologies, Inc

Project Members:

Howard Baek, Eliza Chai, Alexis Harris, Ingrid Luo, Makayla Tang

Executive Summary

In 2020, *Nature Methods* crowned spatially resolved transcriptomics Method of the Year (Nature, 2021). Spatially resolved transcriptomics has changed the landscape of RNA sequencing and cell type analyses. When single-cell RNA sequencing (scRNA-seq) was developed, researchers could investigate a finer-grained assessment of each cell's transcriptome – a cell's gene expression. However, scRNA-seq dissociates cells from tissue to discern cell types based on the gene expression. Now with spatially resolved transcriptomics, researchers and scientists can retrieve cell transcriptomic data and locate the positional context of those cells in tissue (Nature, 2021). While spatial analysis is not yet routinely used to deliver transcriptome wide information of all single cells, spatial single cell transcriptomics is expected to be the next step after scRNA-seq. A critical component in analyzing spatial transcriptomic data is cell typing, which is the process of determining for every cell in the dataset, "what kind of cell is this?" NanoString Technologies, Inc., has developed a cell typing algorithm, InSituType. InSituType is designed for statistical and computational efficiency for spatial transcriptomic data. For this capstone, we ran competing cell typing methods against InSituType on two different datasets and scored the accuracy of these models. Additionally, we performed a benchmarking analysis to determine how the performance of InSituType varies based on tuning parameters. We found InSituType's performance is competitive compared to current well-known cell typing methods.

About NanoString

NanoString Technologies, Inc. is a publicly held biotech company offering discovery and translational research solutions. Their products and services provide robust analyses for spatial biology and spatial transcriptomics. Their recent product, CosMx™ Spatial Molecular Imager, provides single-cell resolution on FFPE and fresh frozen tissue sections with quantification of up to 1,000 RNAs. They also recently developed a cell typing method called InSituType – this model provides tools for analyzing spatial transcriptomic data.

Background

What is spatial biology?

Spatial biology is defined as the study of tissues within their own 2D or 3D context and is the new frontier of molecular biology. The principle of utilizing GPS location coordinates to create a map can be compared to spatial biology visualizations on a cellular and molecular level. Through this technology, we can map the spatial architecture of a cell and how it talks to and interacts with its surroundings. Spatial biology is like being inside a tissue sample at a molecular level. It enables single-cell analysis. Through spatial biology tools, you are empowered to see things that are not possible by single-cell sequencing or any other technologies available.

Why is spatial genomics important?

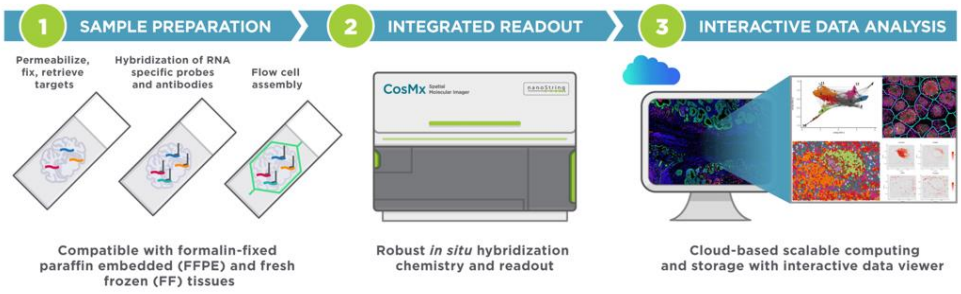
Biology is inherently spatial, and individual cells do not function in isolation but work together forming a complex network of gene interactions spatially to organize themselves and communicate with their surroundings. The spatial profile of each gene expression within a cell is tightly linked

to the biological functions it drives. Therefore, localization of gene expression with a positional context and understanding how transcriptional dynamics vary within such context is crucial to interpreting various biological, physiological, and pathological processes occurring in a single cell or tissue. For example, the endocrine system responds to cues from the hypothalamus which results in changes in gene expression. This change in gene expression within the endocrines system regulates many physiological processes such as blood pressure and metabolism. Thus, spatial biology serves as a useful tool when studying oncology, immune-oncology, neurobiology, and even COVID research.

Spatial genomics, also known as spatial transcriptomics, includes profiling methods for genomic (total DNA and RNA), transcriptomic (RNA transcripts), and epigenomic (molecular compounds) which are all carried out in intact tissue with positional context. Spatial transcriptomic *technologies* provide an unbiased map of targeted mRNA transcripts through tissue sections. To provide this map, this technology combines approaches in molecular biology encompassing immune fluorescence techniques, gene expression, and next-generation sequencing. A common tissue sample used in spatial transcriptomics is formalin-fixed paraffin-embedded tissue specimen (FFPE). FFPE tissue specimen are samples preserved and prepared from biopsy specimens. This method of preservation of tissue specimen has been a primary technique for single-cell research which aids in examination, experimental research, and diagnostic and drug development. Fluorescence techniques are used on the tissue sample to label target proteins or RNA. Once this region is identified, the user will cleave off molecular tags using ultraviolet light. These tags are then processed and analyzed.

An example of this process can be visualized in the following diagram. This diagram depicts the flow of spatial transcriptomics analysis for NanoString Technologies, Inc. Products.

Work Flow of Spatial Transcriptomics for NanoString Technologies, Inc.



Source: https://nanostring.com/products/geomx-digital-spatial-profiler/geomx-dsp-overview/

Cell Typing and InSituType

Cell typing is a fundamental aspect of single cell transcriptomics analysis, which involves categorizing cells based on their expression profiles. While unsupervised clustering is commonly used, supervised classification can also be employed in tissues with high-quality reference datasets to avoid clustering's instability and interpretation difficulties. Spatial transcriptomics platforms detect individual RNA molecules in tissue samples and maintain cells' location data, producing datasets that can be used for cell typing. In this context, the InSituType algorithm is introduced for cell typing in spatial transcriptomics data.

NanoString's cell typing model InSituType utilizes a likelihood model to enable supervised cell typing from reference datasets via a Bayes classifier and unsupervised or semi-supervised cell typing via an Expectation Maximization algorithm. InSituType is designed to address the challenges of sparse and multi-modal spatial transcriptomics data, and it employs an escalating subsampling scheme to handle large datasets efficiently. Compared to existing cell typing methods, InSituType offers several advantages, including its ability to handle large datasets, its use of a likelihood model to weigh the evidence from every transcript in a cell, its incorporation of alternative data types such as images and spatial context, and its ability to identify new clusters alongside reference cell types.

Description of the Problem

What is the scientific gap?

We investigated the performance of InSituType's supervised and unsupervised cell typing methods and how it compares to competing cell typing models. In a benchmarking analysis, InSituType was found to outperform competitor methods in unsupervised cell typing, but further analysis is needed to understand how its performance varies based on tuning parameters. Additionally, it is unclear whether supervised InSituType is the most effective approach. Hence, we compared the performance of InSituType's supervised method to competing cell typing models.

Objectives

We have two objectives for our projects. The first objective is to understand how the performance of the unsupervised InSituType method varies due to tuning parameters of interest. We investigated the InSituType parameters relating to the number of clusters, number of iterations, subsample size for random starts, and number of cohorts. To measure the performance of the model, we used the Adjusted Rand Index, Bayesian information criterion, and recorded computation time to run the model on a consistent computing environment. The second objective is to assess the performance of the supervised InSituType method and other open-source supervised algorithms using a kidney biopsy dataset.

Data sources

To benchmark InSituType, we utilized two samples of tissue datasets. To analyze the performance of the unsupervised method, we use a dataset obtained from a "cell pellet array" (CPA) where various cell lines are pelleted onto a slide. This CPA was profiled using the CosMx™ Spatial Molecular Imager. Each field of view (FOV) comprises cells solely from a distinct cell line, providing us with accurate cell identity information. This dataset contains information for 52,518 cells and has thirteen known clusters, or cell lines.

To benchmark the performance of the InSituType supervised cell typing method, we used a dataset from a kidney biopsy taken from a lupus nephritis patient. This dataset was also profiled using the CosMx™ Spatial Molecular Imager and is a single cell RNA dataset with 61,073 mature kidney cells. We perform cell typing on this dataset using reference profiles from the Kidney Single Cell Atlas. This training dataset is from a single cell RNA study that performed an analysis of all the cell types in the human kidney and is comprised of 40,268 mature kidney cells and 33 cell types.

Overview of the Method

Benchmarking unsupervised clustering

We examine the performance of unsupervised clustering method by tuning parameters of interest. In this analysis, we tuned the parameters for the number of clusters, the number of iterations, the subsample size for phase 1 random starts, and the number of cohorts argument in the "fastCohorting " function (used to quickly split cells into cohorts). Benchmarking was performed on the full CPA dataset and two down sampled CPA datasets containing a random sample of 50% of the genes and the 10% most highly variable genes (HVG).

To access performance, we used three evaluation metrics: Bayesian Information Criterion (BIC), Adjusted Rand Index (ARI), and computation time. With lower BIC values, we obtain models that better fit the data. A high ARI indicates high similarity between the true cell type labels and the clustering partition. ARI equals 1 when there is perfect correspondence and takes a value near 0 for a random partition. The recorded system run time allows us to access the computation resources required when running on a consistent computing environment.

Number of clusters

The number of clusters is the number of Field of Views (FOV) in this dataset. The field of view for a microscope is the extent of the observable area in distance unit. However, this may not always give optimal fit and accuracy. In this CPA dataset used for benchmarking, we know there are thirteen cell types, however this information will not always be known when performing unsupervised cell typing on spatial transcriptomic data. To tune the number of clusters parameter, we tested a range of 10 to 25 clusters to observe the change in performance due to the initial estimate of the number of clusters.

Number of iterations

InSituType iteratively runs an Expectation Maximization (EM) algorithm on subsamples of cells in the input dataset. With more iterations, we are more likely to get a better fit at the cost of longer runtime. The default value is 10 iterations, but we run the function with values ranging from 1 to 20 iterations and evaluate the performance with the three proposed metrices.

Number of cells in Phase 1

The number of cells in phase 1 represents the subsample cell size for phase 1 random starts. The default cell sample size for phase 1 is 10,000 cells. InSituType randomly selects 10,000 cells from the input dataset and performs early iterations of EM. The early iterations analyzed in phase 1 are used for an escalating subsampling scheme to quickly approach an approximate solution in large datasets. After phase 1 is complete, phase 2 iterates the EM algorithm on a subsample of 20,000 cells to obtain a more precise solution, and then phase 3 iterates over 100,000 cells. The final phase analyzes the whole dataset once where cells are classified using profiles obtained from previous phases.

To tune the number of cells in phase 1, we investigated a subsample size range from 1,000 to 20,000 cells, indexing by 1,000 cells and evaluated the performance using the three proposed metrics.

Number of cohorts

"Cohorting" is a pre-clustering step performed before InSituType is run. The use of the cohort parameter in InSituType incorporates alternative data types. Examples of these alternative data types include information on the spatial context of the cell types in tissue, images, and immunofluorescence stains with diverse distributions. The cohort membership defines the cell's prior probability of belonging to each cell type, in other words informing the likelihood calculations in the EM algorithm. We have no idea how fine-grained these pre-clusters should be. As a result, we explore values ranging from 0 to 100 to see how the default value of 25 performs. Similar to the other parameters, we evaluated the performance using ARI, BIC, and system time and tested only on the full CPA dataset.

Benchmarking supervised cell typing

To perform supervised cell typing, it is necessary to have both a training set and a test set. Our test set is a kidney biopsy from a lupus nephritis patient, and the Kidney Single Cell Atlas serves as the training set which provides average expression profiles for each kidney cell type based on their single-cell RNA-seq studies. Since there is no definitive gold standard cell type data for this sample, we cannot determine with certainty whether a cell type call is accurate. To evaluate the quality of our cell typing results, we rely on known kidney biology and use two metrics.

The first metric evaluates the accuracy of glomerular cell assignments to glomeruli, as kidneys contain substructures called glomeruli that comprise distinct cell types. Our expectation is that 100% of podocytes and glomerular endothelial cells will be assigned to glomeruli, while 0% of proximal tubule cells will be assigned to glomeruli.

We use bar charts (**Figure1, 2, 3**) to present the first metrics, showing the percentage of each cell type in the glomeruli, along with confidence intervals. We use a red line to indicate the target percentage for each cell type (i.e., 100% for podocytes and glomerular endothelial cells, and 0% for proximal tubule cells). The cells are grouped based on their transcript expression level, including all cells and cells with 0-50, 51-100, 101-200, and >200 transcripts.

We perform benchmarking on three datasets: the full dataset, and two down sampled datasets which are a random sample of 50% of the genes and the top 10% most highly variable genes. We do this in response to certain company competitors who offer products focused on analyzing the top 10% most highly variable genes.

Second, we record the intensity of a few "marker proteins" that are highly informative of cell type for each cell. We compare the marker protein mean expression intensity between cell types that are either positive or negative for the markers, such as CD45 for immune cells versus other cell types, or PanCK for cytokeratin+ versus cytokeratin- cell types.

Similar to the first metrics, the second metrics use bar charts (**Figure 4, 5**) to display the results. We present the logarithmic difference in mean PanCK stain intensity between cytokeratin+ and cytokeratin- cells (**Figure 4**), and the logarithmic difference in CD45 stain intensity between immune and non-immune cells (**Figure 5**). The classification of cell types as either cytokeratin-positive or cytokeratin-negative, as well as immune or non-immune, is determined through biological knowledge. We use logarithm base 2 ratio instead of a simple difference between the

mean stain intensity for two cell groups because a logarithmic scale can better represent the fold difference between groups.

To evaluate the performance of our supervised cell typing approach, we compare InSituType to SingleR, CHETAH, SeuratV3, SingleCellNet, scPred, and SVM. Additionally, we conduct all analyses in a consistent computing environment and record computation times.

- SingleR: An unbiased cell typing method for scRNA-seq by leveraging reference transcriptomic datasets of pure cell types to infer the cell of origin of each single cell independently.
- CHETAH (Characterization of Cell Types Aided by Hierarchical classification): A scRNA-seq classifier by hierarchical clustering of the reference data. The classification tree enables a step-wise, top-to-bottom classification.
- SeuratV3: A single-cell transcriptomics classifier can anchor diverse datasets together, enabling us to integrate single-cell measurements not only across scRNA-seq, but also across different modalities (e.g. scATAC-seq).
- SingleCellNet: A random forest classifier to learn cell type-specific gene pairs from cross-platform and cross-species datasets and thus quantitatively assesses cell identity at a single-cell resolution.
- ScPred: A scRNA-seq classifier by using a combination of unbiased feature selection from a reduced-dimensional space (e.g. PCA), and machine-learning probability-based prediction methods.
- SVM (support vector machines): Supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis.

Results

The InSituType supervised benchmark analysis demonstrated that InSituType is a competent cell typing method compared to other publicly available approaches, though its performance is not consistently the best. We excluded scPred from testing its accuracy since it left 98% of cells unassigned. CHETAH left half of the cells unassigned, so we need to interpret its results with caution. Additional insights can be found in the figure/table legends.

The second section presents the results from tuning parameters in the InSituType unsupervised method. From tuning the parameters of interest, we found the accuracy of the InSituType unsupervised method is relatively robust. This is important for application as researchers will not always have a known “true” cell type to evaluate the accuracy of the InSituType on their spatial transcriptomic data. Further insight into the tuning analysis can be found under each figure.

Benchmarking supervised cell typing

The accuracy of glomerular cell assignments to glomeruli

Regarding the cell typing accuracy of podocytes, SVM showed the highest performance, followed by InSituType in second place **(Figure 1)**. InSituType demonstrated a relatively consistent performance across all genes, half genes, and 10% highly variable genes. There was no significant difference observed between the no cohort and IF cohort. On the other hand, CHETAH showed

the lowest performance. SingleCellNet showed high accuracy in all genes and 10% highly variable genes, but its accuracy significantly dropped when using only half of the genes. With respect to the accuracy of cell typing for glomerular endothelial cells, CHETAH demonstrated better performance in predicting cell types for all genes, while InSituType had the highest accuracy for half genes **(Figure 2)**. There was no significant difference between the NO cohort and IF cohort. SingleCellNet performed reasonably well in all genes and 10% highly variable genes, but its accuracy significantly decreased when using only half of the genes. In general, SingleR and Seurat had the lowest performance. In terms of the accuracy of cell typing for the epithelial cells of the proximal tubule in glomeruli, InSituType, SingleR, and CHETAH demonstrated the highest performance, while SingleCellNet had the lowest accuracy **(Figure 3)**. The difference between the NO cohort and IF cohort was negligible. These results remained consistent across all genes, half genes, and 10% highly variable genes.

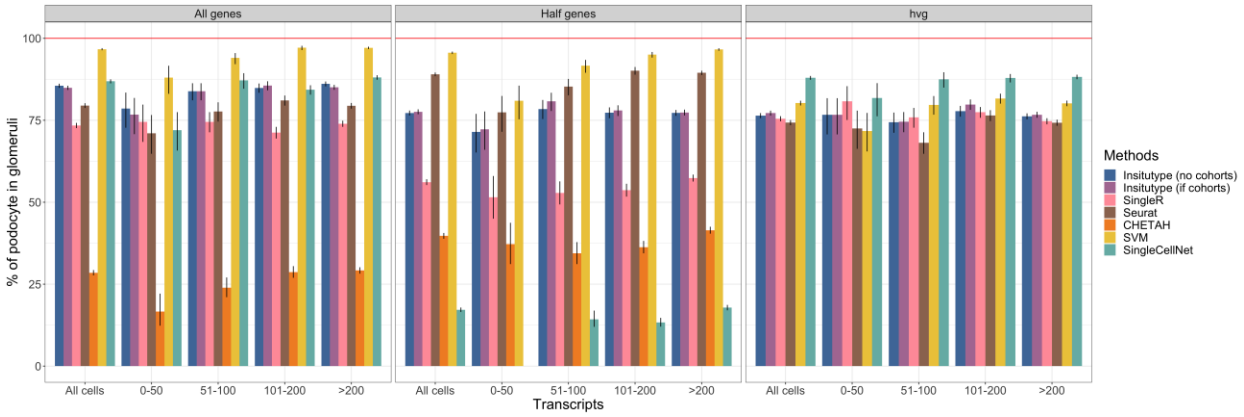


Figure 1. Supervised Benchmarking Cell Typing Performance for Podocytes in Glomeruli. We benchmarked the performance of different supervised cell typing methods using full dataset, half of the genes randomly selected, and the top 10% of most highly variable genes. Each bar shows the percentage of podocytes in glomeruli after cell typing, calculated by the exact number of podocytes in glomeruli divided by the total number of podocytes after cell typing. Error bars represent the lower and upper bounds of the 95% confidence interval for the binomial proportions calculated using the Wilson method. The ideal percentage of podocytes in glomeruli is 100%. Note: CHETAH is not compatible with the top 10% of the most highly variable genes.

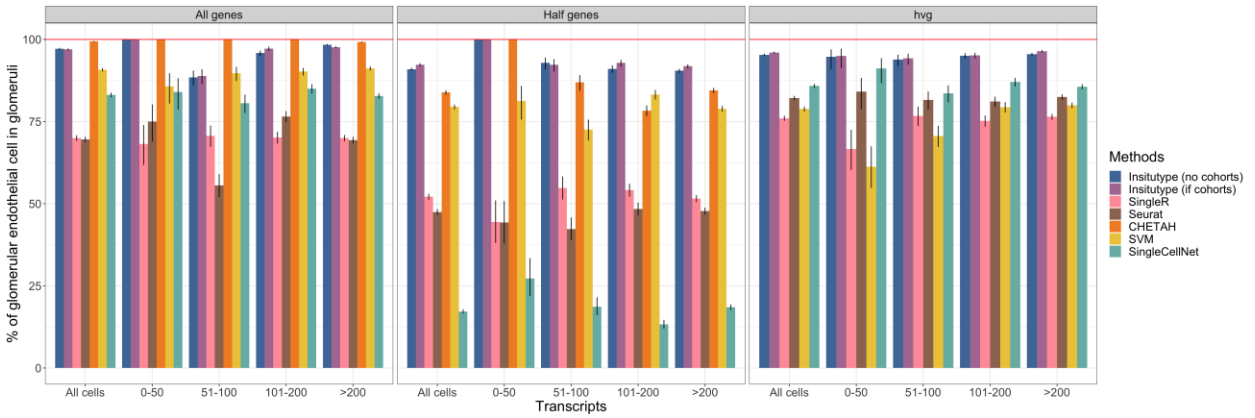


Figure 2. Supervised Benchmarking Cell Typing Performance for Glomerular Endothelial Cells in Glomeruli. We benchmarked the performance of different supervised cell typing methods using full dataset, half of the genes randomly selected, and the top 10% of most highly variable genes. Each bar shows the percentage of glomerular endothelial cells in glomeruli after cell typing, calculated by the exact number of glomerular endothelial cells in glomeruli divided by the total number of glomerular endothelial cells after cell typing. Error bars represent the lower and upper bounds of the 95% confidence interval for the binomial proportions calculated using the Wilson method. The ideal percentage of glomerular endothelia cells in glomeruli is 100%. Note: CHETAH is not compatible with the top 10% of the most highly variable genes.

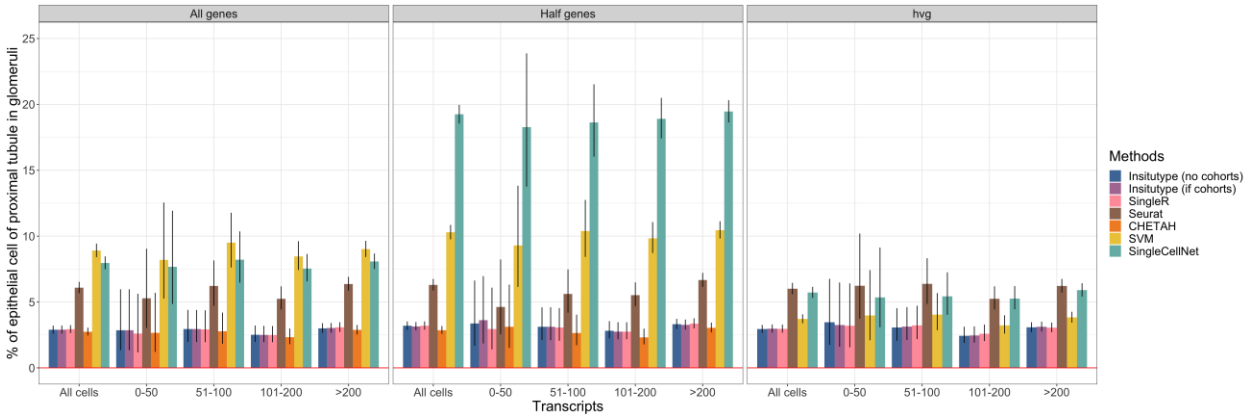


Figure 3. Supervised Benchmarking Cell Typing Performance for Epithelial Cells of Proximal tubule in Glomeruli. We benchmarked the performance of different supervised cell typing methods using full dataset, half of the genes randomly selected, and the top 10% of most highly variable genes. Each bar shows the percentage of epithelial cells of proximal tubule in glomeruli after cell typing, calculated by the exact number of epithelial cells of proximal tubule in glomeruli divided by the total number of epithelial cells of proximal tubule after cell typing. Error bars represent the lower and upper bounds of the 95% confidence interval for the binomial proportions calculated using the Wilson method. The ideal percentage of epithelial cells of proximal tubule in glomeruli is 0%. Note: CHETAH is not compatible with the top 10% of the most highly variable genes.

Fold difference of biomarker stain intensity in CK+/CK- and immune/non-immune cells

To evaluate the accuracy of accuracy, we also compared the difference in protein marker intensity between two cells typing groups using logarithmic intensity ratio. A log2 ratio of 1 represents a 2-fold change, a log2 ratio of 2 represents 4-fold change. The higher log intensity ratio shows the staining is more specific to the two cell typing groups, indicates the cell typing method can reliably distinguish between the two cell type groups. We compared the differences in PanCK intensity between cytokeratin+ cells and cytokeratin- cells **(Figure 4)**. SVM demonstrated the best performance, while CHETAH and SingleCellNet performed the worst accuracy compared to the other methods. Especially, the accuracy of SingleCellNet dropped significantly when using half of the genes randomly selected. In terms of CD45 intensity, we compared the differences between immune cells and non-immune cells **(Figure 5)**. CHETAH and SVM performed relatively well compared to the other methods when using full genes. On the other hand, the accuracies of SingleCellNet were unstable across different subsets of genes, especially the accuracy dropped significantly when using half of the genes randomly selected.

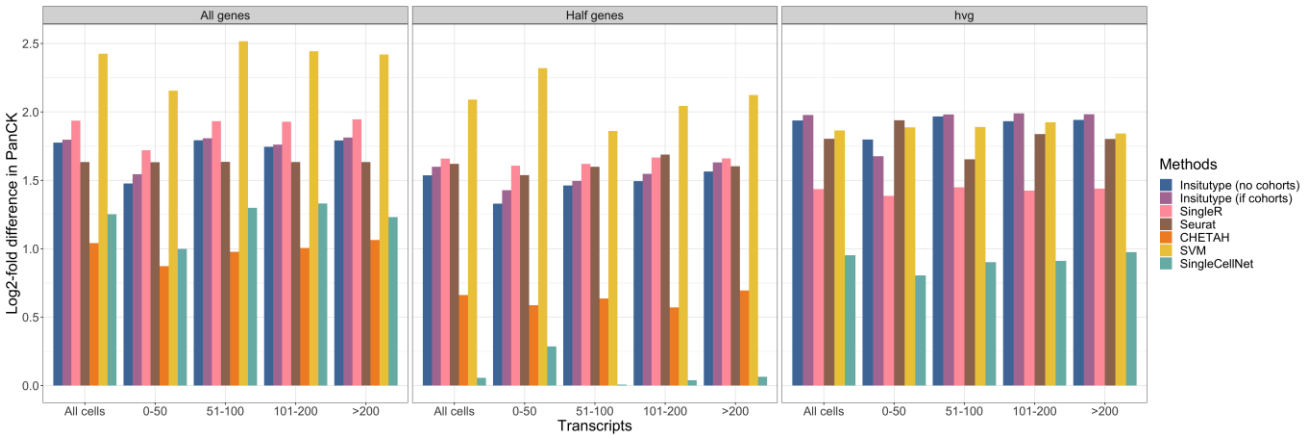


Figure 4. Supervised Benchmarking Performance in Cell Typing of Mean difference in PanCK intensity between cytokeratin+ and cytokeratin- cells. The Logarithmic PanCK intensity ratio between cytokeratin+ and cytokeratin- cells using full dataset, half of the genes randomly selected, and the top 10% of most highly variable genes. Each bar shows the log2-fold mean difference in PanCK between cytokeratin+ and cytokeratin- cells, where the mean PanCK intensity of the cytokeratin+ cells is divided by the mean PanCK intensity of the cytokeratin- cells. The higher log intensity ratio indicates the cell typing method is more accurate. Note: CHETAH is not compatible with the top 10% of the most highly variable genes.

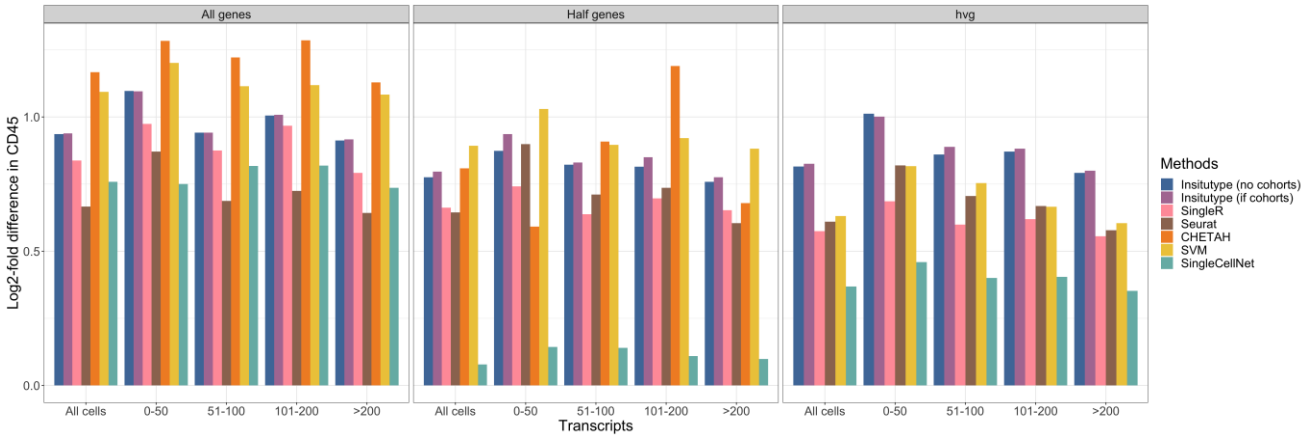


Figure 5. Supervised Benchmarking Performance in Cell Typing of Mean difference in CD45 intensity between immune and non-immune cells. Logarithmic CD45 intensity ratio between immune and non-immune cells using full dataset, half of the genes randomly selected, and the top 10% of most highly variable genes. Each bar shows the log2-fold mean difference in CD45 between immune cells and non-immune cells, where the mean CD45 intensity of the immune cells is divided by the mean CD45 intensity of the non-immune cells. The higher log intensity ratio indicates the cell typing method is more accurate. Note: CHETAH is not compatible with the top 10% of the most highly variable genes.

Computational time

InSituType and SingleR are the most efficient, while SingleCellNet is the slowest.

	InSituType (no cohort)	InSituType (if cohort)	SingleR	Seurat	CHETAH	SVM	SingleCellNet
All genes	31.25 sec	33.0 sec	1.18 min	2.99 min	26.11 min	6.48 min	~1.5 hr
Half genes	15.65 sec	15.73 sec	44.07 sec	3.99 min	34.27 min	3.55 min	64.13 min
10% hvg	5.24 sec	5.06 sec	3.93 sec	5.93 min	NA	1.79 min	39.58 min

Table 1. Supervised Benchmarking Running Time for Each Cell Typing Methods.

Benchmarking Unsupervised cell typing

We use scatter plots with LOESS (locally estimated scatterplot smoothing) curves and 95% confidence intervals (95% CI) to present the results for the three metrics. Each scatter plot displays the performance in terms of ARI, BIC, and computation time against the parameter of interest; the black diamonds in the scatter plot indicate the result for the default parameter value in the InSituType unsupervised model. The use of cohorts is indicated by IF cohort (i.e., pre-clustering step performed) and otherwise is indicated as no cohort. The following results are divided into sections of the tuning parameter. The number of clusters, iterations, and cells in phase 1 contain metrics for the unsupervised analysis of the full CPA dataset, a downstream random sample of 50% of the genes, and the 10% most highly variable genes. The last variable controlling for the number of cohorts presents the metrics on the full gene dataset.

Number of clusters – Results

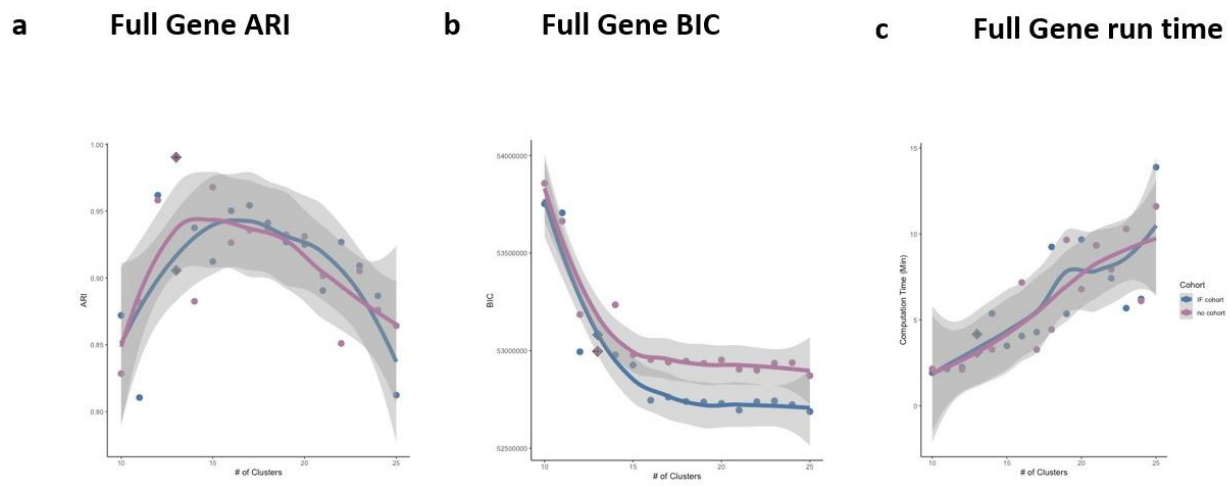


Figure 6. Unsupervised Benchmarking Performance in Cell Typing. This was performed on the full gene dataset.

We can see that the default number of clusters specified in the InSituType function, 13, performs the best in terms of ARI and BIC. This makes sense as the true number of clusters is 13 in the CPA dataset. Looking at the computation time, we can see that it increases linearly with the number of clusters increasing.

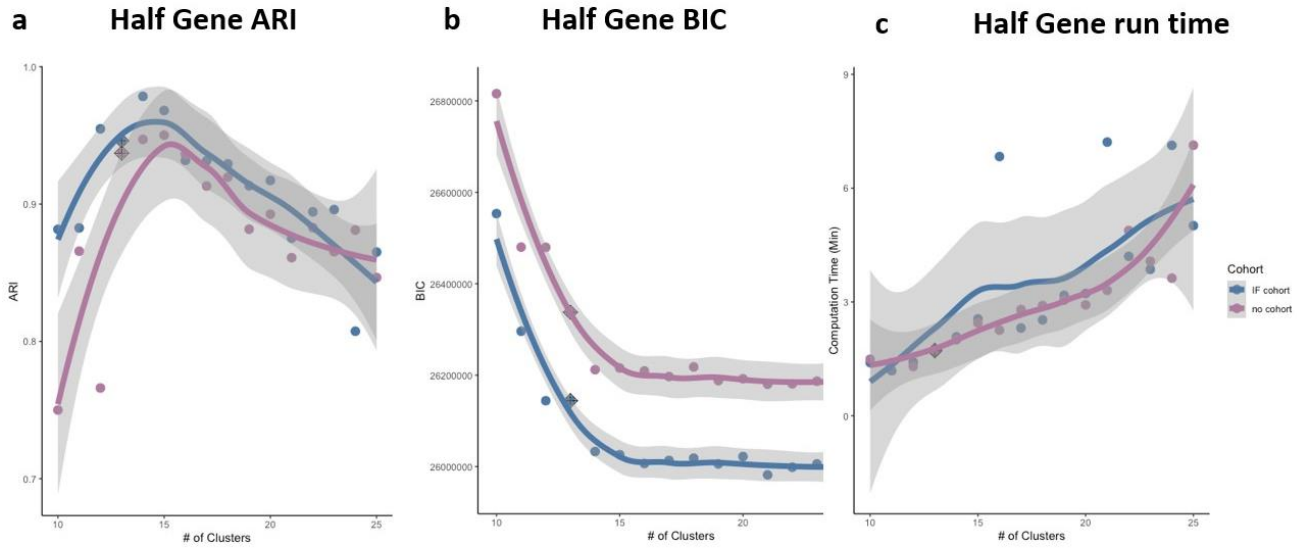


Figure 7. Unsupervised Benchmarking Performance in Cell Typing. This was performed on the half gene dataset.

The default number of clusters specified in the InSituType function, 13, also performs the best in terms of ARI and BIC. Looking at the computation time, we can see that it increases linearly with the number of clusters increasing.

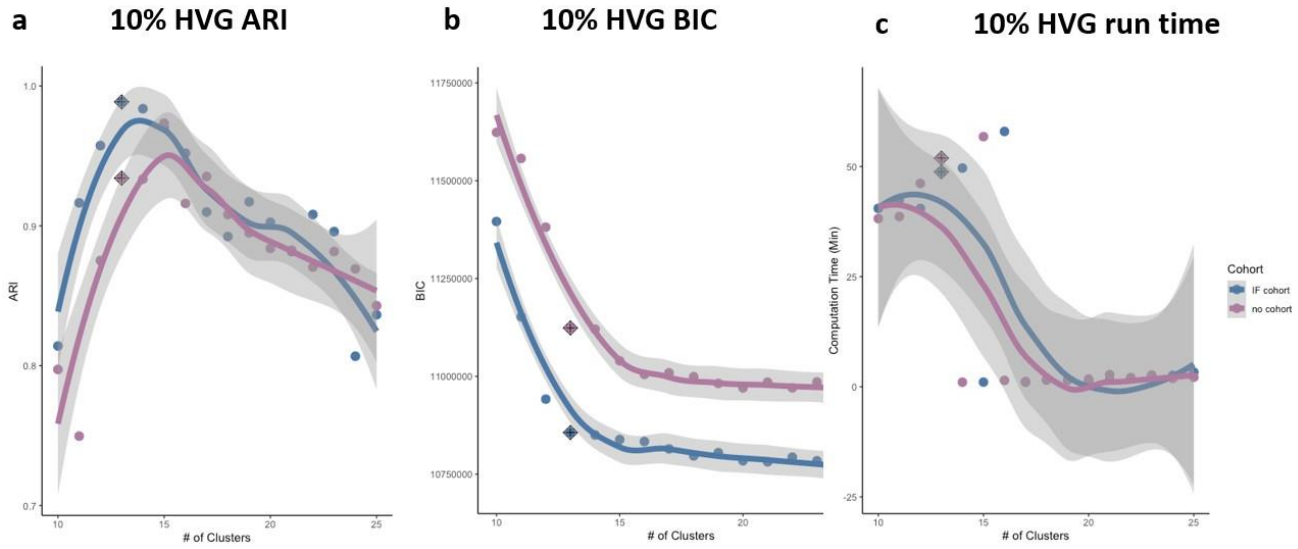


Figure 8. Unsupervised Benchmarking Performance in Cell Typing. This was performed on the 10% highly variable gene dataset.

We can see that the default number of clusters specified in the InSituType function, 13, similarly performs the best in terms of ARI and BIC. Looking at the computation time, the method takes longer to converge for lower values of clusters compared to higher values.

Number of iterations - Results

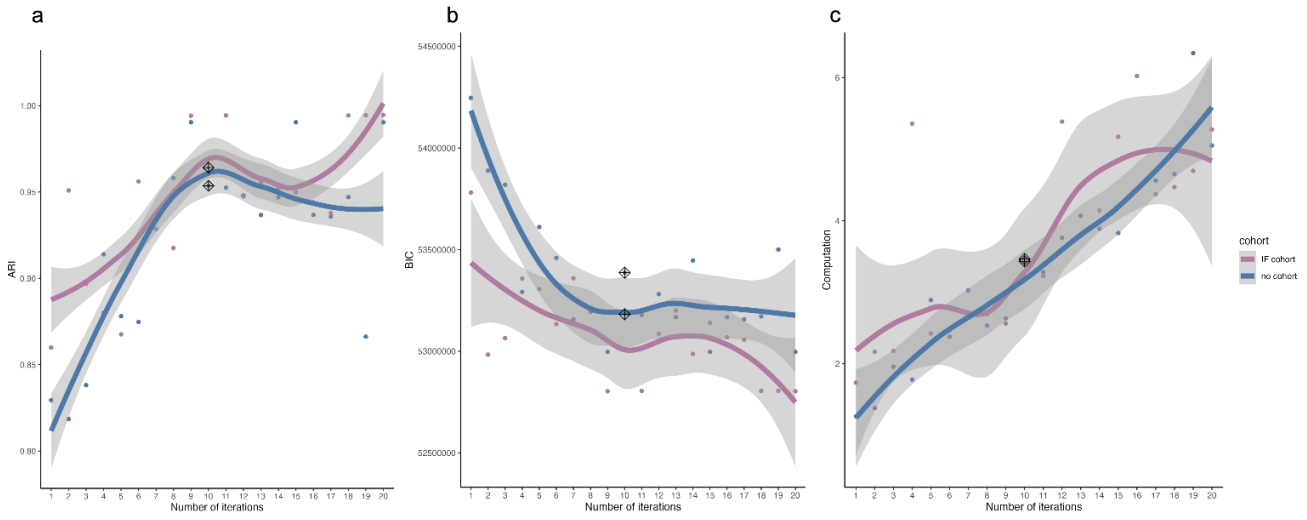


Figure 9. Unsupervised Benchmarking Performance in Cell Typing on the full gene dataset. The default number of iterations specified in the InSituType function, 10.

Here, the default number of iterations specified in the InSituType function, 10, performs the best in terms of ARI and BIC. Computation time increases linearly as the number of iterations increases.

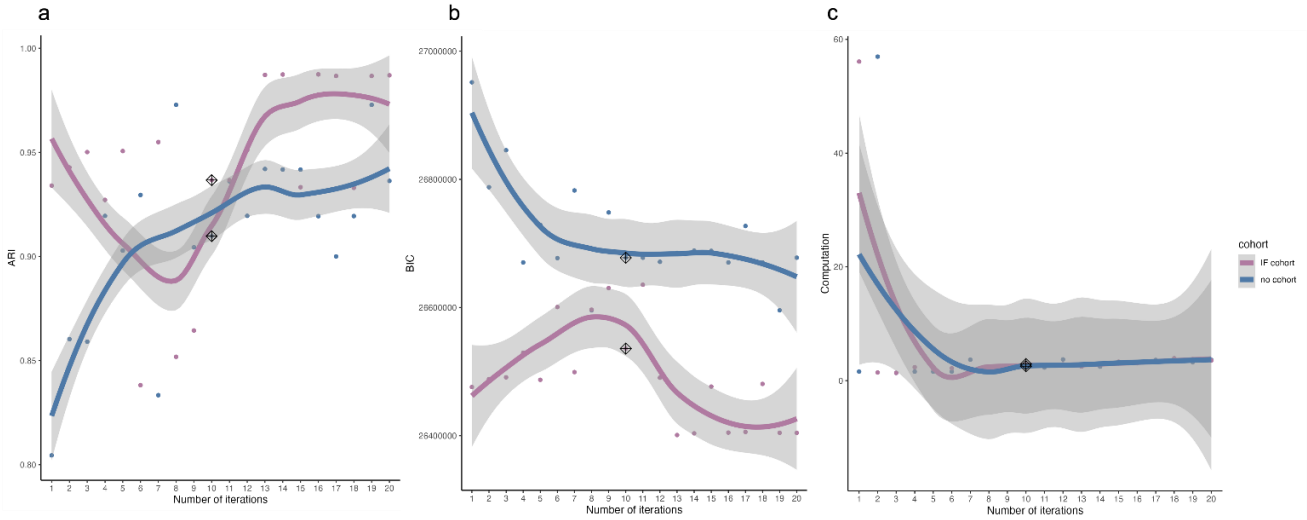


Figure 10. Unsupervised Benchmarking Performance in Cell Typing on the half gene dataset. The default number of iterations specified in the InSituType function, 10.

We can see that the default number of iterations specified in the InSituType function, 10, also performs the best in terms of ARI and BIC. Looking at the computation time **(c),** the model appears to take longer to complete cell typing for performing iterations under 10, and then levels for iterations greater than 10.

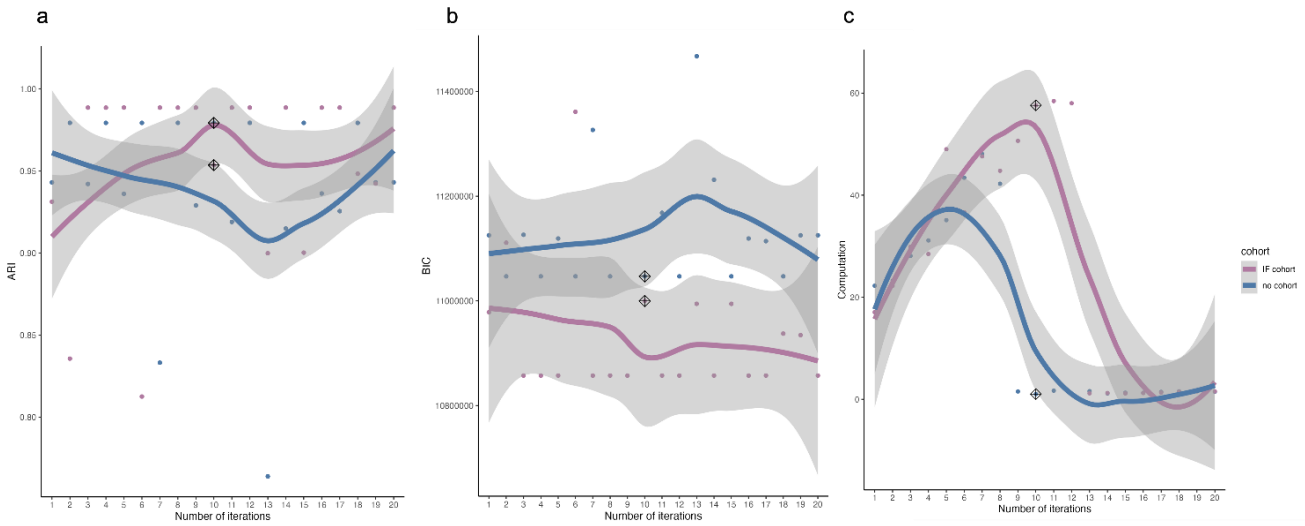


Figure 11. Unsupervised Benchmarking Performance in Cell Typing on the 10% highly variable gene dataset. The default number of iterations specified in the InSituType function, 10.

The default number of iterations specified in the InSituType function, 10, also performs the best in terms of ARI and BIC. Looking at the computation time, we can see that it increases linearly with the number of iterations increasing and peaks at 10 iterations, and then quickly drops after.

Number of cells in Phase 1 – Results

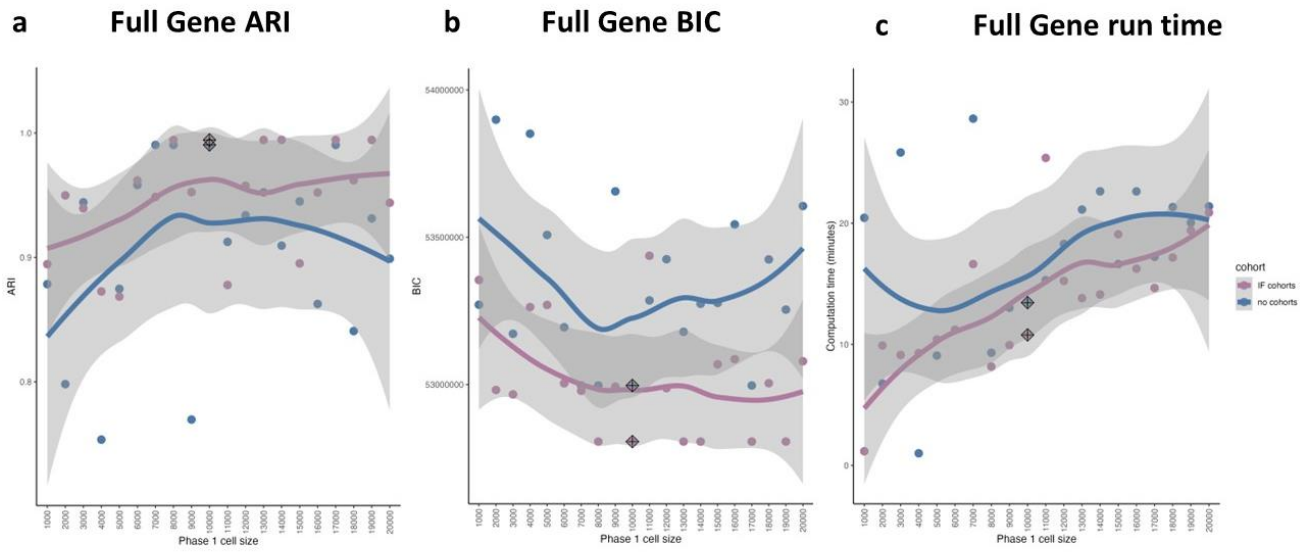


Figure 12. Unsupervised Benchmarking Performance in Cell Typing on the full gene dataset with respect to number of cells in phase 1; default subsample = 10000 cells.

In this figure we present performance metrics for **(a)** ARI, **(b)** BIC, and **(c)** computational time in minutes against tuning the parameter controlling the size of random subsample cells for phase 1. The phase 1 subsample size ranges from 1000 to 20000 in 1000 cell size increments. In terms of ARI and BIC, we see that using additional cohorting information (**IF cohorts**) appears to perform

on average better than using no cohorts. However, the 95% CI for all metrics overlaps in many areas, potentially indicating there are not significant differences. Additionally, both ARI and BIC do not show a defined trend as subsample size in phase 1 increases. The computational time **(c)** increases relatively linear as cell subsample size increases which is what we expect given the InSituType model is iterating over larger samples in the first phase.

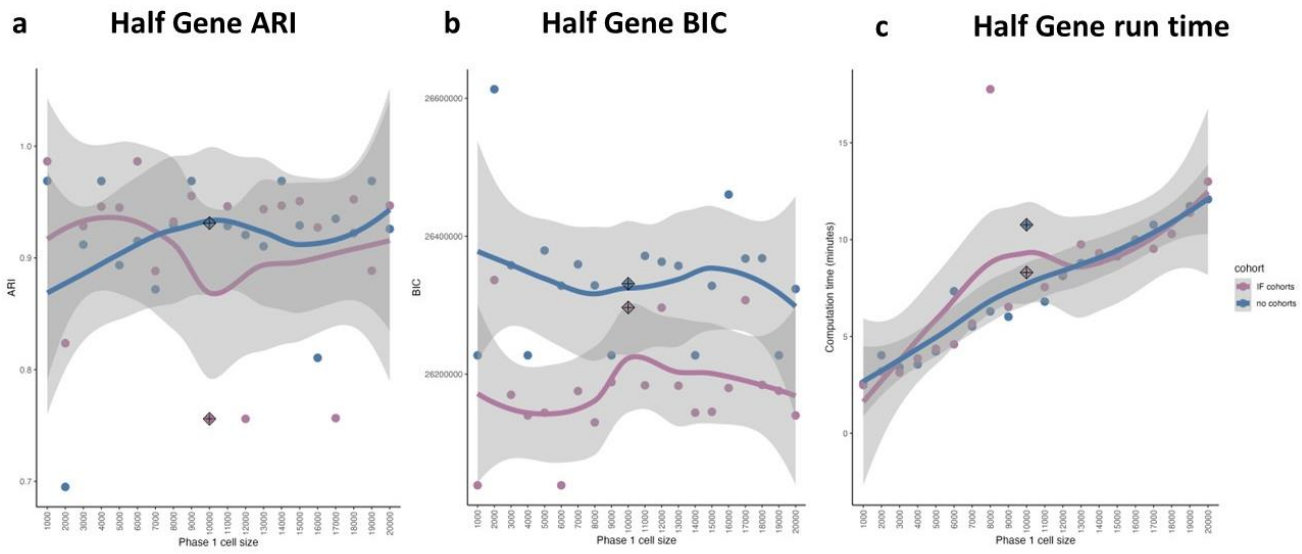


Figure 13. Unsupervised Benchmarking Performance in Cell Typing on the half gene dataset with respect to number of cells in phase 1; default subsample = 10000 cells.

In this figure we present performance metrics for **(a)** ARI, **(b)** BIC, and **(c)** computational time in minutes against tuning the parameter controlling the size of random subsample cells for phase 1. The phase 1 subsample size ranges from 1000 to 20000 in 1000 cell size increments. We see both the IF cohorts and no cohorts unsupervised model appears to perform similarly regarding ARI **(a)**, and the IF cohorts unsupervised model consistently trends below the cohort unsupervised model regarding BIC **(b)**. However, increasing the subsample size in phase 1 does not seem to improve performance, but it does increase computational time **(c).** This may lead us to consider using a smaller subsample size for phase 1, as the increase in performance metrics (**a** & **b**) may not be significant but the computational time will increase.

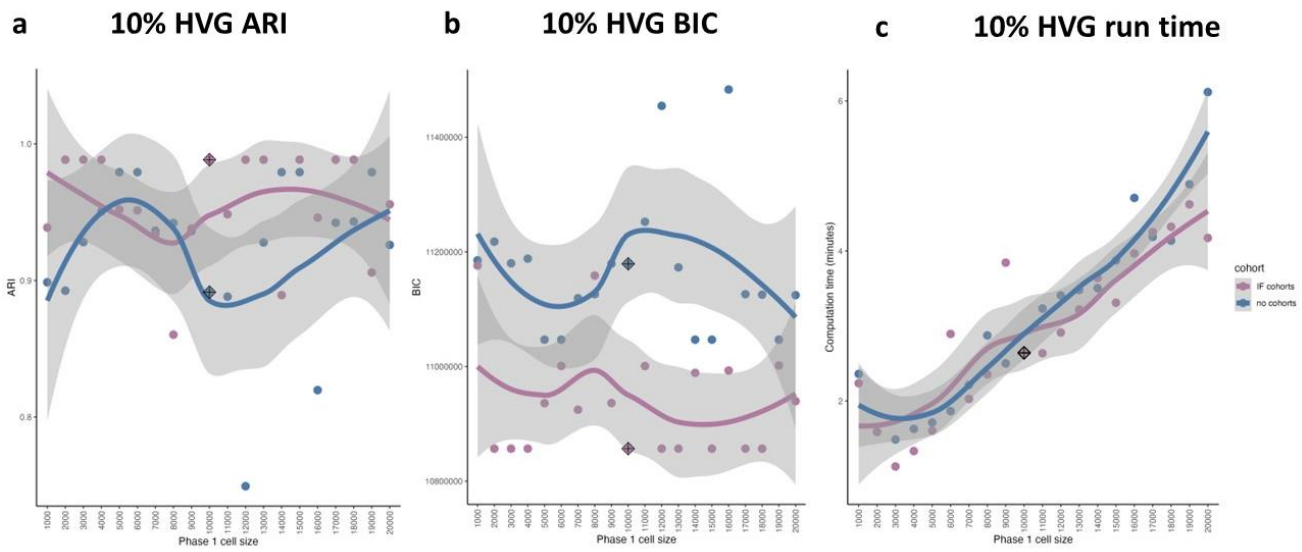


Figure 14. Unsupervised Benchmarking Performance in Cell Typing on 10% highly variable gene dataset with respect to number of cells in phase 1; default subsample = 10000 cells.

In this figure we present performance metrics for **(a)** ARI, **(b)** BIC, and **(c)** computational time in minutes against tuning the parameter controlling the size of random subsample cells for phase 1. The phase 1 subsample size ranges from 1000 to 20000 in 1000 cell size increments. We see both the IF cohorts and no cohorts unsupervised model appears to perform similarly regarding ARI **(a)**, and the IF cohorts unsupervised model consistently trends below the cohort unsupervised model regarding BIC **(b)**. However, increasing the subsample size in phase 1 does not seem to improve performance, but it does increase computational time **(c).**

<u>Number of cohorts – Results</u>

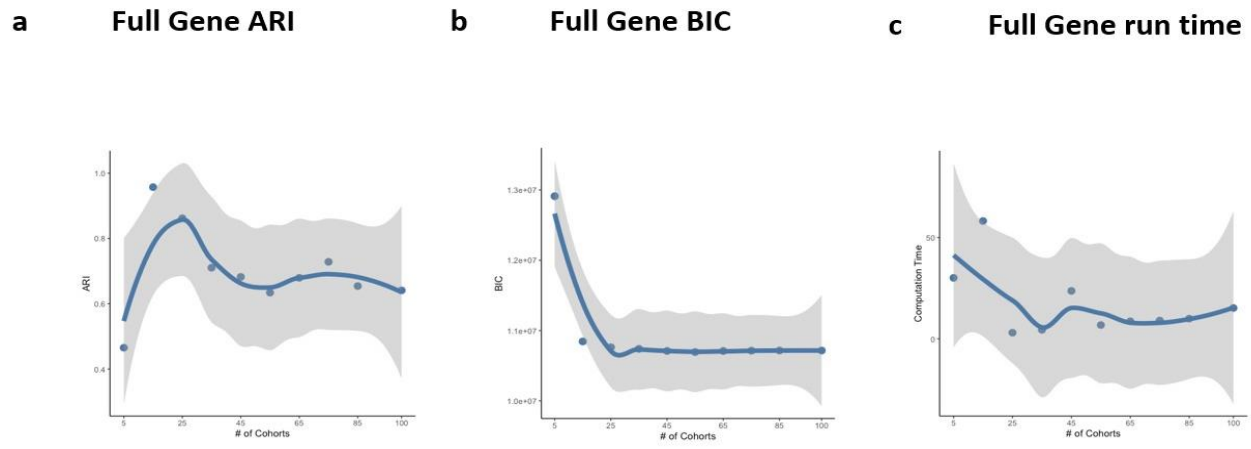


Figure 15. Unsupervised Benchmarking Performance in Cell Typing. This was performed on the full gene dataset; default cohort value = 25.

We explored values of the number of cohorts ranging from 0 to 100 and evaluated the performance using ARI, BIC, and system time in minutes. The default number of cohorts, 25,

performs the best in terms of ARI and BIC. The computation time decreases as the number of cohorts reaches 25 and then levels for cohort greater than 25.

Discussion

In our benchmark analysis, we found NanoString Technologies, Inc.'s model InSituType provides applicable cell typing methods for spatial transcriptomic data and research. InSituType's supervised cell typing method competitively performed against established cell typing methods: SingleR, CHETAH, SeuratV3, SingleCellNet, scPred and SVM. However, we found no supervised cell typing method outperformed all others in every aspect. Furthermore, supervised methods accuracy rates across cells with different transcript numbers were consistent, but accuracy rates were inconsistent across all genes, half genes, and 10% highly variable genes. SingleR and InSituType are the fastest (<2 min), followed by Seurat and SVM (~5 min), then CHETAH (~30 min), and SingleCellNet takes the longest time to run (~1 hr).

In our benchmark analysis of InSituType's unsupervised cell typing method parameters, we observed interesting trends. Tuning the number of clusters parameter presented a relatively accurate results when estimating close to the true number of clusters (13), and the model still performed relatively well (ARI > 0.8, see **Figure 6 – number of clusters on full gene dataset**) when estimating the number of clusters almost twice the amount of the true number of clusters (25 estimated clusters compared to 13 true clusters). When tuning the number of iterations, we found a consistent increase in performance as the number of iterations increased, and this trend seemed more apparent when cell typing on the full gene dataset compared to the half gene and 10% HVG. Manipulation of the cell subsample size in phase 1 did not show any apparent positive or negative performance in accuracy across the full gene, half gene, and 10% HVG; however increasing sample size does linearly trend positive in computational time. This may suggest making use of smaller subsample sizes in phase 1 to decrease computational time without concern for decrease in accuracy. However, this benchmark analysis for tuning parameters does not provide a significant estimate of any of the trends we observed. These results should rather be considered qualitatively and as an exploratory analysis of tuning parameters.

Impact and limitation

This benchmark analysis provided further insight into the performance of NanoString's InSituType cell typing model which is improving and developing for accurate use in spatial transcriptomic research. There are limitations of this analysis that should be considered. The benchmark analyses were performed on separate R Studio MacOS computing environments. It would be optimal to run these analyses on the same R computing environment. Additionally, the results for the unsupervised benchmark analysis could be improved with more benchmark results – thus providing larger sample sizes to see if there are stronger accuracy trends with respect to parameters. These limitations provide roadmaps for future work to improve this analysis. In addition to these limitations, the unsupervised benchmark analysis could be enhanced by investigating combinations of the parameters of interest and utilizing parallel computation for working with the large transcriptome datasets. We will be transitioning the code for this benchmark analysis to NanoString Technologies where they will have the option to further improve on the results as they see fit.

Team member contributions

Supervised methods: Ingrid Luo, Makalya Tang

Unsupervised methods: Howard Baek, Eliza Chai, Alexis Harris

Reference: (APA Format)

1. Insitutype: likelihood-based cell typing for single cell spatial transcriptomics. Patrick Danaher, Edward Zhao, Zhi Yang, David Ross, Mark Gregory, Zach Reitz, Tae K. Kim, Sarah Baxter, Shaun Jackson, Shanshan He, Dave Henderson, Joseph M. Beechem bioRxiv 2022.10.19.512902; doi: https://doi.org/10.1101/2022.10.19.512902
2. Pierre, L. T. (2022, October 5). Why spatial biology enhances spatial transcriptomics data. NanoString. Retrieved November 21, 2022, from https://nanostring.com/blog/why-spatial-biology/
3. NanoString. (2022, August 2). Spatial Transcriptomics Overview. NanoString. https://nanostring.com/blog/what-is-spatial-gene-expression/
4. CosMx SMI FFPE Dataset. (n.d.). NanoString. Retrieved November 21, 2022, from https://nanostring.com/products/cosmx-spatial-molecular-imager/ffpe-dataset/
5. Azure DevOps Services | Sign In. (n.d.). Spsprodcus2.Vssps.visualstudio.com. Retrieved November 21, 2022, from https://dev.azure.com/Nanostring/Gemini/_wiki/wikis/Gemini.wiki/1087/Nanopipeline-Documentation-(V.1)?anchor=uniform-manifold-approximation-and-projection-(umap)
6. Blog. (n.d.). NanoString. Retrieved November 21, 2022, from https://nanostring.com/blog/
7. Nature. Retrieved November 21, 2022, from https://www.nature.com/articles/s41598-019-41695-z#Fig3
8. Cell. June 06, 2019, from https://www.cell.com/cell/fulltext/S0092-8674(19)30559-8#secsectitle0025 (Seurat v3)
9. BMC Genome Biology. December 12, 2019, from https://genomebiology.biomedcentral.com/articles/10.1186/s13059-019-1862-5 (scPred)
10. SingleCellNet, github page from https://github.com/pcahan1/singleCellNet#introduction
11. bioRxiv. February 22, 2019, from https://www.biorxiv.org/content/10.1101/558908v1 (CHETAH)
12. Nature. December, 2018, from https://www.nature.com/articles/s41592-018-0229-2.epdf?author_access_token=5sMbnZl1iBFitATlpKkddtRgN0jAjWel9jnR3ZoTv0P1-tTjoP-mBfrGiMqpQx63aBtxToJssRfpqQ482otMbBw2GIGGeinWV4cULBLPg4L4DpCg92dEtoMaB1crCRDG7DgtNrM_1j17VfvHfoy1cQ%3D%3D (scVI)
13. Marx, V. Method of the Year: spatially resolved transcriptomics. Nat Methods 18, 9–14 (2021). https://doi.org/10.1038/s41592-020-01033-y