Supplement: 'Dhaka: Variational Autoencoder for Unmasking Tumor Heterogeneity from Single Cell Genomic Data'

Sabrina Rashid,[1] Sohrab Shah[2,3,4], Ziv Bar-Joseph[5], Ravi Pandya[6*]

[1] Computational Biology Department, Carnegie Mellon University, Pittsburgh, USA
[2] Department of Computer Science, University of British Columbia, Vancouver, Canada
[3] Department of Molecular Oncology, BC Cancer Agency, Vancouver, Canada
[4] Department of Pathology and Laboratory Medicine, University of British Columbia, Vancouver, Canada
[5] Machine Learning Department and Computational Biology Department, Carnegie Mellon University, Pittsburgh, USA
[6] Microsoft Research, Redmond, USA
*To whom correspondence should be addressed; E-mail: ravip@microsoft.com

1 Appendix

1.1 Software

We have developed a python package for the Dhaka variational autoencoder using the Keras module [1]. The package is released as open source (`https://github.com/MicrosoftGenomics/Dhaka`). Since this is a probabilistic encoding of the genomic data, often we need to do multiple warm starts of the encoder to select the best encoding. For example, if we are interested in identifying clusters, from each projected encoding we will compute the silhouette score, and select the encoding that maximizes the score [2]. We have used multiple warm starts only for the synthetic data analysis. We did not use multiple warm starts for the copy number and gene expression data. The number of warm starts is a user parameter for the package (5 in case of synthetic dataset). The Dhaka package can also perform gene selection, if needed. We have three options for selecting informative genes for analysis.

- Coefficient of variation (CV) score: CV of gene i with expression profile $g_i \in R^{1\times m}$ is defined as $CV_i = std(g_i)/mean(g_i)$. Here m is the total number of cells.
- Entropy En: $En_i = -sum(p_i log2(p_i))$. Here p_i is the estimated histogram from g_i.
- Average expression value $\bar{A}$: This is simply the average expression value of a particular gene across all cells.

The gene selection criteria and number of genes to be included in the analysis are both user parameters. We have used gene selection for the three RNA-Seq gene expression datasets, (5000 genes with $\bar{A}$ criteria). The variational autoencoder is robust to the dropout events, therefore we did not have to model the dropout events separately. The other parameters of the package are the number of the latent dimensions, learning rate, batch size, number of epochs, and clip norm of the gradient [1]. We have used the following values for these parameters, learning rate = 0.0001, batch size =50, clip norm =2, and number of epochs =5. We used the same parameter values for all the datasets analyzed in the paper. A sensitivity analysis on these parameters on the simulated dataset are summarized in Table S1. For each of the parameters we considered three possible values keeping the other parameter values fixed at the aforementioned values. We can see that the most sensitive parameters are the batch size and number of epochs. If we reduce the batch size to 20 we see a significant decrease in the performance, we can see similar effect in increasing the number of epochs

[1] Gradients will be clipped when their L2 norm exceeds this value. This parameter is used for the stability of the gradient descent algorithm.

in training. The learning rate and clip norm parameters have relatively stable ARI scores. For the largest dataset Oligodendroglioma, the software takes 105 seconds to train for 5 epochs (used in the paper) and 170 seconds if we continue to train for 10 epochs.

Learning rate	Batch size	Clip norm	Number of epochs	ARI
0.0001	50	2	5	.73
0.001	50	2	5	.71
0.01	50	2	5	.65
0.0001	100	2	5	.72
0.0001	20	2	5	.22
0.0001	50	1	5	.74
0.0001	50	.5	5	.65
0.0001	50	2	10	.67
0.0001	50	2	20	.26

Table S1. Sensitivity analysis of the Dhaka hyperparameters

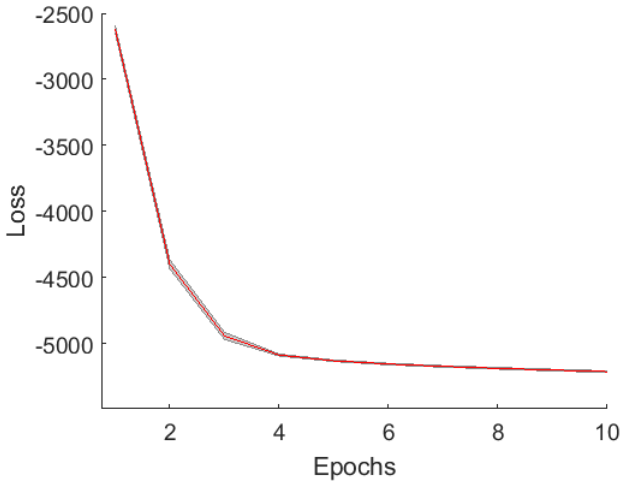


Fig. S1. Loss function plot from 50 independent trials of the Dhaka method on the Oligodendroglioma dataset. The low standard error in the loss function plot demonstrates the robustness of the Dhaka method. Even though we train for 5 epochs in the reported results, here we show loss function plot for upto 10 epochs to show the stability of the training method.

1.2 Runtime comparison

We have compared the runtime of the proposed method and all the comparing methods for the simulated dataset. For each method the time to compute the three dimensional projection given fixed parameters is reported in Table S2. All the methods were performed on a 32 GB 3.4 GHz windows machine. As can be seen from Table S2, although PCA is the fastest method it has the lowest ARI score. Dhaka is the second fastest with highest ARI score.

	Dhaka	PCA	t-SNE	ZIFA	SIMLR
ARI	.73	.16	.27	.58	.69
Runtime (s)	3.43	.20	3.57	501.23	17.18

Table S2. Runtime comparison on simulated dataset

1.3 Relative gene expression

The relative gene expression $Er_{i,j} = E_{i,j} - mean(E_{i,1,\ldots,n})$. Here i and j correspond to gene and cell, respectively.

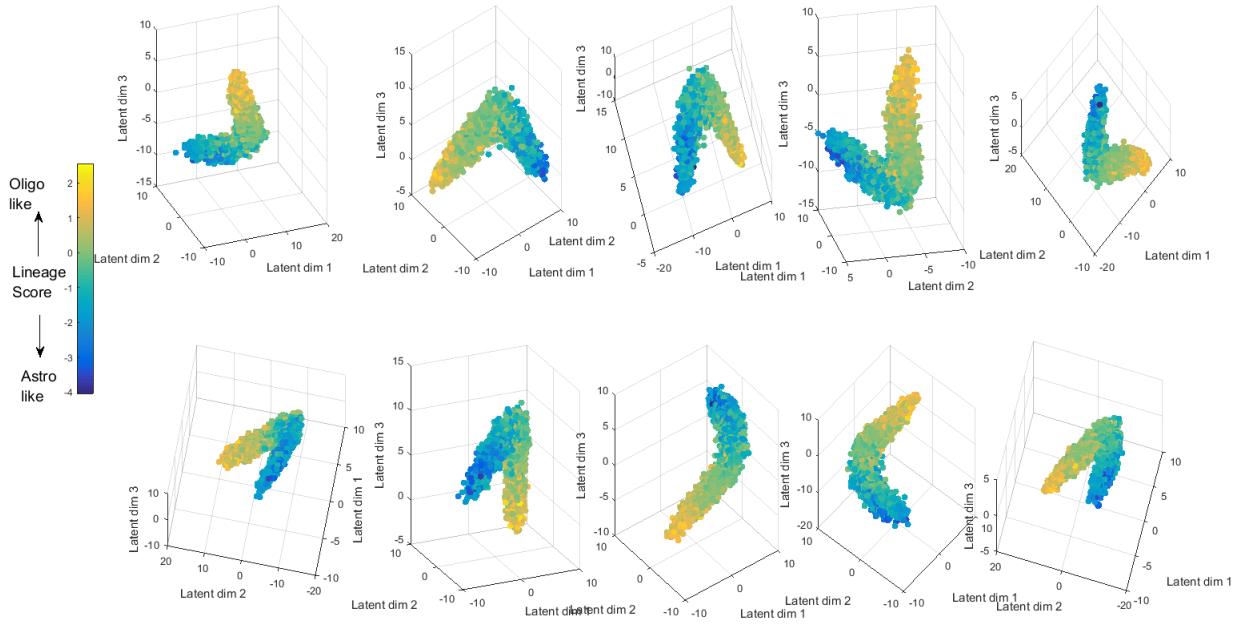


Fig. S2. Autoencoder projection from 10 independent runs on Oligodendroglioma dataset with signature genes. Same v-structure is captured in all of the runs.

1.4 Correlation score computation

Latent projections from different methods have very different axis range. Hence instead of computing Pearson correlation, Spearman rank correlation is chosen as performance evaluation metric. In clustering evaluation, metrics like nearest neighbor preservation, adjusted rand index, and silhoutte score are typically used as performance metric. However, in this paper the primary focus is to uncover evolutionary trajectory of tumor populations instead of just clustering. Hence, a correlation score with lineage/differentiation metrics are a better indicator of the algorithm performance in latent projection computation.

For 2D scoring metric such as in case of Oligodendroglioma (X= Lineage score, Y= Differentiation score), we computed 2D projections for all the competing methods along with the proposed autoencoder. Let us consider, $T \in R^{n\times 2}$ is the true scoring metric, and $P \in R^{n\times 2}$ is

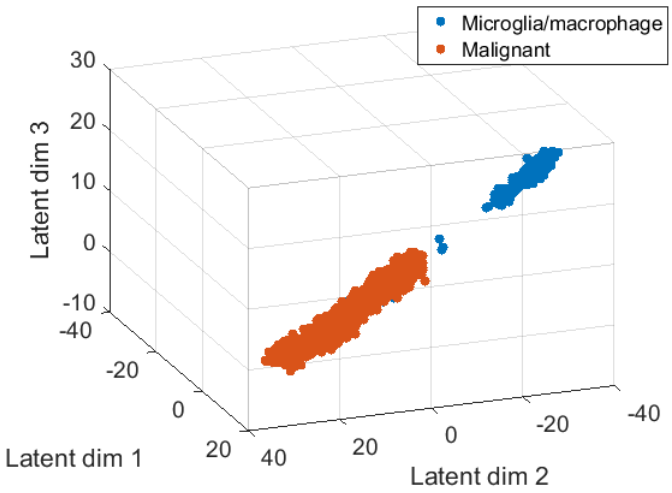


Fig. S3. Oligodendroglioma dataset with 5000 auto-selected genes. Autoenocder projection separating malignant cells from non-malignant microglia/macrophage cells.

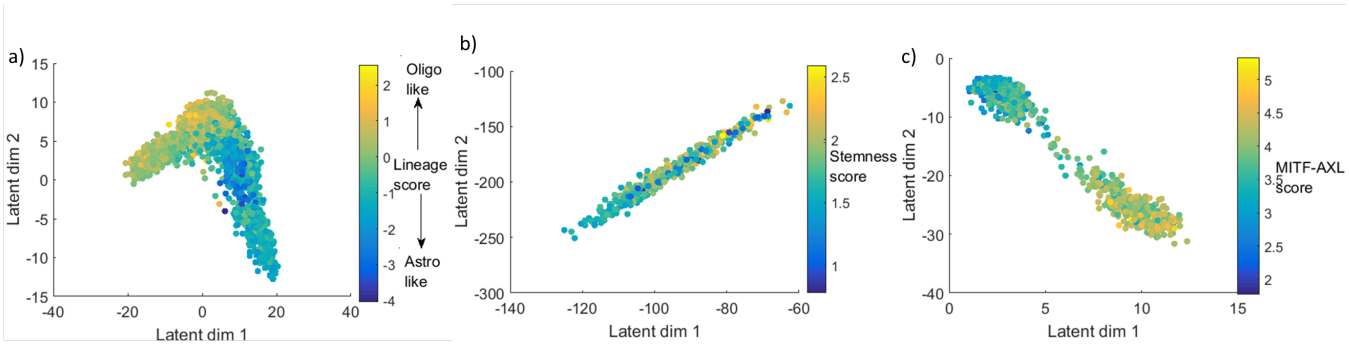
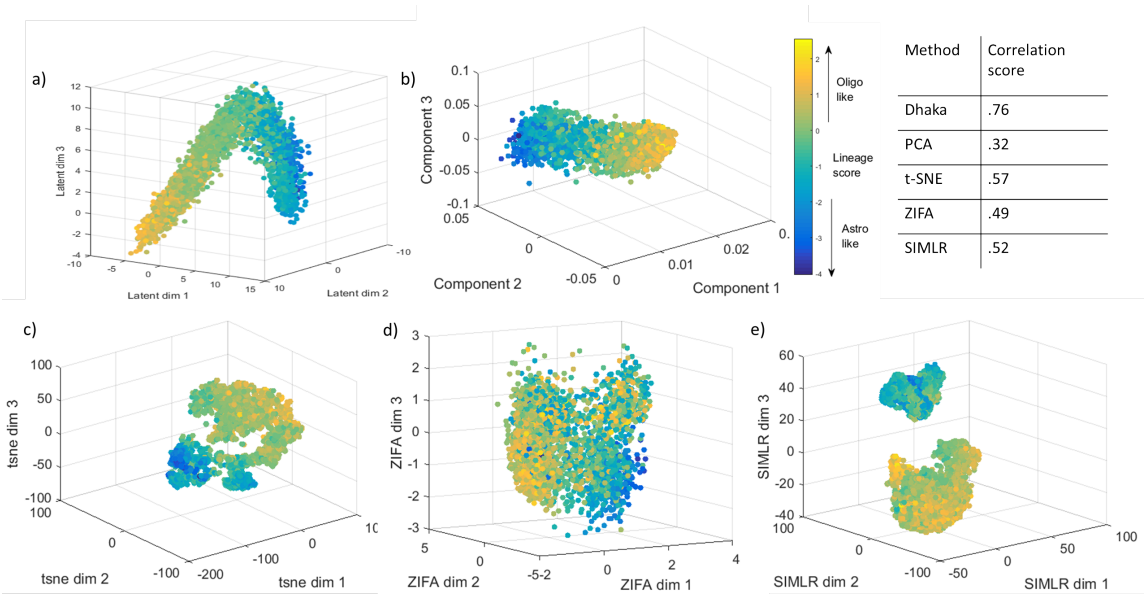


Fig. S4. Two dimensional projections obtained from Dhaka method. a) Oligodendroglioma b) Glioblastoma c) Melanoma. We can see that the v-structure of Oligodendroglioma dataset is preserved in 2D projection as well (Correlation score 0.61). Even though the linear trajectory is preserved in the Glioblastoma dataset, the correlation score decreased from 3D case to 2D case (0.61 from 0.72). For Melanoma dataset, we still see a good separation of the two tumor subpopulation based on MITF-AXL score. The correlation score decreases slightly from 0.68 in 3D projection to 0.66 in 2D projection.



Method	Correlation score
Dhaka	.76
PCA	.32
t-SNE	.57
ZIFA	.49
SIMLR	.52

Fig. S5. Comparison of variational autoencoder with PCA, t-SNE, ZIFA, and SIMLR on Oligodendroglioma dataset with 265 signature genes. a) Autoencoder, b) PCA, c) t-SNE, d) ZIFA, e) SIMLR projections colored by the lineage score. The Spearman rank correlation scores with the scoring metric (lineage and differentiation score) and the autoencoder and the reported method projections are shown in tabular form. We can clearly see that Dhaka preserves the original scoring metric the best.

the learned 2D projection. Then the Spearman correlation coefficient will be a CC=2 × 2 matrix. Here $CC(i,j) = Corr(T(:,i), P(:,j))$. Then the overall correlation score is computed as $CS = max\left(\sqrt{\frac{1}{2}\sum_i^2 CC(i,i)^2}, \sqrt{\frac{1}{2}\sum_{i\neq j}^2 CC(i,j)^2}\right)$.

In case of 1D scoring metric such as Stemness score in Glioblastoma and MITF-AXL score in Melanoma, we computed correlation coefficient of 1D scoring metric with each dimension of the 3D projection, i.e., $CC \in \mathbb{R}^{1\times 3}$. Then CS is simply set as $CS = max(CC)$.

1.5 Dropout analysis

To evaluate robustness of the method to dropout genes, we introduced additional dropouts in the Oligodendroglioma dataset. Each gene in the dataset have fraction of dropout ranging from 0 to 0.8 (Please see Supporting Fig. S6a). To introduce additional dropouts, for each gene we randomly select a subset of cells, for example in the 20% case we randomly select 20% of the cells for each gene. Then we make the expression value of these cells for that gene 0. After artificially introducing 20% more dropout, we should expect all the genes will have minimum dropout fraction of .2. Hence we see that now the histogram is shifted to the right from 0 to 0.2. We can see similar scenario for 30% and 50% case. Note that since we are randomly selecting cells to be dropped out, some of the selected cells might already have 0 expression values. Hence when we introduce 50% additional dropout, the histogram actually shifts to 0.4 instead of 0.5. As can be seen from the Fig. S6 up to 30% additional dropout, the autoencoder can still retain the v-structure. At 50%, we lose the v-structure, but the method can still separate oligo-like and astro-like cells even with this highly sparse data.

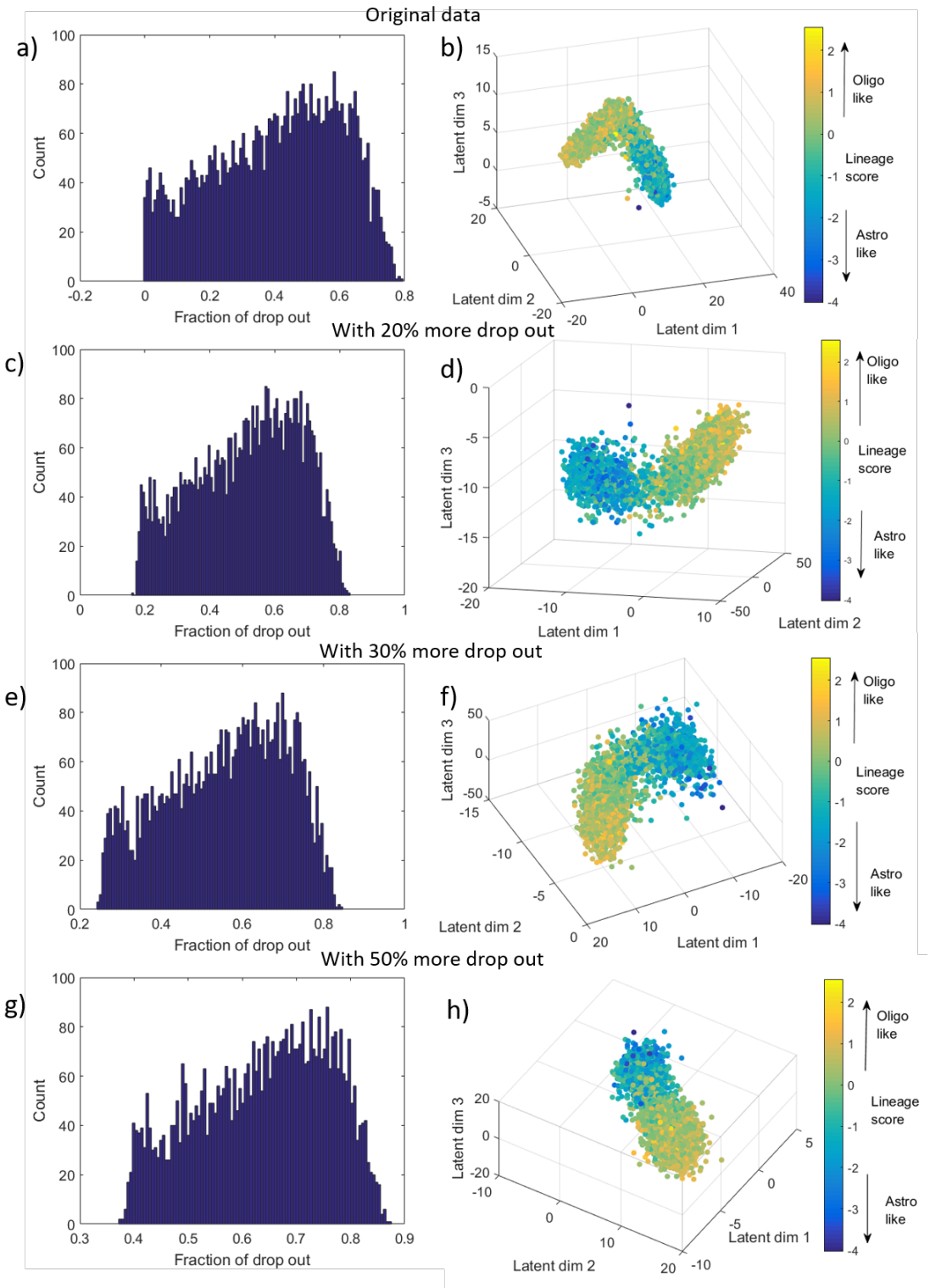


Fig. S6. Robustness analysis with Oligodendroglioma. a,c,e,g Histogram of dropout fraction in each gene after forcing 0%, 20%, 30%, and 50% more genes to be dropped out. b,d,f,h corresponding autoencoder projection of the data. We can see that up to 30%, the autoencoder can correctly identify v-structure. Beyond that the autoencoder loses the v-structure but still shows good separation between oligo-like and astro-like cells.

1.6 t-SNE, ZIFA, and SIMLR implementation details

To compute t-SNE projection, first the input data was pre-processed through PCA. PCA reduced the dimensions to 50. Then t-SNE was performed on the reduced dimension dataset. The perplexity parameter was set at 20. We have tested perplexity parameters $\{10, 20, 30, 40, 50\}$ and 20 gave the best projection for our discussed datasets. We have used MATLAB implementation of basic t-SNE (`https://lvdmaaten.github.io/tsne/`).

The SIMLR MATLAB implementation was used in the paper (`https://github.com/BatzoglouLabSU/SIMLR`). The input read count datasets were log10 transformed as required by the package. The SIMLR package requires an estimated number of clusters to compute the similarity matrix. The estimated number of clusters were computed using $Estimate_Number_of_Clusters_SIMLR.m$ function of the package.

SIMLR also uses t-SNE for dimensionality reduction following their similarity matrix estimation. Since t-SNE is stochastic in nature, we get slightly different output in each run. Hence, both t-SNE and SIMLR were ran 10 times on each dataset and average ARI/Correlation score were reported in the result subsection.

The python implementation of ZIFA package was used from `https://github.com/epierson9/ZIFA`.

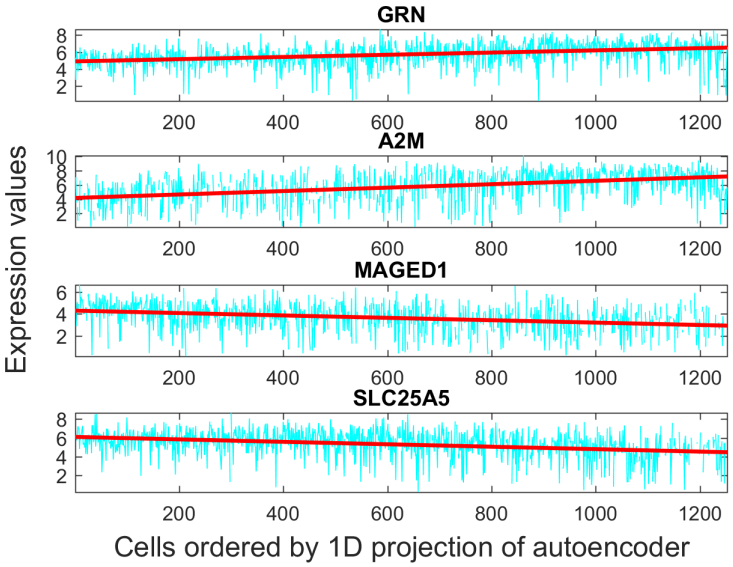


Fig. S7. Known marker genes for Melanoma MITF-AXL program.

1.7 Analysis of Astrocytoma data

The Astrocytoma dataset contains a total of 6341 cells with about $23K$ genes, among which 5097 are malignant cells. Astrocytoma is another type of brain tumor and this is a followup dataset from the Oligodendroglioma. Hence, we performed same analysis as for Oligodendroglioma. The non-malignant microglia/macrophage cells were clearly separated from the malignant cells (Fig. S8a) in this dataset too. The authors did not compute differentiation and lineage metric for this dataset, but

did mention that most of the cells fall in the intermediate state. When we fed the expression profile of the malignant cells to the autoencoder, it correctly placed most of the cells near the bifurcation point of the v-structure (Fig. S8b). For reference, we have also showed the Oligodendroglioma cells in the same plot colored by their differentiation score.

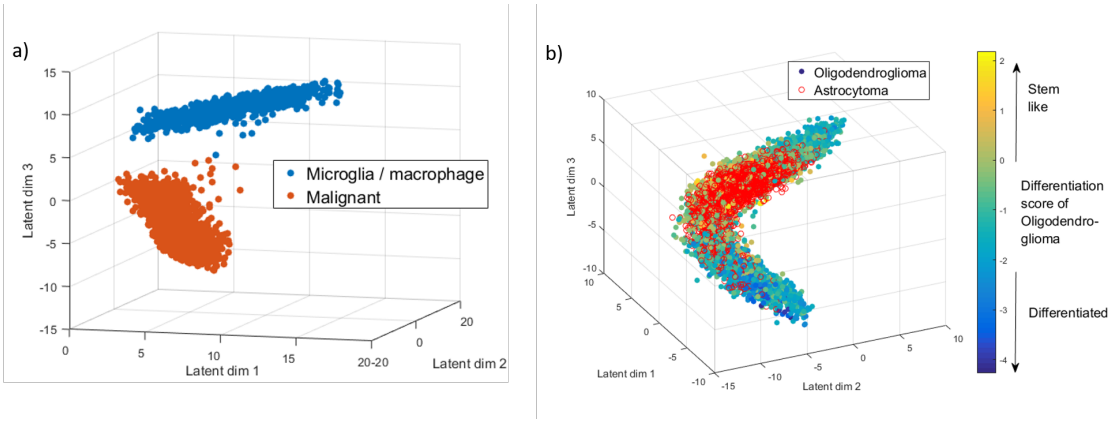


Fig. S8. Astrocytoma dataset. a) Autoencoder output of Astrocytoma dataset with 5000 autoselected genes separating malignant cells from microglia/macrophage cells. b) Autoencoder output from relative expression profile of malignant Astrocytoma cells(red) along with malignant Oligodendroglioma cells.

References

1. F. Chollet, "keras," *GitHub repository*, 2015.
2. L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*, vol. 344. John Wiley & Sons, 2009.
3. C. Trapnell, D. Cacchiarelli, J. Grimsby, P. Pokharel, S. Li, M. Morse, N. J. Lennon, K. J. Livak, T. S. Mikkelsen, and J. L. Rinn, "The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells," *Nature biotechnology*, vol. 32, no. 4, pp. 381–386, 2014.

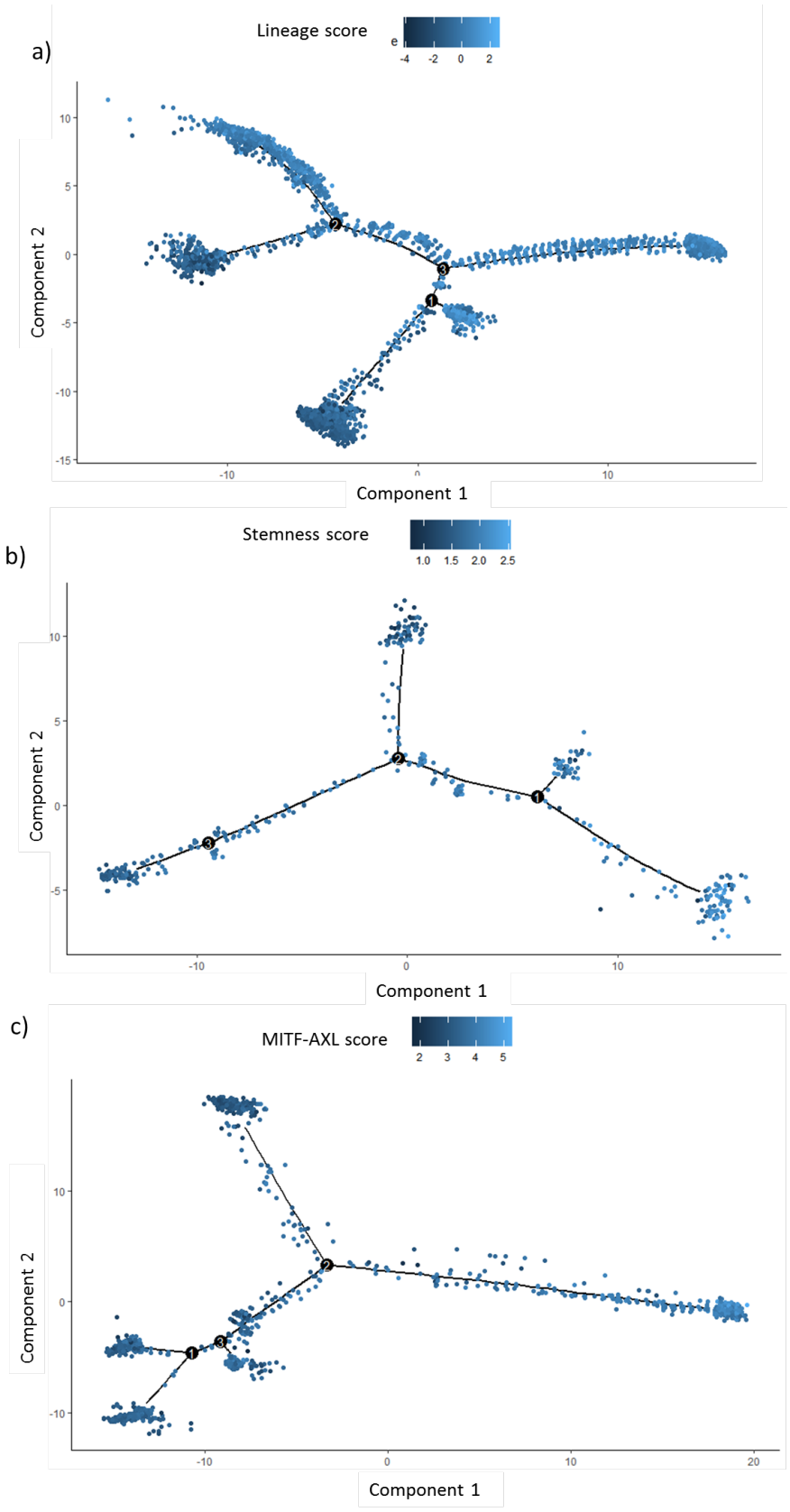


Fig. S9. Monocle [3] pseudotime ordering of a) Oligodendroglioma, b) Glioblastoma, c) Melanoma datasets. As can be seen from the figure, the pseudotime orderings from Monocle fail to capture the biological trend of the data. Monocle divides the cells in each dataset to multiple branches but none of the branches show significant correlation with the underlying tumor subpopulation and/or the scoring metrics. Monocle obtained correlation score of 0.32 (Oligodendroglioma), 0.23 (Glioblastoma), and 0.27 (Melanoma). The 'R' package of Monocle was installed through bioconductor.