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

Sabrina Rashid,[1,2] Sohrab Shah[3,4,5], Ravi Pandya[2*]

[1] Computational Biology Department, Carnegie Mellon University, Pittsburgh, USA
[2] Microsoft Research, Redmond, USA
[3] Department of Computer Science, University of British Columbia, Vancouver, Canada
[4] Department of Molecular Oncology, BC Cancer Agency, Vancouver, Canada
[5] Department of Pathology and Laboratory Medicine, University of British Columbia, Vancouver, Canada

*To whom correspondence should be addressed; E-mail: ravip@microsoft.com

Intra-tumor heterogeneity is one of the key confounding factors in deciphering tumor evolution. Malignant cells will have variations in their gene expression, copy numbers, and mutation even when coming from a single tumor. Single cell sequencing of tumor cells is of paramount importance for unmasking the underlying the tumor heterogeneity. However extracting features from the single cell genomic data coherent with the underlying biology is computationally challenging, given the extremely noisy and sparse nature of the data. Here we are proposing 'Dhaka' a variational autoencoder based single cell analysis tool to transform genomic data to a latent encoded feature space that is more efficient in differentiating between the hidden tumor subpopulations. This technique is generalized across different types of genomic data such as copy number variation from DNA sequencing and gene expression data from

RNA sequencing. We have tested the method on two gene expression datasets having 4K to 6K tumor cells and two copy number variation datasets having 250 to 260 tumor cells. Analysis of the encoded feature space revealed subpopulations of cells bearing distinct genomic signatures and the evolutionary relationship between them, which other existing feature transformation methods like t-SNE and PCA fail to do.

Introduction

Tumor cells comes from a highly heterogeneous landscape. In typical cancer progression, we often see prolonged clinically latent period with ongoing mutations, environment change, short generation time, and so on (*1*). Along this process the tumor cells will acquire various genomic properties, such as changes in gene expression, copy numbers, mutation, and so on (*2*), (*3*). Because of such genomic variance, we often see multiple subpopulations of cells connected in a evolutionary structure within a single tumor. The goal of effective cancer treatment is to treat all malignant cells without harming the originating host tissue. Hence, clinical approach should be adaptive to such underlying evolutionary structure. For that, clear understanding of the evolutionary process that separates malignant cells from their normal cell of origin is necessary. We also need to learn whether the ancestral clones of the tumor eventually disappear (chain like evolution) or several genotypically different clones of cells evolve in parallel (branched evolution) (*1*). When we apply treatment to a cancer patient, these cells will respond differently, which often leads to therapy resistance and possible cancer recurrence. Therefore, it is crucial that we characterize the nature of their hidden subpopulations and the underlying evolutionary structure. In earlier literature, bulk sequencing has been used to investigate tumor evolution (*4*), (*5*). In bulk sequencing a number of cells are sequenced together, which averages out the genomic characterisitics of the tumor cells that make them so heterogenous. Hence in recent

years single cell sequencing has emerged as a useful tool to study such cellular heterogeneity (*6*), (*7*), (*8*), (*9*).

But at the same time single cell sequencing comes with its own computational and experimental challenges. Because of the low quantity of genetic material available in single cell sequencing, the resulting data is often very noisy and sparse with many dropout events (*10*). This applies to both gene expression data from RNA-sequencing and copy number and mutation data from DNA sequencing. For high fidelity estimation of copy number variation and mutation, very deep sequencing is required, which is not often possible due to financial constraints (*11*). Given the nature of the data, it remains challenging to identify meaningful features that can characterize the single cells in terms of their clonal identity and differentiation status. Many of the existing feature transformation techniques that are usually applied to genomic data fail to capture these type of information. For example, t-SNE (*12*) and diffusion maps (*13*) are very efficient in projecting data to a low dimensional space. But quite often they show segregation between different samples or tumor type instead of characterizing the underlying biology. There have also been several clustering algorithms proposed for single cell RNA-seq data. For example, SNN-cliq (*14*) constructs a shared k-nearest neighbor graph across all cells and then finds maximal cliques as clusters. PAGODA is another clustering approach (*15*) that relies on prior set of annotated genes to find transcriptomal heterogeneity. Although these methods can distinguish between different groups of cells in a dataset, these clustering approaches are often restricted when it comes to deciphering the relationship between the detected clusters. Also these approaches are usually developed with a focus on only one type of genomic data such as gene expression data. Here we are interested in finding a more generalized approach that will work on different types of single cell genomic profiles.

Previously neural networks have also been used to reduce the dimensions of the single cell gene expression data (*16*), (*17*). For example in Lin *et al.* (*16*), the authors used some prior

knowledge like protein-protein interaction or protein-DNA interaction to learn the architecture of the neural networks and subsequently project the data to a lower dimensional feature space. Here we are using neural networks in an unsupervised manner to learn a latent representation of the data. We know neural networks are very efficient in learning their own encoding of arbitrary input data. Here we are using a variational autoencoder that combines Bayesian inference with such unsupervised deep learning, so that a probabilistic encoding of the input data can be learned. This encoding is generalized across different types of genomic data. In this paper we have shown results from two single cell RNA seq datasets. The variational autoencoder projected the cells to a three dimensional latent space. The cells were automatically arranged according to their differentiation status and clonal signatures in the latent space. We have also tested two single cell copy number variation datasets from very low depth (0.05X) DNA sequencing. The autoencoder successfully finds two subpopulations of malignant cells and upon projecting the data from two xenograft samples from two different time points, it clearly demonstrated that upon transplanting the tumor from one mouse to the next, only one subpopulation progressed to the next time point. These results corroborates the effectiveness of the Dhaka method in learning the underlying biology of the cancer samples.

The paper is organized by first discussing the variational autoencoder method and then followed by detailed discussion of the results on the gene expression and copy number variation data.

Method

Variational autoencoder

The variational autoencoder used in the project is adapted from (*18*). Autoencoders are multi-layered perceptron neural networks that sequentially deconstruct data (x) into latent representation (z) and then use these representations to sequentially reconstruct outputs that resemble the

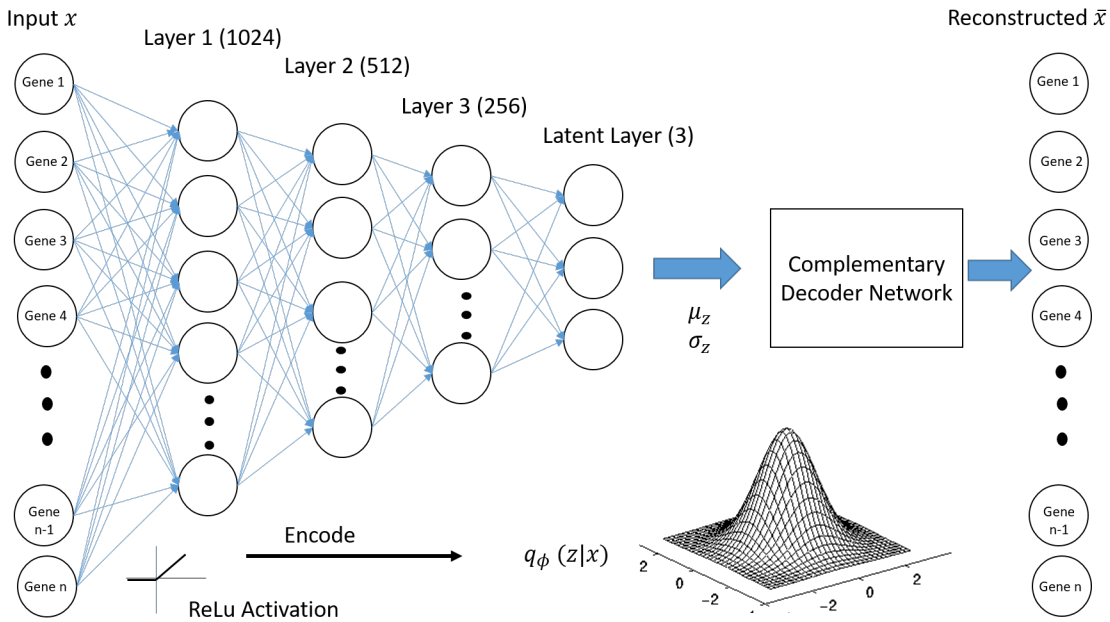

Figure 1: Structure of the variational autoencoder used in Dhaka.

inputs. The main advantage of this approach is that the model learns its own definition of the data in a completely unsupervised manner. In variational autoencoder (VAE) the concept of this unsupervised deep learning is merged with that of Bayesian inference. Instead of generating an unconstrained representation of the data we impose a regularization constraint. We assume that the latent representation is coming from a probability distribution, in this case a spherical Gaussian ($N(0, I)$). The intuition behind such representation is that these heterogeneous cells are actually result of some underlying biological process governing them. These processes are modeled here as distribution over latent space, each having their distinct means and variances. Hence the autoencoder actually encodes not only the mean (μ_z) but also the variance(σ_z) of this Gaussian distribution. The latent representation (z) is then sampled from the learned posterior distribution $q_\phi(z|x) \sim N(\mu_z, \sigma_z I)$. Here ϕ is the parameters of the encoder network (such as biases and weights). The learned latent representation is then passed through the decoder network to reconstruct the input $\bar{x} \sim p_\theta(x|z)$, θ is the parameters of the decoder network. Although the

model is trained holistically, we are actually interested in the latent representation z of the data.

Learning

To learn the parameters ϕ, θ, we need to maximize $log(p(x|\phi, \theta))$, the log likelihood of the data points x, given the model parameters. The marginal likelihood $log(p(x))$ is the sum of a variational lower bound (*18*) and the Kulback-Leibler (KL) (*19*) divergence between the approximate and true posteriors.

$$log(p(x)) = L(\phi, \theta; x) + D_{KL}(q_\phi(z|x)||p_\theta(z|x))$$

The L can be decomposed as following:

$$L(\phi, \theta; x) = E_{z \sim q_\phi(z|x)}[log(p_\theta(x|z))] - D_{KL}(q_\phi(z|x)||p_\theta(z))$$

The first term can be viewed as the typical reconstruction loss intrinsic to all autoencoders, the second term can be viewed as the penalty for forcing the encoded representation to follow the Gaussian prior. We then use 'RMSprop'[1] technique to minimize $-L$. Detailed derivation of the loss computation can be found in (*18*).

Structure

Fig. 1 shows that structure of the autoencoder used in this paper. The input layer will consist of nodes equal to the number of genes we are analyzing for each cell. We are introducing non-linearity in the structure by using Rectified Linear unit(ReLu) activation function. In the result section we show performance comparison between different structures and their corresponding runtime.

We have used three dimensional latent space in this paper. But, the number of latent dimensions is a parameter in Dhaka that can be changed as per users' choice. We initially started to

[1]A variant of stochastic minibatch gradient descent, where the learning rate for a weight is divided by the running average of the magnitudes of recent gradients for that weight (*20*).

use three dimension for visualization convenience but later found that with only three dimensions the autoencoder was able to unmask underlying tumor heterogeneity. In future work, we will explore more latent dimensions. The current structure can handle datasets with atleast 1000 genes per cell. If we want to explore datasets with fewer genes than that, we will reduce the number of nodes in the intermediate layers. All the datasets reported in this paper had around 20k genes and the reported structure (Fig. 1) was sufficient for them.

Results

Quantitative validation with simulated dataset

To quantify the effectiveness of the Dhaka method with that of the state-of-the-art t-SNE (*12*) method and PCA dimensionality reduction method (*21*), we have used 2 simulated datasets with 3K genes and 500 cells. We have followed the method described in (*22*) to create the simulated datsets. In each dataset, the cells come from five different clusters having 100 cells each. Both the datasets contains a mix of noisy genes along with actual genes representative of each cluster. In the first dataset, there are 500 noisy genes among 3000 total genes. We have used the Bayesian Information criterion (BIC) from a Gaussian Mixture Model to estimate the number of clusters from the projected feature space. We have also computed adjusted random index (ARI) metric to quantify the quality of clustering. Fig 2 (first column) shows the result of autoencoder, PCA, and t-SNE projection of the simulated data. We can see that only the Dhaka autoencoder can correctly identify all 5 clusters (Fig 2g), whereas the PCA method identifies only 1 and t-SNE identifies only 3. The ARI is also highest for the autoencoder projection with 0.98 compared to PCA (0.0, since only one cluster was detected), and t-SNE (0.61).

Next we increased the number of noisy genes to 2500 out of 3000. Now, the Dhaka autoencoder can identify 4 clusters in the dataset, whereas t-SNE can only identify 2. The PCA projection also identified 4 clusters but with a much lower ARI score of .16 (Fig 2 second

Table 1: Comparison between structures of autoencoders and t-SNE

	VAE structure 1	VAE structure 2	t-SNE
ARI	0.71	0.5	0.27
Runtime (sec)	3.43	2.13	3.57

column). Although the autoenoder drops one cluster it can still differentiate others very well resulting in a high ARI score of 0.71. This corroborates the robustness of the methods used in Dhaka.

We have also compared two different structures of the autoencoder (structure 1: $Input \rightarrow 1024 \quad nodes \rightarrow 512 \quad nodes \rightarrow 256 \quad nodes \rightarrow 3 \quad latent \quad dims$, structure 2: $Input \rightarrow 1024 \quad nodes \rightarrow 3 \quad latent \quad dims$ in terms of ARI and runtime (Table 1) on the sythetic data with 2500 noisy genes. The VAE structure 1 (Fig. 1) gives the best ARI score and both the VAE structures are faster than t-SNE.

As we know, drop outs are a major issue in single cell RNA-seq data analysis. Hence we also tried to analyze the robustness of the Dhaka method to drop out genes. We have simulated the drop out effect by randomly picking a percentage of genes for each cell to be dropped out. The set of genes dropped out in each cell is not necessarily same since these are picked randomly. Fig. 3 shows the performance of the Dhaka method in terms of ARI as we gradually increase the percentage of drop out genes in each sample. We can see that autoencoder is the most robust when compared to t-SNE and PCA. Both t-SNE and PCA fail to detect more than one cluster when more than 5% drop out genes were introduced in the sample. The dataset we used here is the synthetic dataset with 2500 noisy genes out of 3000 genes.

Gene expression data

We have tested the method on two single cell RNA-seq tumor datasets: i) oligodendroglioma (*7*) and ii) astrocytoma (*6*).

In the oligodendroglioma dataset the authors performed single-cell RNA-seq from six un-

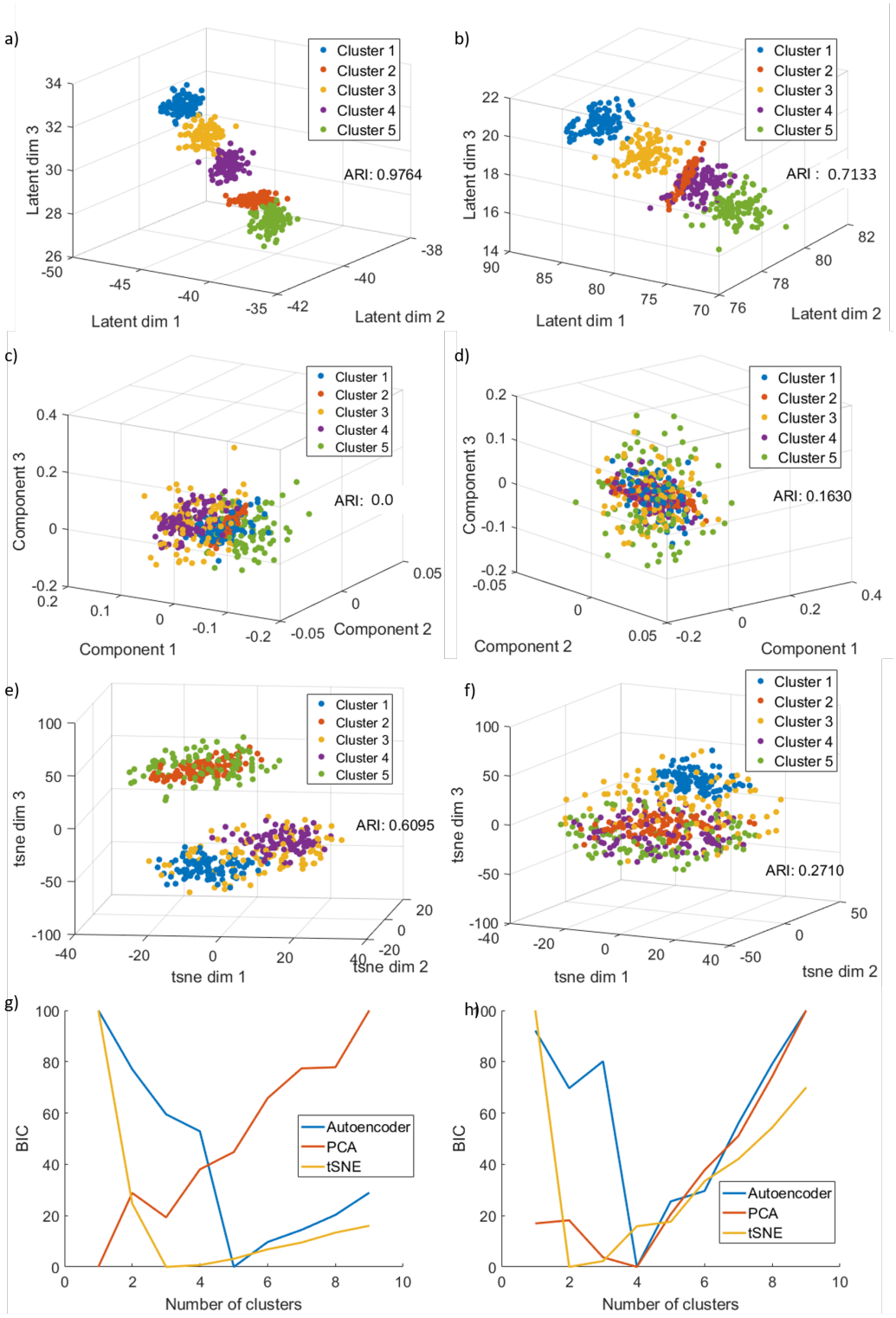

Figure 2: Comparison of the Dhaka method with t-SNE and PCA on two simulated datasets. First column: result on simulated dataset with 500 noisy genes (17% of total genes). Second column: result on simulated dataset with 2500 noisy genes (83% of total genes). a),b) Autoencoder output. c),d) t-SNE output. e),f) PCA output. g),h) Plot of BIC calculated from fitting Gaussian Mixture Model to the data to estimate number of clusters. The number with lowest BIC is considered as the estimated number of clusters in the data.

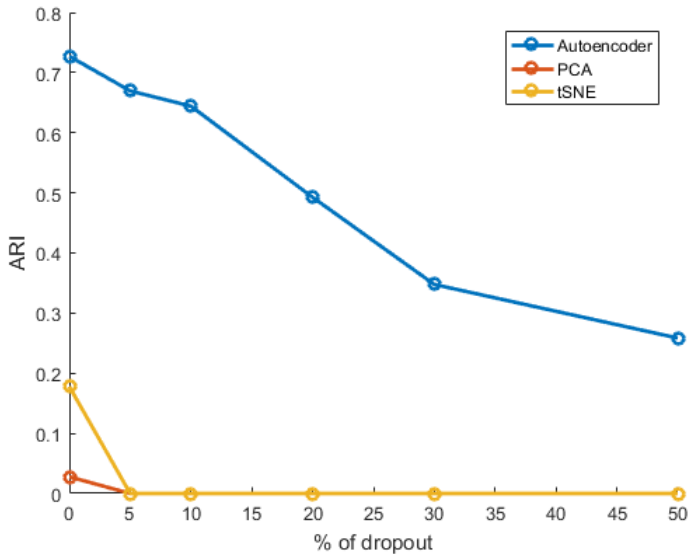

Figure 3: Robustness to drop out genes in single cell expression data.

treated oligodendroglioma tumors containing 4347 cells and about $23K$ genes. The dataset is comprised of both malignant and non-malignant cells. Copy number variations (CNV) were estimated from the $log2$ transformed RPKM expression data. The authors then computed two metrics, lineage score and differentiation score by comparing 265 signature gene CNV profiles with that of a control gene set. Based on these metrics, the authors showed that the malignant cells have two subpopulations, oligo-like and astro-like, and they share a common lineage. The analysis also showed the state of differentiation of each cell.

In Dhaka we fed the RNA-seq expression (with $log2$ transformation) directly to the autoencoder, skipping the CNV analysis. With only three latent dimensions the algorithm successfully separated malignant cells from non malignant microglia/macrophage cells(see Fig. 4a). We next analyzed the malignant cells only, to identify the different subpopulations and the relationship between them. We fed the relative expression values ($Er_{i,j}$, see appendix) to the autoencoder. Fig. 4b-c) shows the projected autoencoder output, where we see two distinct subpopulations originating from a common lineage, thus recapitulating the claim stated by the original paper,

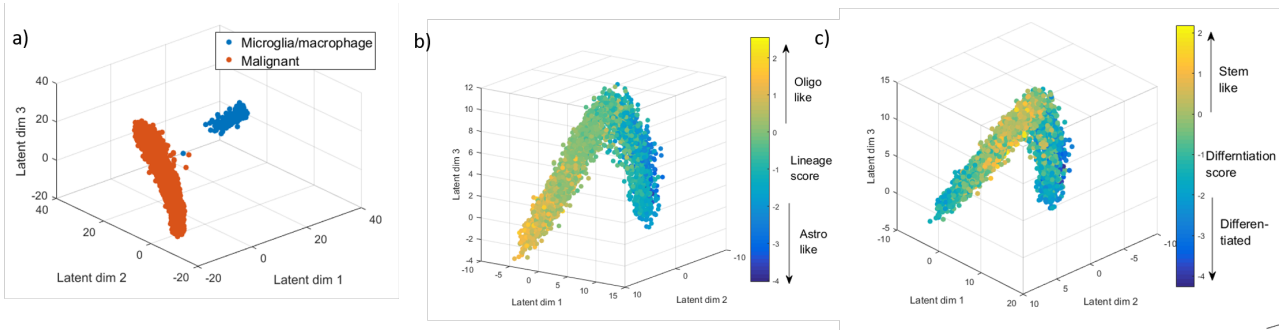

Figure 4: Oligodendroglioma dataset.a) Autoenocder projection separating malignant cells from non malignant microglia/macrophage cells. b)-c) Autoencoder output from relative expression profile of malignant cells using 265 signatures genes. b) Each cell is colored by their assigned lineage score which differentiates the oligo-like and astro-like subpopulations. c) Each cell is colored by their assigned differentiation score, which shows that most stem like cells are indeed placed near the bifurcation point.

without any prior for lineage or differentiation metric. The autoencoder was not only able to separate the two subpopulations, but also to uncover their shared glial lineage. To compare the results with the original paper, we have plotted the scatter plot with color corresponding to lineage score (Fig. 4 b)and differentiation score (Fig. 4c) from (*7*). We can see from the figure that the autoencoder can separate oligo-like and astro-like cells very well by placing them in opposite arms of the v-structure. Also we can see from figure Fig 4c that most of the cells with stem like property are placed near the bifurcation point of the v-structure. Fig. 4b-c are computed with the only the mentioned 265 signature genes.

Next we tried to identify whether similar structure can be learned from auto selected genes, instead of using signature genes from prior knowledge. Fig. 5a shows the autoencoder projection using 5000 auto-selected genes based on $\bar{A}$ score (see appendix). As we can see from Fig. 5a, the autoencoder can learn similar structure without needing prior knowledge about signature genes. We have also compared the autoencoder output to the t-SNE and PCA methods (Fig. 5b-c). The PCA method can separate the oligo-like and astro-like structure to some extent, but their separation is not as distinct as the autoencoder output. The t-SNE method can recover

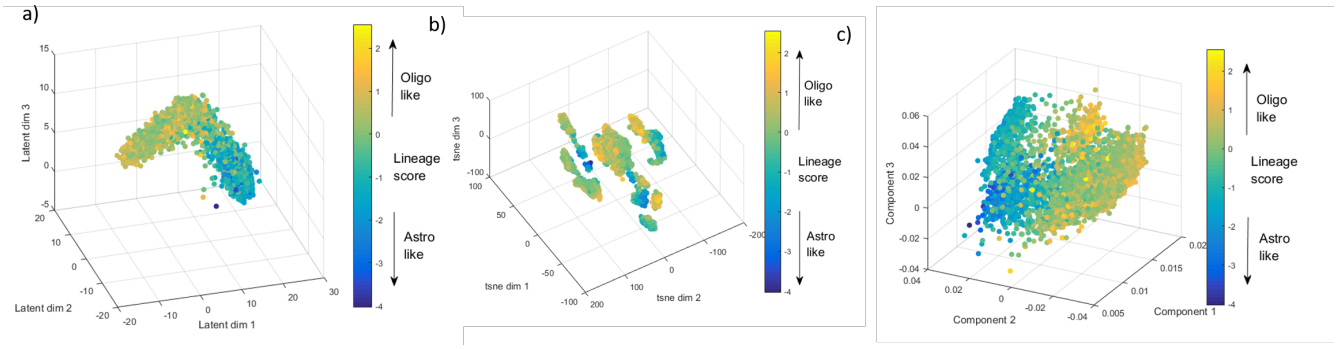

Figure 5: Comparison of variational autoencoder with t-SNE and PCA. a) Autoencoder output using 5000 auto-selected genes colored by lineage score. b) t-SNE projection colored by lineage score. c) PCA projection colored by lineage score.

clusters of cells from the same tumor but completely fails to identify the underlying lineage and differentiation structure of the data.

We have also tested robustness of the autoencoder with respect to drop out genes in the oligodendroglioma dataset. Fig. 6a shows the histogram of fraction of dropout in the auto selected 5000 genes. The fraction of drop out is calculated as the ratio of number of zero value cells for that particular gene to the total number of cells. The $\%$ of drop out ranges from 0 to 80% (Fig. 6a). Fig. 6c,e,g shows the histogram after artificially forcing 20%, 30%, and 50% more genes to be dropped out. As can be seen from Fig. 6g minimum $\%$ of dropout is lower than 50%, this happens because in some cases the randomly picked genes will already have zero value in the original data. Despite that after artificially injecting drop outs the histograms are considerably shifted to the left (Fig. 6c,e,g). Fig. 6b,d,f,h shows the projection of autoencoder after adding 0%, 20%, 30%, and 50% more drop out genes, respectively. We can see that up to 30%, the autoencoder can retain the v-structure even though the cells are a bit more dispersed. At 50% we lose the v-structure, but it can still make a good separation between the oligo-like and astro-like cells even with this highly sparse data. This actually further corroborates the robustness of the Dhaka autoencoder method.

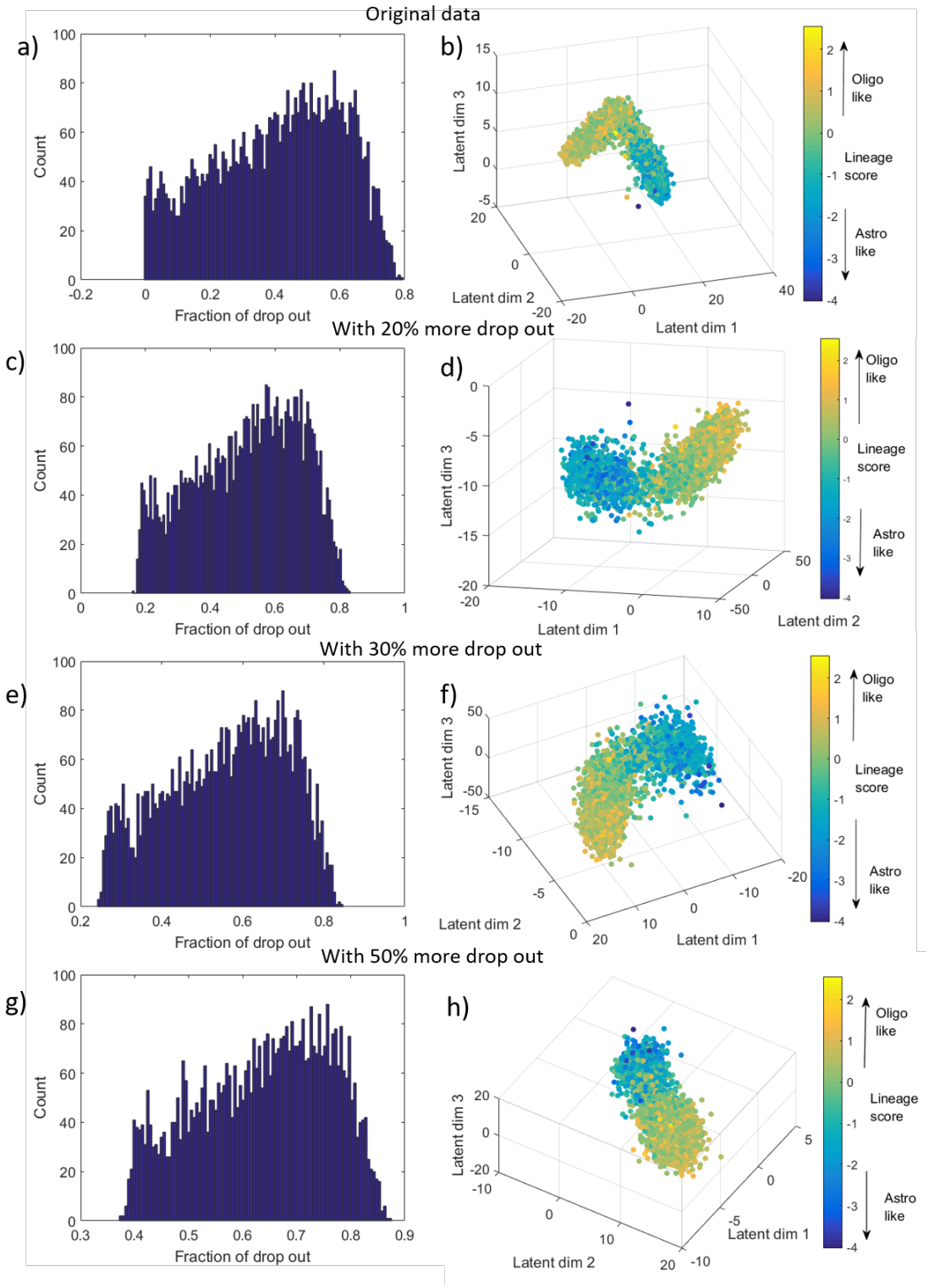

Figure 6: Robustness analysis with oligodendroglioma. a,c,e,g Histogram of drop out fraction in each gene after forcing 0%, 20%, 30%, and 50% more genes to be dropped out. b,d,f,h corresponding autoencoder projection of the data. We can see that upto 30%, the autoencoder can correctly identify v-structure. Beyond that the autoencoder loses the v-structure but still shows good sepration between oligo-like and astro-like cells.

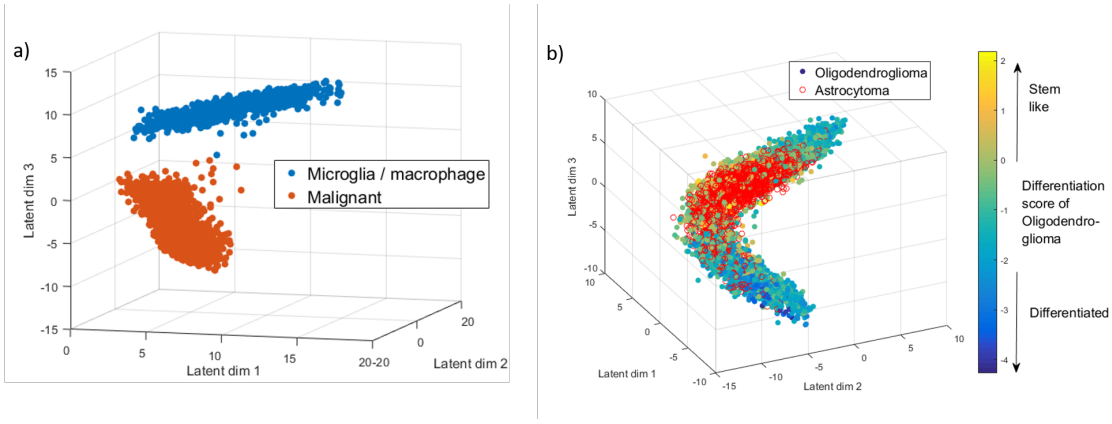

Figure 7: Astrocytoma dataset. a) Autoencoder output of Astrocytoma dataset with 5000 autoselected genes separating malignant cells from microglia/macrophage cells. b) Autoencoder output from relative expression profile of malignant astrocytoma cells(red) along with malignant oligodendroglioma cells.

The next dataset we analyzed is the astrocytoma dataset from the same group (*6*). The dataset contains a total of 6341 cells with about $23K$ genes, among which 5097 are malignant cells. We performed same analysis as for oligodendroglioma. The non-malignant microglia/macrophage cells were clearly separated from the malignant cells (Fig. 7a) 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. 7b). For reference, we have also showed the oligodendroglioma cells in the same plot colored by their differentiation score.

Copy number variation data

To test the universality of the method we also tested Dhaka with copy number variation data. We have copy number profiles from two xenograft breast tumor samples (xenograft 3 and 4) (*8*). xenograft 4 can be considered as the later time point of the xenograft 3. xenograft 3 has 260 cells and xenograft 4 has 254 cells. Both of the datasets have around 20K genomic bin count.

These cells are sequenced at a very low depth 0.05X. The authors estimated the copy number states using a hidden Markov model (*23*). When we fed the copy number profile of the xenograft 3 sample to the autoencoder, it identified 1 major cluster of cells and 1 minor cluster of cells (Fig 8a). The identified clusters also agree with the phylogenetic reconstruction analysis in the original paper. Fig. 8b shows the copy number profiles of cells organized by phylogenetic analysis. Even though the copy number profiles are mostly similar in most parts of the genome, we do see that there is a small number of cells that have two copies (as opposed to one in majority of cells) of the genes (marked by red circle) in x chromosome. The autoencoder was able to correctly differentiate the minor cluster of cells from the rest. Next we analyzed the xenograft 4 samples. The projected autoencoder output showed only 1 cluster which aligns itself with the major cluster identified in xenograft 3. We can say that the minor cluster from xenograft 3 probably did not progress further after serial passaging to the next mouse, whereas the major cluster persisted. This observation also agrees with the claim stated in the original paper (*8*) that after serial passaging only one cluster remained in xenograft 4 which is a descendant of the major cluster in xenograft 3.

Discussion

In this paper, we have proposed a new way of extracting useful features with biological significance from single cell data. The method is completely unsupervised and requires minimal preprocessing of the data. In many of the single cell algorithms, one of the key preprocessing step is modeling the drop out event in single cell data. Here we have eliminated that. The variational autoencoder projection is relatively robust to the drop out events. In the next steps, we would like to investigate more latent dimensions instead of three. Our intuition is that different group of latent dimensions will capture different aspects of the heterogeneity. For example, in this case the three dimensions captured the differentiation and lineage information, other other

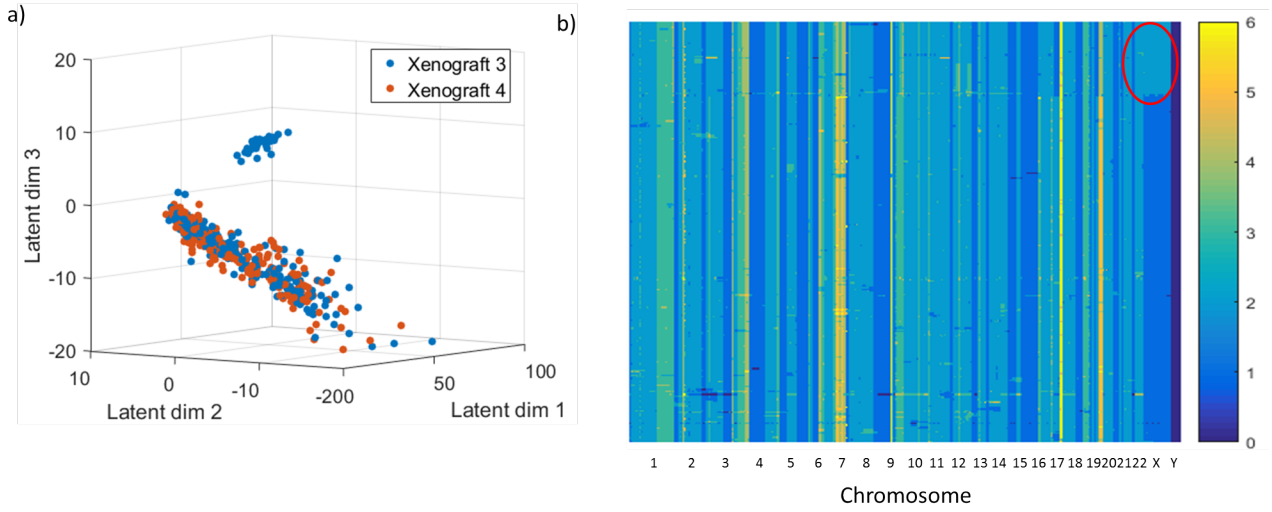

Figure 8: Autoencoder ouptut of two xenograft breast tumor samples' copy number profile. a) Identification of two subpopulations of cells in xenograft 3 and one subpopulation in xenograft 4. b) Copy number profile of cells in xenograft 3 ordered by phylogenetic analysis, which shows that there are indeed two groups of cells present in the data.

dimensions may reflect on other aspects such as tumor microenvironment and other possible epigenetic properties. The autoencoder latent representation could be used in pseudotime ordering of single cells as a viable alternate to t-SNE/PCA. We also want to investigate mutation profiles in future. The variational autoencoder does not only clusters the cells, it also shows a evolutionary structure among the malignant cells if there is any, such as the V structure for the oligodendroglioma. We want to investigate the biological significance of the learned structure in more detail. For example, we are interested in differentially expressed genes in the two arms of the bifurcation structure. Such analysis can lead us to new set of potential genes that are driving the differentiation process.

References and Notes

1. E. C. de Bruin, *et al.*, *Science* **346**, 251 (2014).

2. N. Andor, *et al.*, *Nature medicine* **22**, 105 (2016).

3. J.-W. Min, *et al.*, *PloS one* **10**, e0135817 (2015).

4. N. E. Navin, J. Hicks, *Molecular oncology* **4**, 267 (2010).

5. H. G. Russnes, N. Navin, J. Hicks, A.-L. Borresen-Dale, *The Journal of clinical investigation* **121**, 3810 (2011).

6. A. S. Venteicher, *et al.*, *Science* **355**, eaai8478 (2017).

7. I. Tirosh, *et al.*, *Nature* **539**, 309 (2016).

8. H. Zahn, *et al.*, *Nature methods* **14**, 167 (2017).

9. A. Giustacchini, *et al.*, *Nature medicine* **23**, 692 (2017).

10. C. Gawad, W. Koh, S. R. Quake, *Nature reviews. Genetics* **17**, 175 (2016).

11. C. Zong, S. Lu, A. R. Chapman, X. S. Xie, *Science* **338**, 1622 (2012).

12. L. v. d. Maaten, G. Hinton, *Journal of Machine Learning Research* **9**, 2579 (2008).

13. S. T. Roweis, L. K. Saul, *science* **290**, 2323 (2000).

14. C. Xu, Z. Su, *Bioinformatics* **31**, 1974 (2015).

15. J. Fan, *et al.*, *Nature methods* **13**, 241 (2016).

16. C. Lin, S. Jain, H. Kim, Z. Bar-Joseph, *Nucleic Acids Research* (2017).

17. A. Gupta, H. Wang, M. Ganapathiraju, *Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on* (IEEE, 2015), pp. 1328–1335.

18. D. P. Kingma, M. Welling, *arXiv preprint arXiv:1312.6114* (2013).

19. J. M. Joyce, *International Encyclopedia of Statistical Science* (Springer, 2011), pp. 720–722.

20. T. Tieleman, G. Hinton, *COURSERA: Neural networks for machine learning* **4**, 26 (2012).

21. I. T. Jolliffe, *Principal component analysis* (Springer, 1986), pp. 115–128.

22. X. Yu, G. Yu, J. Wang, *PloS one* **12**, e0171429 (2017).

23. K. Wang, *et al.*, *Genome research* **17**, 1665 (2007).

24. F. Chollet, *GitHub repository* (2015).

25. L. Kaufman, P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*, vol. 344 (John Wiley & Sons, 2009).

1 Appendix

1.1 Software

We have developed a python package for the Dhaka variational autoencoder using the Keras module (*24*). The package will be released as open source [2]. 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 (*25*). 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 drop out events, therefore we did not have to model the drop out events separately. The other parameters of the package

[2]Contact author for availability, ravip@microsoft.com

are the number of the latent dimensions, learning rate, batch size, number of epochs, and clip norm of the gradient [3].

1.2 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.

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