Data visualisation using t-SNE

Nitin Sai Beesabathuni*

*Department of Chemical Engineering, University of California, Davis, CA 95616
USA (e-mail: nsbee@ucdavis.edu).

Abstract: The report reviews and illustrates the application of t-distributed Stochastic Neighborhood Embedding(t-SNE) for data visualization purposes. The theory behind t-SNE is briefly discussed and a popular data set (MNIST) was used to compare the performance of t-SNE with PCA for data visualization purposes. Furthermore, spatial-temporal viral proteomic data was used to illustrate the proper application of t-SNE. The procedure of optimizing and choosing the hyperparameters (perplexity and the maximum number of iterations) for the t-SNE algorithm is discussed. Finally, the multi-dimensional spatial-temporal viral proteomic data was visualized in two-dimensions using the optimized parameters of t-SNE.

1. INTRODUCTION

Visualisation of high dimensional data into a lower dimension is a significant challenge in many fields. The objective of dimensionality reduction is to preserve the significant structure of high dimension data as much as possible when converted into lower dimensions. Multiple linear techniques such as PCA have been proposed to address this challenge. However, linear methods are not capable of capturing non-linear manifolds and do not preserve the local structure of the data. Various nonlinear dimensionality reduction techniques such as Sammon Mapping and Local Linear Embedding (LLE) that aim to preserve the local structure of data have been proposed (Verleysen and Lee, 2013). However, the performance of these techniques on real-world high dimensional data is poor. Especially, these are not capable of retaining both local and global structure of the data accurately.

Van Der Maaten and Hinton in 2008 introduced a new technique, known as t-distributed Stochastic Neighbourhood Embedding (t-SNE) (Van Der Maaten and Hinton in 2008). t-SNE is a non-linear dimensionality reduction technique primarily developed for data visualisation of high-dimension data into lower dimensions. t-SNE majorly focusses on capturing the local structure of the high-dimensional data, while also revealing global structure to some extent such as clusters. The performance of t-SNE for visualizing high dimensional data is superior compared to other previously discussed methods. A brief overview of t-SNE is discussed in the second section. A popular data set, called MNIST was used to compare the performance of t-SNE and PCA in section 3.1. t-SNE is currently used for a wide range of applications such as bioinformatics, music analysis, and cancer research. Spatial-temporal viral proteomics data(Jean Beltran, Mathias and Cristea, 2016) was used to illustrate the proper application of t-SNE. The report majorly focuses on two parameters of the t-SNE algorithm, namely perplexity and the maximum number of iterations. Multiple parameter values were tested for the proteomics data set and the set of parameters was selected to perform t-SNE for generating the final visualization of the data.

2. METHODS

2.1 Working of t-SNE

t-SNE is based on previously discussed Stochastic Neighbourhood Embedding (SNE) framework (Hinton and Roweis, 2003). t-SNE starts by representing pairwise similarities between data points in the higher dimension as conditional probabilities($p_{j|i}$). Equation 1 represents the conditional probability $p_{j|i}$, that x_i would pick x_j as a neighbour given that neighbours were picked based on the probability density under a gaussian centered at x_i.

$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\|x_i - x_k\|^2 / 2\sigma_i^2\right)}, \tag{1}$$

σ_i represents the variance of the gaussian that is centered at datapoint x_i. The variance for each data point is chosen based on a parameter known as perplexity. The perplexity is defined as

$$Perp(P_i) = 2^{H(P_i)} \tag{2}$$

Where $H(P_i)$ is the Shannon entropy given by

$$H(P_i) = -\sum_j p_{j|i} \log_2 p_{j|i} \tag{3}$$

The t-SNE algorithm performs a binary search for the value of σ_i that produces a P_i with a fixed perplexity based on users' specifications. Loosely speaking, perplexity can be interpreted as a measure of the effective number of neighbours for each data point. The value of perplexity varies based on the kind of data, but the typical values range between 5 and 50. Once the pairwise probability densities($p_{j|i}$) are calculated for each data point, the conditional probabilities are symmetrized by defining joint probabilities(p_{ij}), given by equation 4.

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n} \tag{4}$$

The symmetrized joint probabilities allow for faster computation. For the lower-dimensional counterparts y_i and y_j of the higher-dimensional datapoints x_i and x_j, pairwise joint probabilities are calculated based on student t-distribution with a single degree of freedom, given by equation 5.

$$q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|^2\right)^{-1}} \qquad (5)$$

Employing student t-distribution was the major novelty of the 2008 t-SNE method compared to the SNE method proposed in 2002. Using student t-distribution for computing joint probabilities in lower dimensions reduces the crowding problem and is the major advantage of t-SNE over other methods. The crowding problem is discussed in section 2.2.

Kullback- Leibler divergence is used as the cost function(C) to measure the accuracy of modelling higher dimension points (x_i and x_j) into lower dimensions (y_i and y_j) and to minimize the difference between p_{ij} *and* q_{ij}. The cost function is given by equation 6. The cost function focusses on retaining the local structure of the data over the global structure.

$$C = KL(P\|Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \qquad (6)$$

Minimization of the cost function defined in equation 6 is performed using the gradient descent method. The gradient of the cost function is given by equation 7.

$$\frac{\delta C}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)\left(1 + \|y_i - y_j\|^2\right)^{-1} \qquad (7)$$

The gradient of the cost function governs the assignment of points in the lower dimension. For example, if two points are close together in a higher dimension, p_{ij} *would* be high and if they are placed far apart in the lower dimension, it would result in a low q_{ij} value. This would result in a net attractive force ($p_{ij} > q_{ij}$) to move the points closer in the lower dimension.

2.2 Crowding Problem

The crowding problem can be stated as the area available of the accommodating moderately distant datapoints in lower dimensions (preferably two-dimensions) will not be large enough compared to the area available for accommodating nearby data points. Therefore, if we want to model nearby data points from a higher dimension to a lower dimension accurately, the moderately distant data points in a higher dimension would be placed farther apart in lower dimensions. This yields unwanted attractive forces between clusters and leads to crowding of the clusters with no separation. Crowding problem is a major challenge for methods such as SNE and Sammon mapping.

2.3 t-SNE reduces the crowding problem.

t-SNE alleviates the crowding problem but using a student-t-distribution to map the probability densities between data points in the lower dimension. The student-t-distribution has a heavy-tailed distribution compared to a gaussian which would allow the moderately distant points to be placed far apart in the lower dimension without collapsing onto each other.

2.4 Data analysis

All data were analysed using R studio. Rtsne package was imported for performing t-SNE. ggplot2 package was imported for generating all the visualizations. Tidyr and dplyr packages were imported for cleaning and manipulation of the data.

3. RESULTS

3.1 Comparison of t-SNE to PCA using the MNIST data set.

To illustrate the performance of t-SNE over other methods such as PCA, I choose a popular data set- MNIST (Lecun et al, 1998). MNIST data set contains grayscale images of handwritten digits (0-9). Each data point represents an image and each image has 28X28 = 784 pixels in it. Each pixel can be considered as a dimension with either 1 or 0 as value. Using princomp function in R, PCA was performed on the first 1000 images. The first two principal components were used to display the 1000 data points on a two-dimensional map, shown in figure 1A. Rtsne package was used to perform t-SNE data visualisation on the first 1000 data points. Figure 1B shows the data visualisation performed by t-SNE. The true value information (digit number) for each data point is known. The true value information was not used during dimensionality reduction. This information was only used to color-code each data point and was overlaid after performing dimensionality reduction.

Even though PCA captures the local structure of the data to some extent (digit one's cluster on the leftmost side and digit zeros cluster on the rightmost side), it does not separate individual clusters. Whereas data visualisation using t-SNE effectively separates each cluster and captures the local structure of the data. This example illustrates the better performance of t-SNE for data visualisation of high-dimension data.

3.2 Selection of t-SNE parameters for spatial-temporal proteomic data.

MNIST is a popular data set and has been well-studied by many groups, therefore, performing t-SNE was not a significant challenge. However, often we would have minimal information about the data, thus making the proper application of t-SNE challenging. To illustrate the workflow of t-SNE, I choose a spatial-temporal viral proteomic data set. The main component of performing t-SNE is parameter optimisation. Here I focussed on the two main t-SNE parameters- perplexity and the maximum number of iterations. The spatial temporal viral proteomic data set contains information about cellular and viral proteins during virus infection conditions. Human cells were infected with HCMV virus and the cells were divided into six fractions using density gradient centrifugation. Each fraction was then sent to mass spectrometry to detect the proteins in each fraction. The data set represents the relative protein concentration of almost 3000 proteins in six fractions. Cell samples were taken at multiple time points (24, 36, 48, 72, 96, and 120 hours) post-infection. The goal of the section is to illustrate the selection of parameters of the t-SNE method to visualize the localization of all the proteins from the six fractions in two dimensions maps.

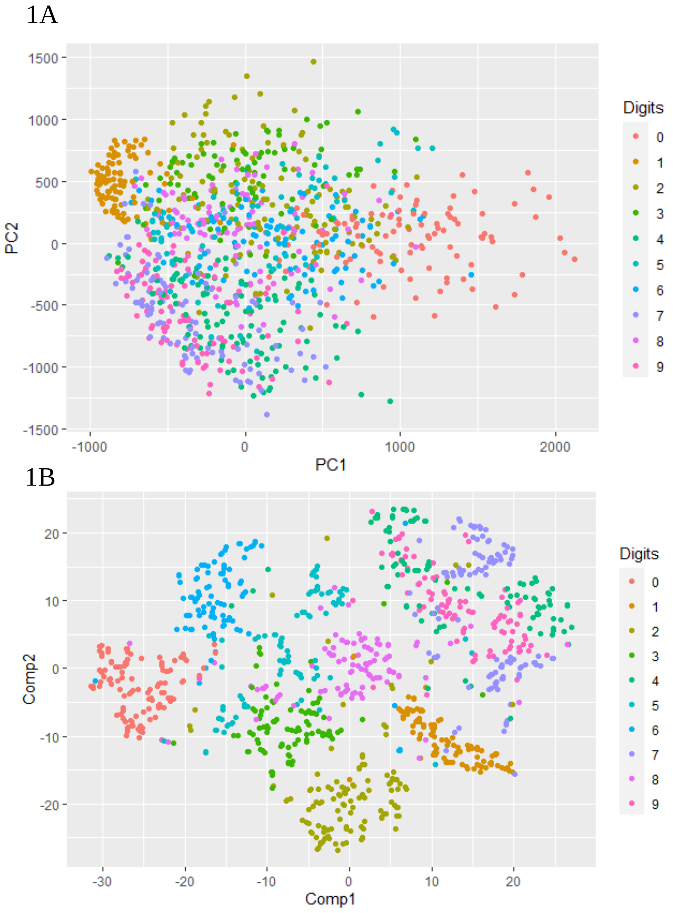

Figure 1: Comparison of PCA vs t-SNE data visualisation performance on the first 1000 data points of the MNIST data set. (A) Dimensionality reduction using PCA, the first two principal components were used for visualisation. (B) Data visualisation using t-SNE, perplexity= 20, and the maximum number of iterations = 500.

3.2.1 Selection of perplexity

24 hours post infection (hpi) data was chosen to test different perplexity values. Multiple perplexity values (5,25,50,75, and 100) were tested keeping the other parameters constant. Figure 2 shows the data visualisation generated using various perplexity values.

Each data point represents a protein and each protein’s relative localization in the 6 fractions at 24-hour post-infection was used to generate the visualisation using t-SNE. The colour coded data points indicate the proteins whose localization to organelles is known, such proteins were called organelle markers. The organelle marker information was overlaid after t-SNE analysis. The white data points (NA) represent the proteins whose localization is unknown initially. Figure 2A represents the visualisation generated using a perplexity value of 5 and no clear clusters can be observed. In figure 2B, a perplexity value of 25 was used, even though some clusters can be detected there is no clear separation between them.

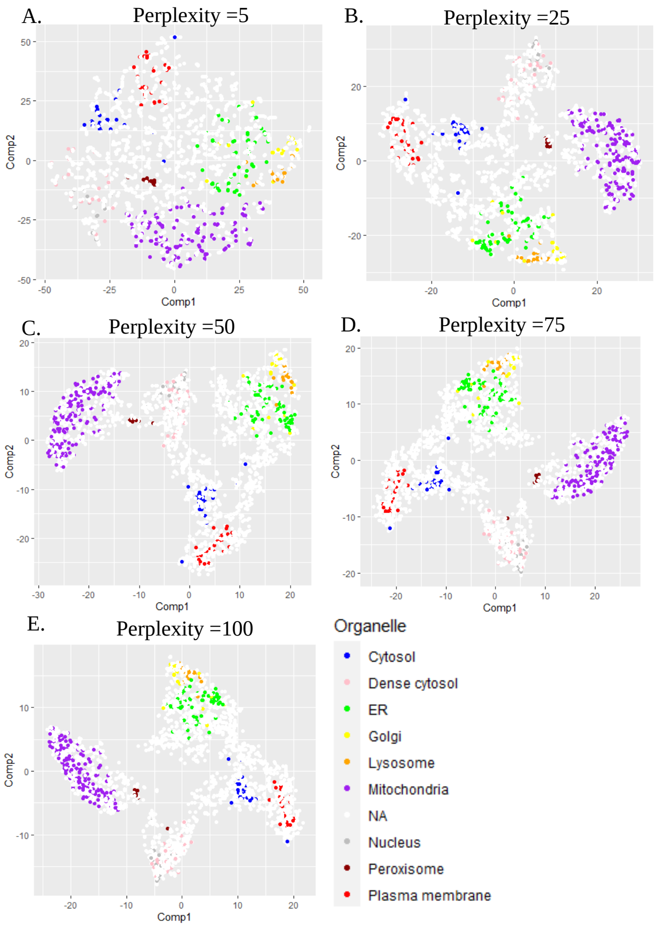

Figure 2: Data visualisation of 24hpi fractionation data for varying perplexity values. Other parameters were kept constant and were assigned default values with maximum number of iterations=500. (A) perplexity = 5, (B) perplexity = 25 (C), perplexity = 50, (D) perplexity = 75, (E) perplexity = 100.

Figure 2C illustrates the visualization generated using a perplexity value of 50. Perplexity value of 50 yields a clear separation between clusters as well as preserves the local structure well. For example, all the mitochondrial proteins localize to one cluster (purple). Increasing the value of the perplexity to 75 and 100 does not produce different results compared to 50. Thus, 50 was chosen as the perplexity value. An important note to make here is the absolute positions of the cluster is unimportant as it depends on the initial random seeding of all points in the lower dimensional map. Instead, the size of each cluster and the relative position of one cluster to another is of importance.

3.2.2 Selection of the maximum number of iterations

With perplexity as 50, the next parameter that was varied is the maximum number of iterations. Using the 24hpi data again, the value of the cost function (Equation 6) for each iteration was monitored. Each data point in Figure 3 represents the iteration cost for every 50 iterations. It can be observed that there is no significant change in the iteration cost after 350 iterations. Therefore, any number of iterations higher than 350 should generate similar visualizations. 500 was chosen as the maximum number of iteration value.

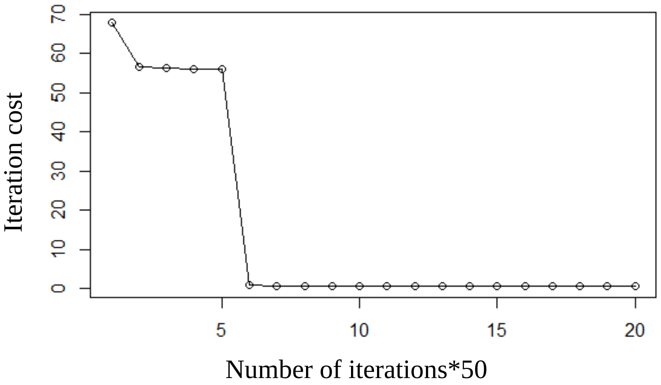

Figure 3: Iteration cost vs the number of iterations. Perplexity value= 50 and other parameter values were left as default.

3.2.3 2D visualisation of spatial-temporal viral proteomic data.

Using the optimized parameter values (perplexity = 50 and maximum number of iterations = 500), all the fractionation data at multiple time points (24, 48, 72, 98, 120) post-infection were visualized. Figure 4 shows the visualizations generated using t-SNE for all time points. The authors of the spatial-temporal viral proteomic work used the organelle marker information to train a neural network to predict the localization of the unknown proteins. Therefore, we now have the nformation on localization of most proteins. Some proteins had unspecified localizations (white dots in figure 4) as their prediction scores were low. The viral proteins are represented by a circle and the cellular proteins are represented by triangles. The color indicates the localization of the protein at different organelles. The two-dimensional visualizations allow us to track the movement of the viral and cellular proteins over multiple organelles overtime during virus infection.

4. CONCLUSIONS

In this report, I demonstrate the usefulness of t-SNE as a tool in generating data visualisation of high-dimensional data. In the first section of this report, I reviewed the working of t-SNE along with the major differences and advantages of t-SNE over other methods. The later section of this report focuses on data visualisation of high dimensional data using t-SNE. Using a popular data set (MNIST), I have demonstrated that t-SNE performs significantly better compared to PCA. Especially in terms of preserving the local structure of the data. Additionally, using a more complicated data set (spatial-temporal proteomics), I illustrated the importance and workflow of optimizing t-SNE parameters for generating reliable visualization. Here, I focused on two key t-SNE parameters, namely perplexity and the maximum number of iterations. The visualization can be further improved by optimizing other parameters of the t-SNE algorithm such as the learning rate.

Even though t-SNE is a great tool for data visualisation, caution should be exercised while selecting the t-SNE parameters. Multiple parameters must be tested especially for data with minimal information. Once the parameters are optimized, multiple runs need to be performed before selecting the final visualisation. Assessing visualisations is one of the key challenges of using t-SNE. Most visualisation produced by t-SNE is evaluated manually by inspecting at them. However, this could yield a lot of variabilities based on the data and the inspector. Furthermore, as the t-SNE cost function is non-convex, the optima of the objective function greatly depends on the data, optimised parameters, and initial random seeding of the map points in the lower dimension. As a result, assessing the visualisation manually could be further challenging. However, new automated tools are being developed to assess the visualisation for various applications (Belkina *et al.*, 2019 Chatzimparmpas, Martins and Kerren, 2020). In conclusion, careful utilisation of t-SNE for visualising high dimensional data in lower dimensions is an invaluable tool.

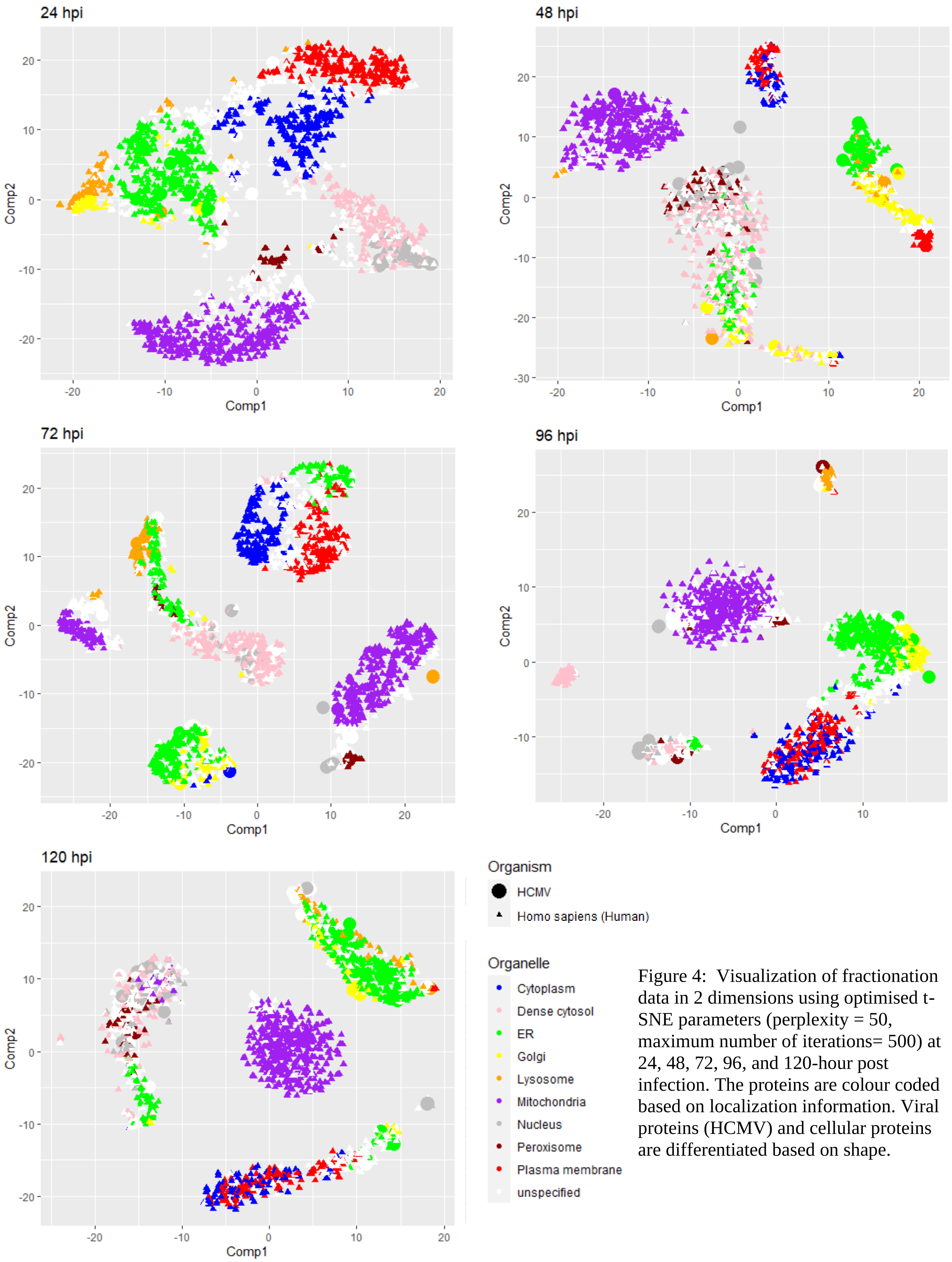

Figure 4: Visualization of fractionation data in 2 dimensions using optimised t-SNE parameters (perplexity = 50, maximum number of iterations= 500) at 24, 48, 72, 96, and 120-hour post infection. The proteins are colour coded based on localization information. Viral proteins (HCMV) and cellular proteins are differentiated based on shape.

REFERENCES

Belkina, A. C. *et al.* (2019) ‘Automated optimized parameters for T-distributed stochastic neighbor embedding improve visualization and analysis of large datasets’, *Nature Communications*. Springer US, 10(1), pp. 1–12. doi: 10.1038/s41467-019-13055-y.

Chatzimparmpas, A., Martins, R. M. and Kerren, A. (2020) ‘t-viSNE: Interactive Assessment and Interpretation of t-SNE Projections’, *IEEE Transactions on Visualization and Computer Graphics*, XX(X), pp. 1–1. doi: 10.1109/tvcg.2020.2986996.

Hinton, G. and Roweis, S. (2003) ‘Stochastic neighbor embedding’, *Advances in Neural Information Processing Systems*.

Jean Beltran, P. M., Mathias, R. A. and Cristea, I. M. (2016) ‘A Portrait of the Human Organelle Proteome In Space and Time during Cytomegalovirus Infection’, *Cell Systems*. Elsevier Inc., 3(4), pp. 361-373.e6. doi: 10.1016/j.cels.2016.08.012.

Van Der Maaten, L. and Hinton, G. (2008) ‘Visualizing data using t-SNE’, *Journal of Machine Learning Research*, 9, pp. 2579–2625.

Verleysen, M. and Lee, J. A. (2013) ‘Nonlinear dimensionality reduction for visualization’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8226 LNCS(PART 1), pp. 617–622. doi: 10.1007/978-3-642-42054-2_77.