SurvClusVis: Exploring Survival Datasets Using Dimensionality Reduction

Lovedeep Gondara

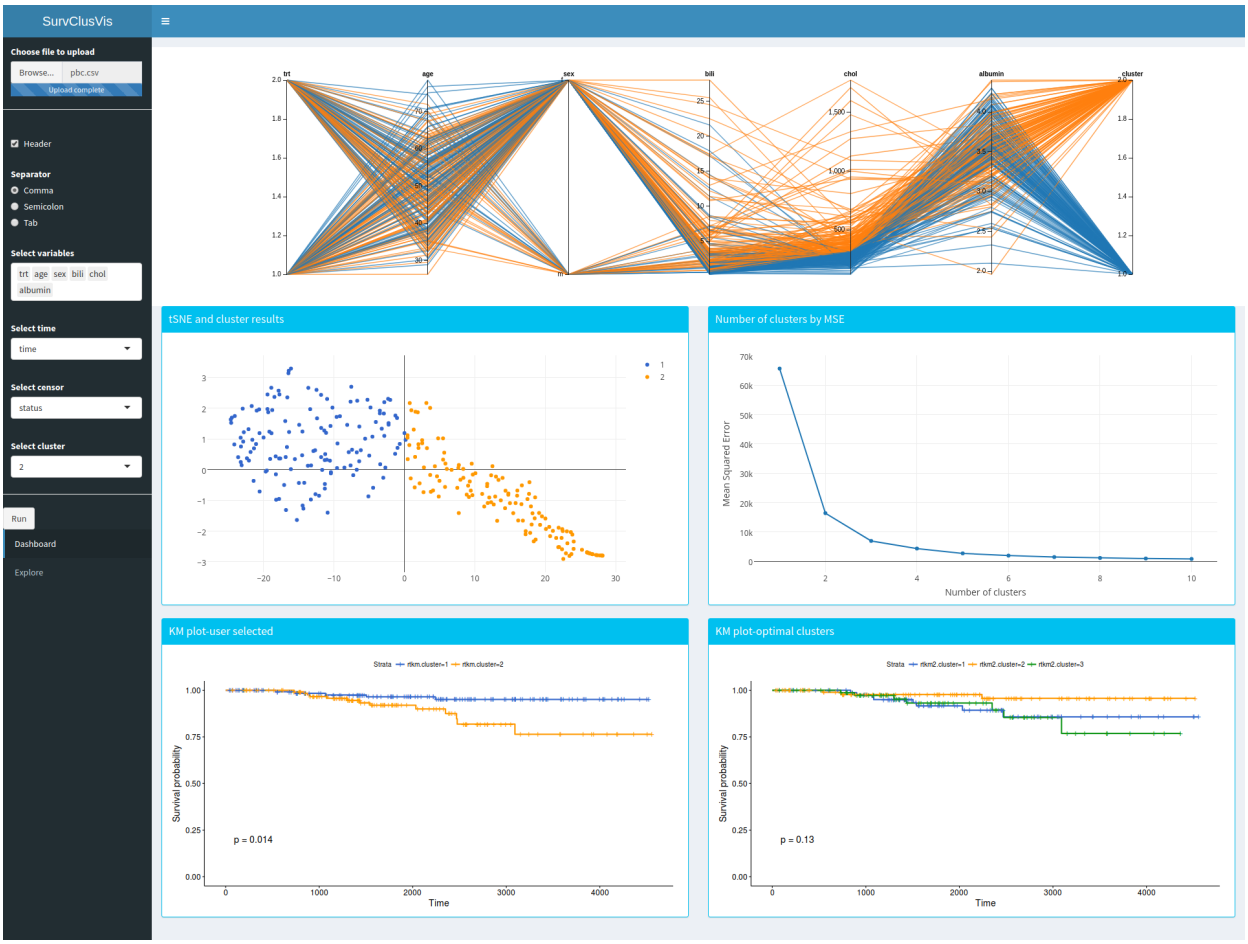

Fig. 1. Teaser of our final application showing its main components

Abstract— Here we present SurvClusVis, an application based on R shiny used to visualize survival data based on tSNE. We first find an appropriate low dimensional representation of the original data using tSNE, which is then clustered using K-means, cluster membership per observation from original data is used to construct survival probabilities by cluster and cluster memberships can be further explored used parallel coordinate plots and scatter plot matrices. This is meant to enhance a clinical researcher's toolbox and help them better explore survival datasets.

Index Terms—Dimensionality reduction, survival data, cluster analysis, tSNE

1 Introduction

In clinical research, for small to medium size studies, data is often collected by physicians for ongoing investigation as separate small datasets. The collected data is then sent to the analysts for providing meaningful interpretations and most of the time, goal is to find heterogeneous group of patients that have different survival patterns. Survival patterns can depend on any observed covariate or latent unobserved factors or a combination of both. Sometimes, depending on the data quality, sample size or collection method, there are no obvious patterns or signals defining different patient groupings. Tight research budgets and limited tool availability limit the researchers potential for effective data visualization and exploration, which can be used to filter interesting studies from uninteresting ones. This is especially the case with physician researchers, where greater monetary and human resources are required and study outcomes have a profound effect. Any indications of different survival patterns in different subgroups will motivate the researchers for further exploration and analysis. But as this

- *Final project for CSPC 547*
  Author email: lgondara@sfu.ca
  GitHub:`https://github.com/lgondara/SurvClusVis`
  Live demo: `https://lgondara.shinyapps.io/survclusvis/`
  Video: `https://youtu.be/7fkFDh8gZlw`

is not straightforward using available tools, researchers end up using cross tabulations and basic bivariate scatter plots to get a sense of the information contained in the dataset before they approach an analyst. This very basic and unstructured approach can sometimes result in not so clear hypothesis and unstructured questions.

It would be very useful if before fully committing to a study, researchers can explore their data efficiently, which currently is not possible because of higher costs for corporate visualization software and programming skills required for open source packages. An effective visualization can not only help researchers screen datasets efficiently, but can help them to frame new questions that they did not think of before, based on new insights gained from exploration.

To answer some of these questions and to take a step towards creating something simple on the frontend, but doing complex analytics on the backend, in this paper, we develop a simple visual analytics tool, that helps researchers to discover hidden structures in their data. We start with using tSNE [29] on original data to reduce its dimensionality and then cluster the tSNE output using K-means [16]. Cluster membership per observation is then used for further analysis, using survival function estimation and basic visual exploration using parallel coordinate plots and scatterplots. We provide user interaction with plots in form of zoom, brushing and linking.

2 Related work

Chia et al. [8] presented a detailed review of existing and evolving visualization methods in medical analytics, ranging from simple Kaplan-Meier plots of survival probability to heatmaps and circos plots for genotype and phenotype visualization. Thorvaldsdóttir et al. [27] presented IGV (Integrative Genomics Viewer) to visualize and explore high dimensional genomics data with a focus on interactive exploration using standard desktop computers. Sharan et al. [20] presented Click and Expander that clusters data based on kernel similarities and provides visualization capabilities for raw and clustered data. Sturn et al. [23] presented Genesis, a tool for microarray data analysis using clustering which provides cluster visualization for various clustering algorithms in form of heatmaps and treemaps. Dhillon et al. [9] presented IslandViewer 3 for interactive genome analysis and gene search based on D3 and circos plots.

Most clinical data is inherently high dimensional and good dimensionality reduction techniques are vital for faithful visualizations. T-Distributed Stochastic Neighbor Embedding (t-SNE) [29] is such a dimensionality reduction technique suitable for high dimensional data visualization, it works by constructing a probability distribution on pairs of high dimensional objects in a way that similar objects have high probability of getting selected. Several versions of t-SNE have been proposed since, including Barnes-Hut-SNE [28], which is a faster version.

Due to t-SNEs attractive property of embedding high dimensional data efficiently to a low dimensional sub-space, it has been used extensively in clinical analytics and visualization. t-SNE has been used to identify prognostic tumor sub-populations using mass spectrometry imaging data [1] and to visualize SNPs [18]. t-SNE was used to visualize clinical text notes for sentiments by Ghassemi et al. [10]. Xu et al. [31] used t-SNE for visualizing phenotypical similarities between different human diseases. t-SNE was used by Becher et al. [4] for cell subset identification.

3 Preliminaries

In this section we provide some essential background for the methods central to this project.

3.1 tSNE

tSNE (t Distributed Stochastic Neighbourhood Embedding) [29] is a recent popular method of exploring and visualizing high dimensional data. Before we explain tSNE, we start by introducing SNE, of which tSNE is a direct extension. SNE (Stochastic Neighbourhood Embedding) [11] converts the high dimensional Euclidean distances between data points into conditional probabilities representing similarities. Given two data points x_i and x_j, conditional probability $p_{i|j}$ represents the probability that x_j will pick x_i as the neighbour if neighbours are picked in proportion to their probability density under a Gaussian centered at x_j, if σ_i is the variance of Gaussian centered at x_j, we can write it as

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

to represent the same in low dimensional sub-space, it is possible to construct low dimensional counterparts y_i and y_j, such as their conditional probability is given by

$$pq_{i|j} = \frac{exp(-||y_j - y_i||^2)}{\sum_{k\neq j} exp(-||x_j - x_k||^2} \tag{2}$$

where the value of σ is set to $1/\sqrt{2}$. If points y_i and y_j provide a faithful representation of x_i and x_j in low dimensional space, the values of p_{ij} and q_{ij} would be same. Objective function of SNE is to find the representation that minimizes the difference between p_{ij} and q_{ij}, which is accomplished by minimizing Kullback-Liebler divergence [15] between both

$$C = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{i|j} \log \frac{p_{i|j}}{q_{i|j}} \tag{3}$$

where we minimize this divergence over all samples and this is usually accomplished using gradient descent. Although SNE works reasonably good, it has some shortcomings such as difficult to optimize cost function and "crowding problem". tSNE remedies these problems by introducing a different cost function and by using Student's t distribution [22] instead of Gaussian. In particular, it uses the symmetric version of SNE's loss function.

tSNE has a tunable parameter, called perplexity, which aims to balance the local and global aspects of the dataset, in a sense, it refers to the number of neighbours a data point has. It is a user-tunable parameter that can have profound effects on output quality. For this project, we leave it at the default value.

3.2 Kaplan-Meier survival analysis

The Kaplan-Meier [12] estimation of survival function is the most popular method to analyze time to event data. Where we define time to event as the time elapsed from start of observation period to the time point when the observation period ends or when a subject has an event of interest. In survival analysis, subjects that did not had the event are marked as censored observations. There are several key assumptions to censoring in survival analysis, such as any censored subjects have the same survival prospects as the patients who are being followed up.

Kaplan-Meier method is also known as the product limit estimator and is used to estimate survival function in a non-parametric way from survival data. Given number of subjects n_i at risk at a given time t_i, and number of subjects that had events d_i, we can estimate survival function as

$$S(t_i) = \prod_{t_i \leq t} 1 - \frac{d_i}{n_i} \tag{4}$$

$S(t_i)$ is a step function that drops every time an event occurs.

3.3 Survival datasets

As our project is concerned with survival analysis, we exclusively use survival datasets, also known as time to event data [2]. Time to event is the outcome of interest in survival analysis and refers to how long it takes for a subject from the start of observation period to experience our event of interest. Event of interest can be adverse, such as death or relapse; or positive, such as hospital discharge or complete recovery.

One of the distinguishing feature of survival datasets from other types is that the event would not have occurred for everyone in the study, that is, some subjects would be alive, not relapsed, not discharged or not recovered by the end of of our followup period. Such cases are known as censored observations. Same as other datasets, there can

be additional attributes collected on subjects that might affect time to event information, such as age, sex, treatment etc.

Table 1 shows a data snippet for a time to event dataset. Attribute "Time" holds the information for how long a subject was under observation and attribute "Outcome" is the event indicator, "1" signifies that a subject has had a cardiovascular event at the recorded "Time" from observation start and "0" means that subjects did not had any cardiovascular event. Other attributes are recorded for subjects including their age, distance to nearest care facility and sex.

Table 1. Data snippet for a cardiovascular study

Patient Id	Age	Distance	Sex	Time	Outcome
1	50	100	Male	80	1
2	60	110	Female	65	0
3	80	140	Male	70	0
4	78	120	Male	100	1

4 Data and Task abstractions

Data and task abstractions used for this project are specific to survival analysis, it is however easy to modify this project to be used with *non-survival* data for similar exploration.

4.1 Data abstraction

Our project is dataset agnostic, and is capable of functioning with all datasets in domain of survival analysis. In this paper however, we use dataset *PBC* [25] for our project evaluation, PBC refers to primary biliary cirrhosis of the liver and the data was collected for a Mayo Clinic trial from 1974 to 1984. A total of 424 PBC patients, referred to Mayo Clinic during that ten-year interval, met eligibility criteria for the randomized placebo controlled trial of the drug D-penicillamine. The first 312 cases in the data set participated in the randomized trial and contain largely complete data. The additional 112 cases did not participate in the clinical trial, but consented to have basic measurements recorded and to be followed for survival. Six of those cases were lost to follow-up shortly after diagnosis, so the data used in this project are on an additional 106 cases as well as the 312 randomized participants. The dataset itself is publicly available through various online and in software portals. For this project, we use the version available in R package survival [26]. The dataset contains time to event information where time is defined as number of days between registration and the earlier of death, transplantion, or study analysis in July, 1986 and event is the status at endpoint, 0/1/2 for censored, transplant, dead. Other available attributes include age, albumin, alkaline phosphate, ascites, aspartate aminotransferase, serum bilirunbin, serum cholesterol, urine copper, edema, presence of hepatomegaly or enlarged liver, platelet count, standardised blood clotting time, sex, spiders, stage, treatment and triglycerides.

As the standard survival analysis can only deal with one cause of failure, here we only concern ourselves with the event "transplant". That is, death is considered censoring and our analysis will investigate the time to transplant based on different attributes.

4.2 Task abstraction

4.2.1 Domain specific tasks

1. Upload and use any survival dataset: As the project targets physician researchers, we only know the basic type of datasets that are used, that is the time to event datasets, but we can not limit our application to a specific time to event dataset. An ideal application will has to have an upload option for researchers to upload their own datafiles that are read properly in various formats.

2. Select variables of interest: Raw datasets can contain various variables that are redundant, incomplete or not of interest for end users. So, it is very important to give users the ability to select the variables of interest and ignore other variables. As we are dealing with survival analysis, we would like to know which variables are observation time and which variable holds the information for censoring, so we can use them for survival modelling later on.

3. Deal with multiple censoring values: Ideally censoring variable would be binary, that is it takes values of 0 for censored observations and 1 for events, but in real life scenarios, there can be multiple values coded as different type of events. We would want the application to be able to only use 1 as event and consider all other values as censoring.

4. Reproducible tSNE clustering: As an inherent property of tSNE algorithm, rerunning it with same parameters would produce different results, with results being dramatically different sometimes. This would confuse a naive user, we would like to set a seed that makes tSNE produce reproducible results at consecutive runs in the same session.

5. User specified clustering: End users would want to explore clustering outcomes with different number of clusters, this can be automated using some algorithmic selection, but users might have a preset hypothesis, according to which they might believe in existence of different number of clusters compared to automated results. So we would like to provide users with a selection method for selecting number of clusters to be used by K-means [16] to assign cluster membership to tSNE outcomes.

6. Use tSNE results for survival analysis: To facilitate the end goal of this project, that is to discover underlying patient groupings that have different survival outcomes, we would want to use the clustering outcomes to somehow investigate the subjects survival probabilities based on cluster membership. We do so using Kaplan Meier method of estimating survival function and plotting the results.

7. Explore data based on KM results: If a user discovers differential survival in different subjects by their cluster membership or there are any other interesting patterns, we would want the user to be able to further investigate the causes for these patterns. To do so, we would want the user to be able to explore the different attributes and subjects by cluster membership.

4.2.2 Visualization tasks

As most datasets are inherently high dimensional, it makes it difficult for users to view the underlying structure using conventional data exploration methods such as bivariate scatter plots and contingency tables. Even more complex, some variables might affect the outcome together, which is near impossible to visualize using conventional visualization platforms. Here we define some visualization specific tasks essential for our project.

1. Data overview: As the datasets are inherently high dimensional, it is not plausible to visualize them using standard scatter, line or barplots. We would want to have a visualization that effectively visualize a complete dataset in given 2D space.

2. Present tSNE results: Using tSNE, we can decrease the dataset dimensionality to two dimensions which we would want to display along with clustering outcome of K-means run on tSNE output.

3. Present optimal number of clusters: With user selected number of clusters, we would also want to show the optimal number of clusters resulting in highest mean squared error decrease to guide the user to make more informed choices.

4. Present updated data overview for selected variables and cluster outcome: After clustering, we would want to show the subjects by cluster membership by attributes, to help discover any apparent patterns in tSNE outcomes.

5. Present survival information associated with cluster membership: For user selected and the optimal number of clusters chosen by the algorithm, we would want to display the subjects survival in a meaningful way, grouped by the individual cluster membership.

5 Solution

5.1 Design overview

We start with a basic layout divided into different panels that hold different visualizations. Main screen is divided into five panels, a top panel that runs the full length holds the dataset visualization using parallel coordinates plot. The space below parallel coordinate plot holds four panels of same size.

Fig. 2. Color scheme used for this project

Figure 3 shows this basic layout. Left panel is reserved for user input such as dataset upload, variable selection, selecting number of clusters and further exploration tabs while the center panel is used to display visualizations. To reduce number of fields displayed on left panel. Variable selection is only displayed once a dataset is loaded and the run button is forced inactive unless variable selection occurs. For the color choice, we decided to use a categorical color palette, namely d3 category 10 shown in Figure 2.

As dataset size for medical informatics is limited, we also limit the choice for choosing maximum number of clusters to be five.

5.2 Overall data view

Figure 4 shows the overall data view using parallel coordinate plot, this shows the distribution of individual data points (subjects) across attributes and makes it easier to see individual attribute distributions across dimensions. Users can use brushing to select a set of observations and observe them across different attributes.

5.3 tSNE output and clustering

After inspecting the attribute distribution using a parallel coordinate plot, users can select which attributes they want to include for tSNE and subsequent K-means clustering using a variable selection from left hand panel using a dropdown with multiple selection enabled. After selecting the attributes, user can press the run button, which starts tSNE and runs K-means clustering over tSNE output. Results are then displayed by updating the parallel coordinate plot to only include the chosen variables and color coding the cluster membership. Users can use brushing to investigate cluster membership by individual observations. Figure 5 shows the output of such an updated parallel coordinate plot that has the cluster as an additional variable at the far right with color coding differentiating different clusters.

Figure 6 shows the scatter plot generated by tSNE output, which is color coded by cluster membership. This plot serves an important purpose of displaying any important underlying hidden patterns present in the dataset. As the dataset being plotted is the output of tSNE, that is the original datasets representation in two dimensions, axis labels are removed to minimize any confusion and wrong interpretations.

5.4 Optimal clusters

There is a trade-off in giving the user autonomy to select the number of clusters, users can select any arbitrary number, whether it does or does not make sense for the dataset being used. To help guide users for better exploration, we have included a simple line plot that shows the mean squared error (MSE) by number of chosen clusters. MSE is calculated as

$$MSE = \sum_{i=1}^{k} \sum_{x \in c_i} ||x_i - \mu_i||^2 \quad (5)$$

where k is the number of clusters and μ_i is the centroid or mean of each cluster. Figure 7 shows such a plot, a drop in MSE by increasing number of clusters is apparent, but the ideal trade-off between the number of clusters and MSE is at the *elbow* of the curve.

5.5 KM plots

As the main goal of this application is to show the survival of subjects by their cluster membership, we use the most often used visualization to show the survival distribution over time, the Kaplan-Meier plot (KM plot). KM plot plots the survival distribution estimates using non-parametric KM method.

Figures 8 and 9 shows the outcome of using KM plots using user selected and optimal number of clusters using MSE. Bigger the separation between different curves, apparent it is that there is a difference in survival distribution of subjects in different clusters. We decided to show both plots as to show the user what the output will look like if they were to go with automated cluster selection. If they like the automated plot, they can change the number of clusters to the one chosen by the algorithm and rerun the clustering process.

5.6 Further exploration

If the users have found some interesting patterns in clustering and they further wish to understand the attribute distributions that lead to this output, they can click on explore button from left panel. This takes them to a new window, where they are presented with a scatter plot matrix of all variables used for clustering, color coded by subjects cluster membership. Same colors are used to indicate clusters to minimize confusion when moving from one view to another.

Figures 10 and 11 show the results. Figure 10 shows the scatter plot matrix presented to the user when entering this view and holds the plots for all variables included in clustering. Users can brush to choose the points in one plot and the views across the plots are linked to highlight the choices. A radio button on the top has the option to view scatter plot matrix for the variables that are statistically significantly different across clusters, this significance is determined using KruskalWallis test [14] at an α of ≤ 0.05.

5.7 Implementation

The application is mainly built using R [19] with shiny and shiny dashboards [6, 7]. Parallel coordinate plots are created using R package parcoords [5], MSE and tSNE plots are created using plotly for R [21], survival plots are created using R package survminer [13]. Scatter plot matrices are created using R package pairsD3 [24]. R packages dplyr [30] is used for data manipulation and package survival [26] is used for survival modelling. Package shinyjs [3] is used for inserting javascript for hiding and updating plots on front page.

Our application follows a standard shiny package format with three main files; global.R, which loads all the required libraries to the running R session; ui.R, creates the user interface with side panel and main panel and all the input options; server.R, does all the computation, modelling and creates visualizations that are passed on to ui.R for display.

6 Use case

A physician researcher Amy has a dataset relating to primary biliary cirrhosis, that she has collected over the years. Satisfied with the sample size, she is ready to formally analyze the data. She starts by opening the application and is presented with an empty page with a button to upload the data. Figure 12 shows this scenario.

After selecting the file to upload in a csv format, Amy is presented with an updated view showing a parallel coordinate plot of all variables in the dataset. Figure 13 shows the updated page, Amy now interacts with the parallel coordinate plot by brushing on the attributes to select a subset of observations. She does not sees anything of an immediate concern or any apparent outliers, so she proceeds with the next step. Amy is interested in investigating the difference between patients that get a liver transplant compared to the ones that did not, she thinks there might be some underlying factors influencing the selection.

So she starts with selecting the variables of her interest from the dropdown provided on the left panel. She selects the treatment received by the subjects, their age, sex, bilirubin score, cholesterol levels and the albumin levels as she thinks these to be linked with liver function. After selection she decides to choose two as the number of clusters she wants to investigate. This process is shown in Figure 14. After pressing

Fig. 3. Start screen for the application, left panel is for file upload, variable selection, user input for number of clusters and further exploration. Center top is reserved for a single parallel coordinates plot for overview of dataset. Bottom center is divided into four panels to host four different visualizations described in following sections.

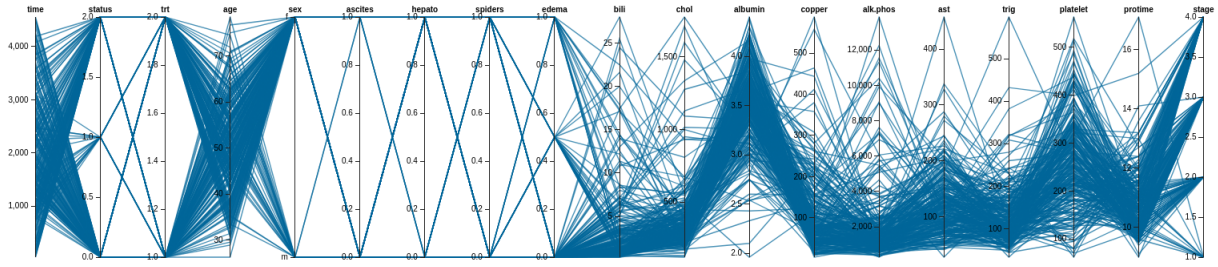

Fig. 4. Parallel coordinate plot presented to users on top center panel instantly after dataset upload to assist users by providing them with attribute distribution.

run button, tSNE runs on the selected variables and Amy is presented with an updated interface.

Figure 15 shows the upper part of this updated interface, where the parallel coordinate plot of all attributes is updated to only include variables selected by Amy and the cluster membership per subject is color coded. Amy then uses brushing to highlight high cholesterol values and finds out that higher values of cholesterol contribute mostly to cluster 2. Any other variables do not show any interesting patterns. Amy then looks at the tSNE plot and sees a nice separation of both clusters, she also looks at the MSE plot and finds the curve elbow at 3, her choice of 2 has a bit higher MSE than 3.

She then scrolls down to look at the survival plots generated using the method of Kaplan-Meier, Amy is presented with two survival plots, one using the grouping of clusters specified by Amy and second using the number from the elbow of MSE plot. Looking at the survival plot using the clusters specified by her, she sees a good separation of survival distributions between clusters and also a low p-value from comparison of both distributions using a log-rank test, suggesting a significant difference between survival distributions. On the other hand looking at the plot generated using MSE inspired number of clusters, there is no clear separation. Amy is content with her selection of number of clusters and wants to explore the clusters further.

Amy clicks on explore tab on the left panel, it takes her to the second page where she is presented with a scatter plot matrix of all variables used for clustering. Figure 17 shows the scatter plot matrix, Amy then proceeds to interact with the scatter plot matrix using brushing, any points chosen in one plot is linked to the same point in all other plots that is automatically highlighted to show the position of that point in all attributes. She knows from previous visualization that subjects in cluster 2 have worse chances of getting a transplant. She brushes high cholesterol points in scatter plot matrix to check their distribution for other variables and finds that it looks like patients with higher cholesterol are also of older age. Seeing some interesting patterns, she decides to proceed further with the detailed analysis and contacts a statistician for detailed modelling and analysis.

7 Conclusion and Limitations

We have presented an application that can be used to effectively visualize survival datasets by first reducing their dimensionality using tSNE, running K-means clustering on tSNE output and assigning cluster membership to individual observations. Users can further investigate the cluster membership using scatter plot and parallel coordinate plots.

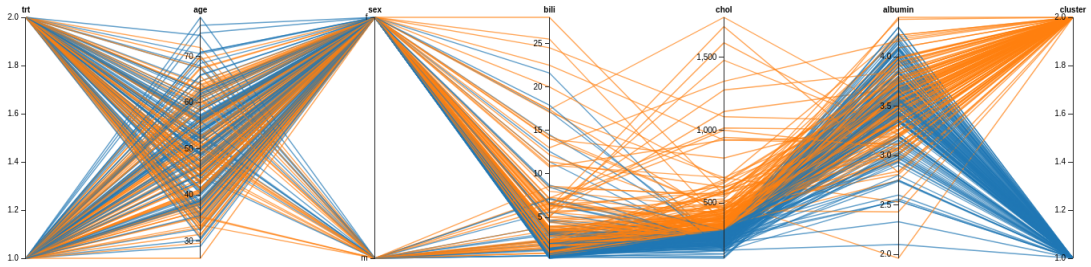

Fig. 5. Parallel coordinate plot after running tSNE on user selected initial variables for clustering and number of clusters, only user selected variables are included with color coded clusters, users can use brushing to select a subset of observations

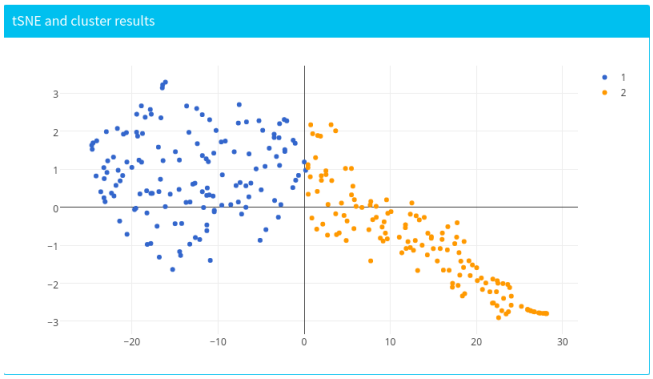

Fig. 6. tSNE output plotted using a 2D scatter plot and colors correspond to clusters selected via K-means, users can hover over for details and zoom in for further inspection

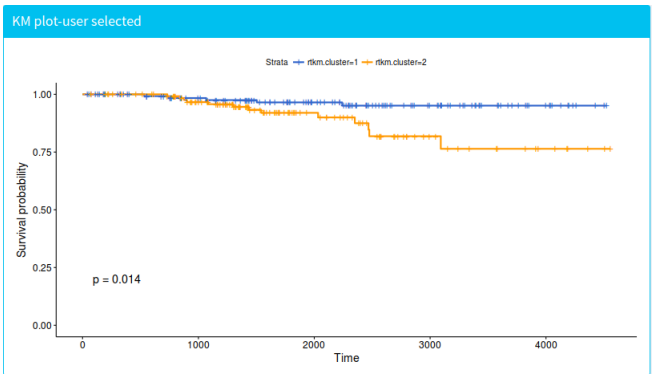

Fig. 8. Kaplan Meier plot of survival probability with groupings using user selected number of clusters

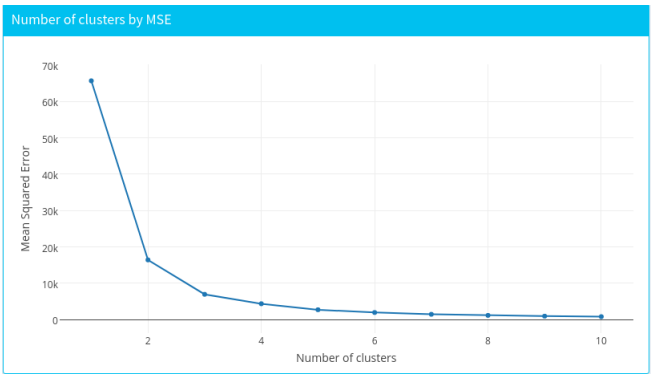

Fig. 7. Mean squared error by number of clusters, optimal number is chosen at the elbow, users can hover over the line to see the MSE values

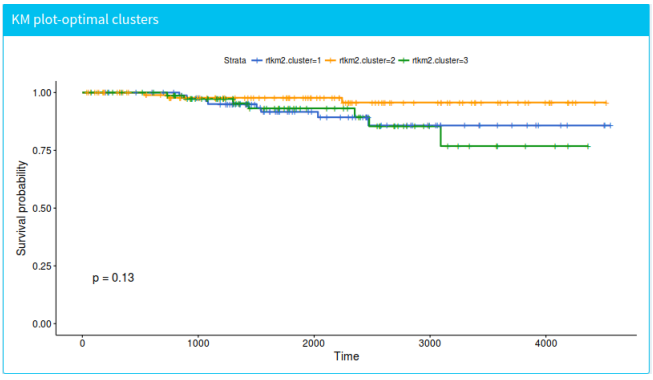

Fig. 9. Kaplan Meier plot of survival probability with groupings using MSE chosen number of clusters by an algorithm referring to the elbow of the MSE plot

Fig. 10. Scatter plot matrix for all variables selected by user for clustering, brushing and linking is available, that is the user can select a subset of points in one plot and the corresponding points are highlighted in rest of the plots.

Fig. 11. Scatter plot matrix for only statistically significantly different variables between cluster from variables selected by user for clustering, brushing and linking is available.

Cluster quality can be judged against the number of clusters selected by user by looking at the MSE plot of cluster numbers. Survival plots are created by cluster membership to show the difference in survival distributions among subjects, any interesting patterns can be further investigated using scatter plot matrices.

Our current implementation can however not handle large datasets, maximum dataset size it can handle is approximately 2000. Any larger datasets would cause delays in tSNE calculations, hence forcing the user to wait for updated visualizations. The application can be scaled using another implementation of tSNE such as the approximated tSNE [17]. In its current form, the visualizations are restricted by shiny, we would like to include more D3 to create the same plots and have more freedom of spacing and utilize screen real estate better. Currently, this application also requires a lot of user input, we would like to automate more parts of the app, such as suggested variable selection based on correlation with the outcome. Also screen real estate on explore view can be better utilized by showing complete scatter plot matrix and scatter plot matrix of significant variables side by side. We have also not yet validated the application on any real end users, we would like to do so in near future and then incorporate the suggested modifications.

8 Acknowledgements

We would like to thank Dr. Tamara Munzner for her valuable feedback throughout this project.

References

[1] W. M. Abdelmoula, B. Balluff, S. Englert, J. Dijkstra, M. J. Reinders, A. Walch, L. A. McDonnell, and B. P. Lelieveldt. Data-driven identification of prognostic tumor subpopulations using spatially mapped t-sne of mass spectrometry imaging data. *Proceedings of the National Academy of Sciences*, 2016.

[2] D. G. Altman and J. M. Bland. Time to event (survival) data. *BMJ*, 317(7156):468–469, 1998.

[3] D. Attali. *shinyjs: Easily Improve the User Experience of Your Shiny Apps in Seconds*, 2016. R package version 0.9.

[4] B. Becher, A. Schlitzer, J. Chen, F. Mair, H. R. Sumatoh, K. W. W. Teng, D. Low, C. Ruedl, P. Riccardi-Castagnoli, M. Poidinger, et al. High-dimensional analysis of the murine myeloid cell system. *Nature immunology*, 15(12):1181–1189, 2014.

[5] M. Bostock, K. Chang, and K. Russell. *parcoords: htmlwidget for d3.js parallel coordinates chart*. R package version 0.4.0.

[6] W. Chang. *shinydashboard: Create Dashboards with 'Shiny'*, 2016. R package version 0.5.3.

[7] W. Chang, J. Cheng, J. Allaire, Y. Xie, and J. McPherson. *shiny: Web Application Framework for R*, 2017. R package version 1.0.2.

[8] P. L. Chia, C. Gedye, P. C. Boutros, P. Wheatley-Price, and T. John. Current and evolving methods to visualize biological data in cancer research. *Journal of the National Cancer Institute*, 108(8), 2016.

[9] B. K. Dhillon, M. R. Laird, J. A. Shay, G. L. Winsor, R. Lo, F. Nizam, S. K. Pereira, N. Waglechner, A. G. McArthur, M. G. Langille, et al. Islandviewer 3: more flexible, interactive genomic island discovery, visualization and analysis. *Nucleic acids research*, 2015.

[10] M. M. Ghassemi, R. G. Mark, and S. Nemati. A visualization of evolving clinical sentiment using vector representations of clinical notes. In *Computing in Cardiology Conference (CinC), 2015*, pages 629–632. IEEE, 2015.

[11] G. Hinton and S. Roweis. Stochastic neighbor embedding.

[12] E. L. Kaplan and P. Meier. Nonparametric estimation from incomplete observations. *Journal of the American statistical association*, 53(282):457–481, 1958.

[13] A. Kassambara and M. Kosinski. *survminer: Drawing Survival Curves using 'ggplot2'*, 2017. R package version 0.3.1.

[14] W. H. Kruskal and W. A. Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American statistical Association*, 47(260):583–621, 1952.

[15] S. Kullback and R. A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[16] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. 1967.

[17] N. Pezzotti, B. Lelieveldt, L. Van Der Maaten, T. Hollt, E. Eisemann, and A. Vilanova. Approximated and user steerable tsne for progressive visual

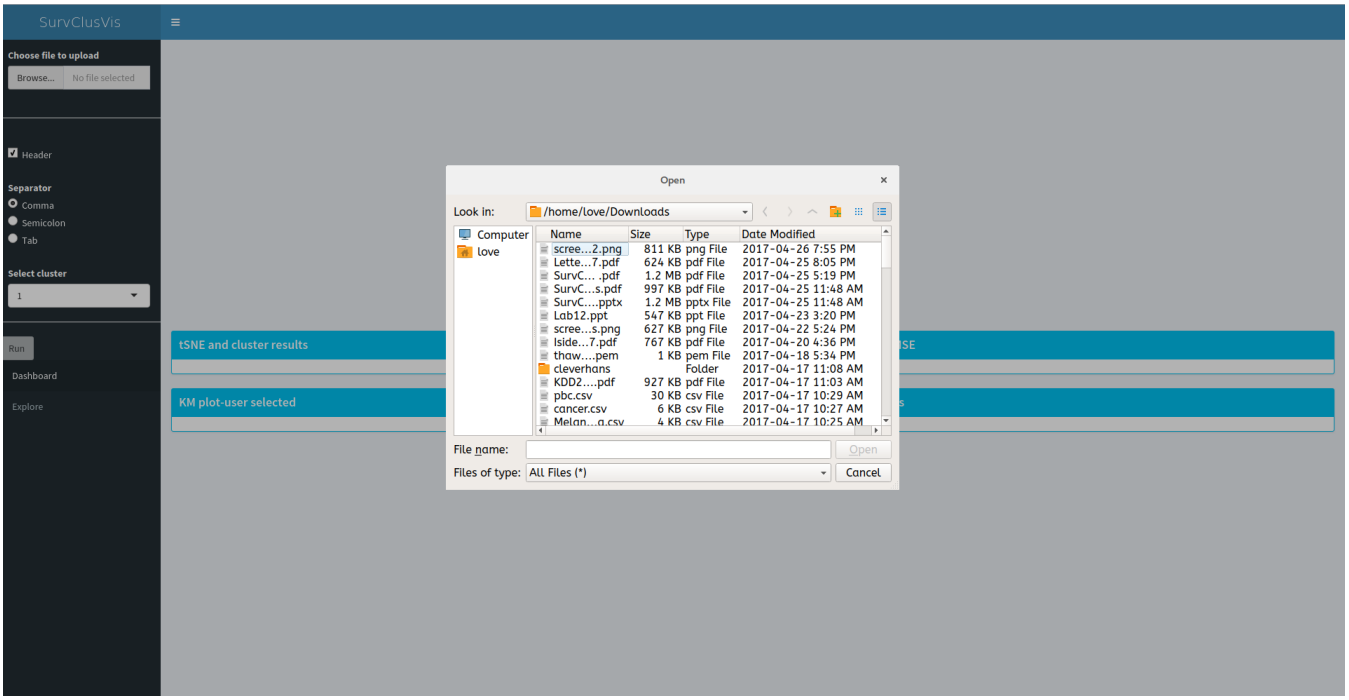

Fig. 12. Welcome screen and popup after clicking the browse button to upload a dataset.

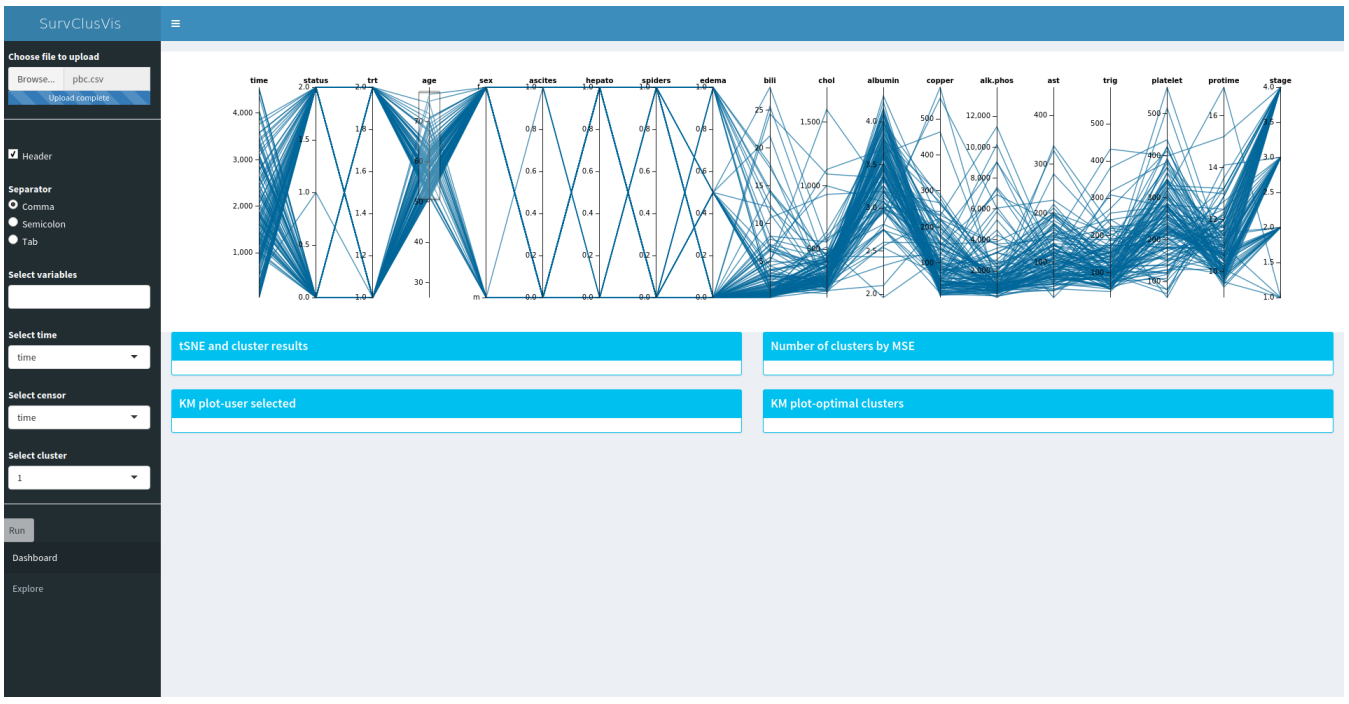

Fig. 13. Updated page with parallel coordinates plot showing distribution of all variables in the dataset with user selected observations using brushing for attribute age.

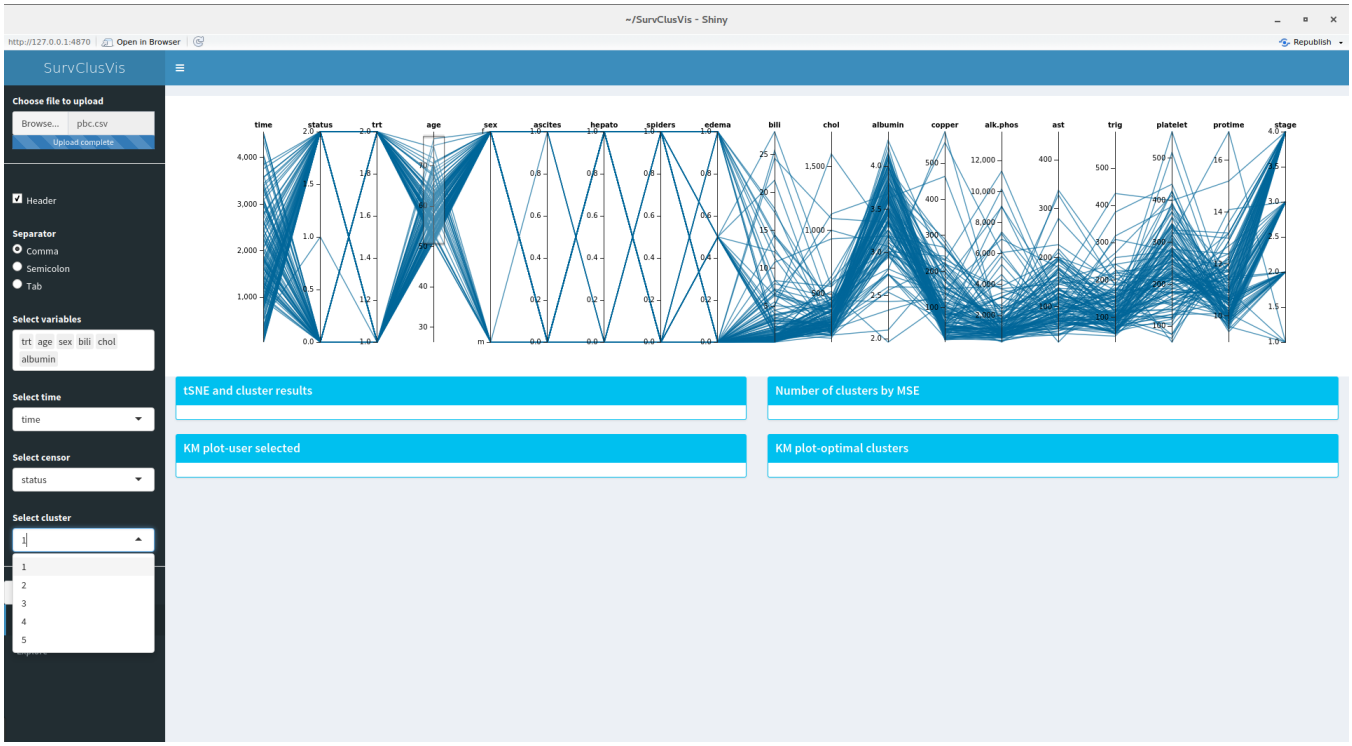

Fig. 14. Selecting variables of interest and number of clusters to investigate from left hand panel.

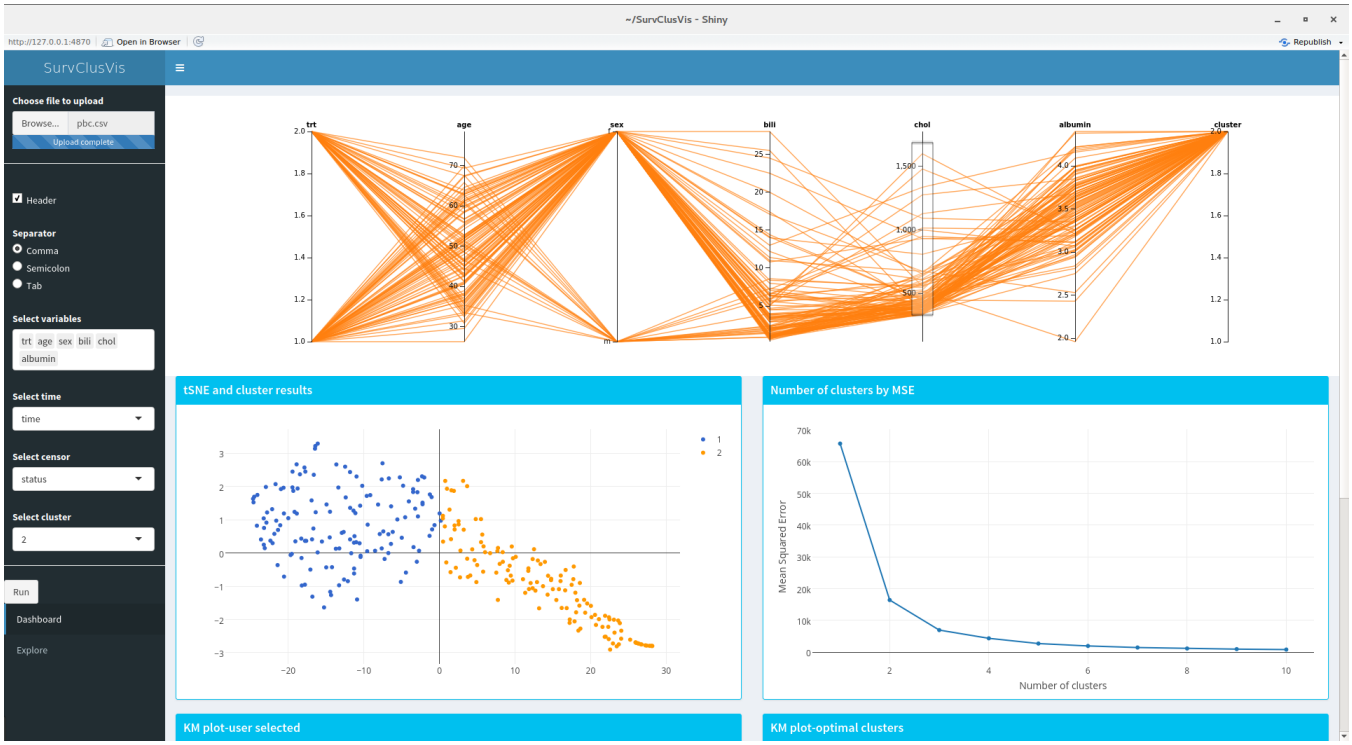

Fig. 15. Selecting variables of interest and number of clusters to investigate, Amy uses brushing to highlight high values of cholesterol constituting majority of one cluster membership

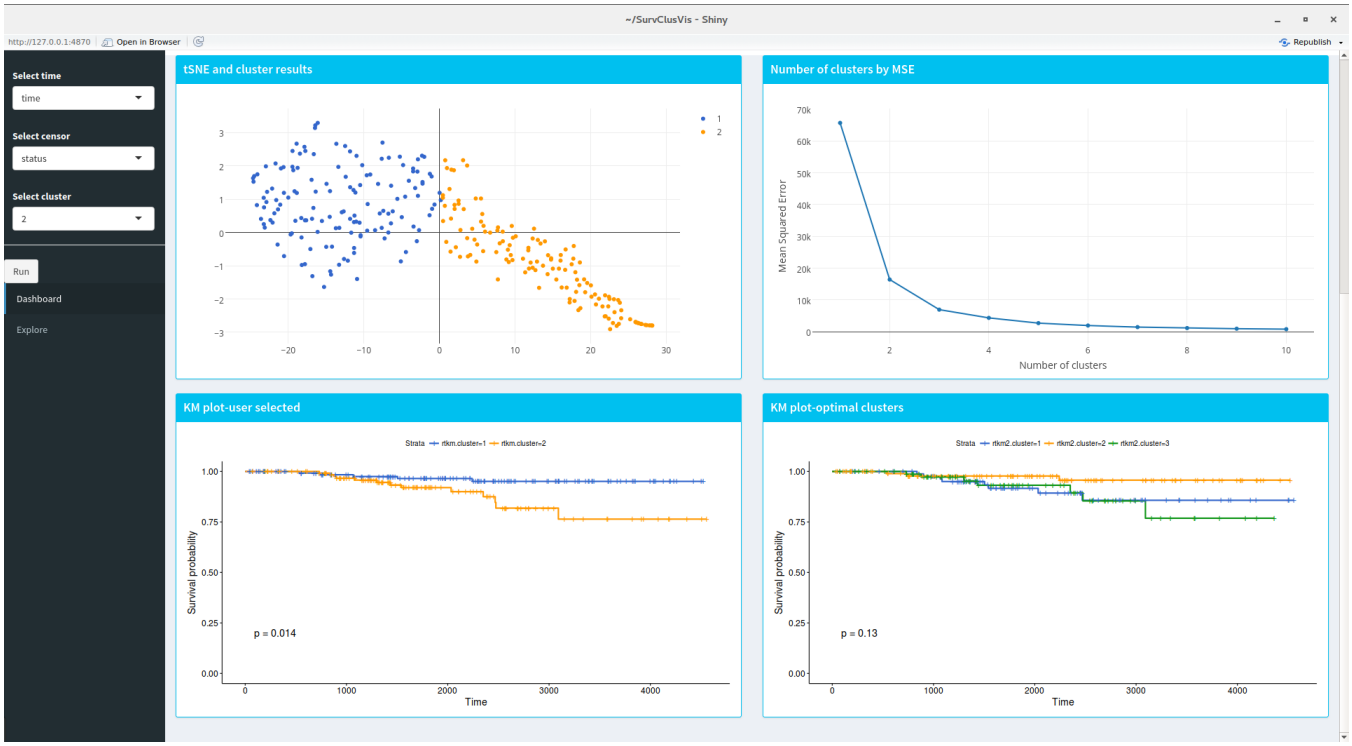

Fig. 16. Kaplan Meir survival plots grouped by cluster membership, left is using user selected number of clusters and right is the optimal number selected using MSE

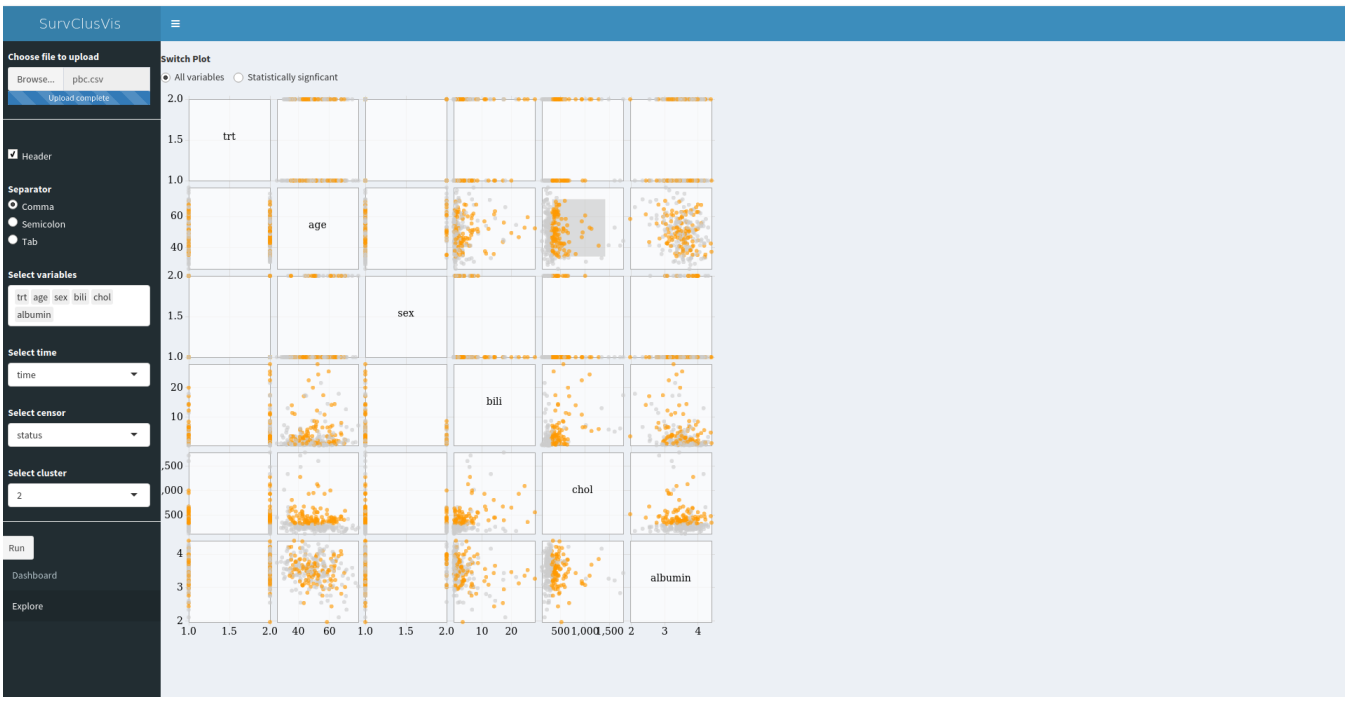

Fig. 17. Scatter plot matrix of all variables used in tSNE and clustering, linking and brushing is available

analytics. *IEEE Transactions on Visualization and Computer Graphics*, 2016.

[18] A. Platzer. Visualization of snps with t-sne. *PloS one*, 8(2):e56883, 2013.

[19] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017.

[20] R. Sharan, A. Maron-Katz, and R. Shamir. Click and expander: a system for clustering and visualizing gene expression data. *Bioinformatics*, 19(14):1787–1799, 2003.

[21] C. Sievert, C. Parmer, T. Hocking, S. Chamberlain, K. Ram, M. Corvellec, and P. Despouy. *plotly: Create Interactive Web Graphics via 'plotly.js'*, 2016. R package version 4.5.6.

[22] Student. The probable error of a mean. *Biometrika*, pages 1–25, 1908.

[23] A. Sturn, J. Quackenbush, and Z. Trajanoski. Genesis: cluster analysis of microarray data. *Bioinformatics*, 18(1):207–208, 2002.

[24] G. Tarr. *pairsD3: D3 Scatterplot Matrices*, 2015. R package version 0.1.1.

[25] T. M. Therneau and P. M. Grambsch. *Modeling survival data: extending the Cox model*. Springer Science & Business Media, 2013.

[26] T. M. Therneau and T. Lumley. Package survival, 2016.

[27] H. Thorvaldsdóttir, J. T. Robinson, and J. P. Mesirov. Integrative genomics viewer (igv): high-performance genomics data visualization and exploration. *Briefings in bioinformatics*, 14(2):178–192, 2013.

[28] L. Van Der Maaten. Barnes-hut-sne. *arXiv preprint arXiv:1301.3342*, 2013.

[29] L. Van Der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[30] H. Wickham and R. Francois. *dplyr: A Grammar of Data Manipulation*, 2016. R package version 0.5.0.

[31] W. Xu, X. Jiang, X. Hu, and G. Li. Visualization of genetic disease-phenotype similarities by multiple maps t-sne with laplacian regularization. *BMC medical genomics*, 7(2):S1, 2014.