**Principal Component Analysis**

**(PCA) Application**

**User’s Guide**

Emilia Gan

10 June 2016

**1 Introduction**

This application uses the R package Shiny (ref\*\*) to create a graphical user interface with which a user can upload a data file and perform principal component analysis on the uploaded data. The application produces a plot of the first two principal components, which can be downloaded. In addition to providing the capability of R to perform this task to users who are not familiar with the R programming language, the application has the further advantage of providing a level of flexibility beyond the typical R script. Specifically, the ease with which the user can toggle between having the row elements or the column elements displayed in the plot and the ease with which new data files can be loaded, make analysis using the Shiny application more efficient even for expert R programmers.

**2 Setting Up Your R Environment**

*2.1 Installation of R, RStudio, and Required R Packages*

In order to run this application locally, you will need to have the free R software environment and the open-source RStudio integrated development environment (IDE) installed. Downloads for R are available for Unix platforms, Windows, and Mac OS. Installation instructions for each of these can be found using the following links:

* The R Project for Statistical Computing: <https://www.r-project.org/>
* RStudio IDE: <https://www.rstudio.com/>

Once you have this software installed, you need to install the following R packages:

* *shiny*: Web application framework for R
* *edgeR*; Empirical analysis of digital gene expression data in R
* *ggplot2*: An implementation of the grammar of graphics
* *limma*: Linear models (for all gene expression technologies)

Installing R packages from within the RStudio IDE is quite straightforward:

1) Open the RStudio IDE.

2) Click on the “Packages” tab to see a listing of currently installed packages.

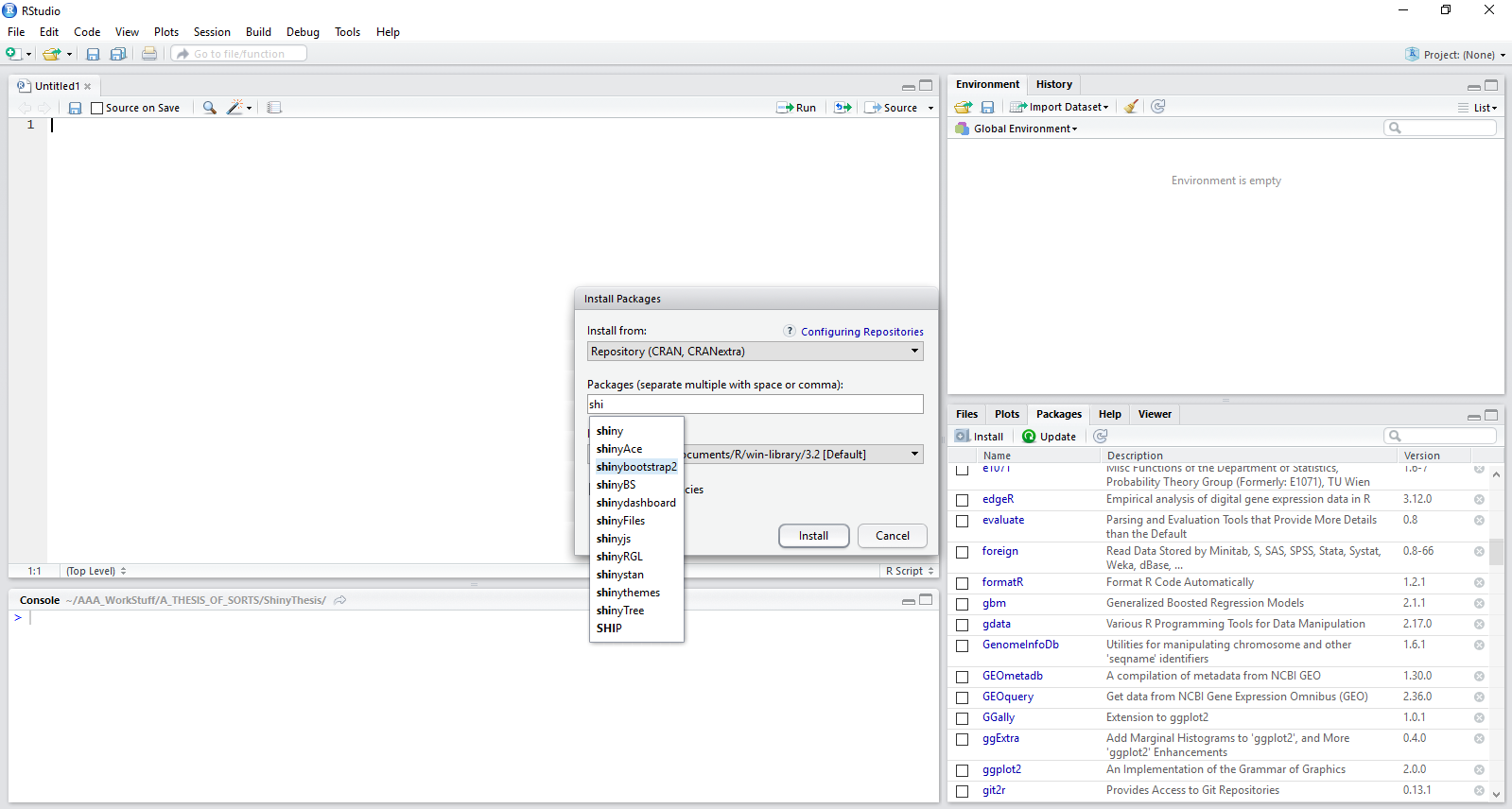
3) All of the packages required to run the PCA Application are in the Comprehensive R Archive Network (CRAN) repository. Accordingly, they can be easily installed from within the RStudio IDE by following this sequence:

a) Click on the “Install” tab. A pop-up window will appear.

b) Type in the name of the package you wish to install.

c) Check the box labeled “install dependencies” to ensure that any packages required by the package you are requesting are automatically installed as well.

A screenshot illustrating the package installation process is provided below:

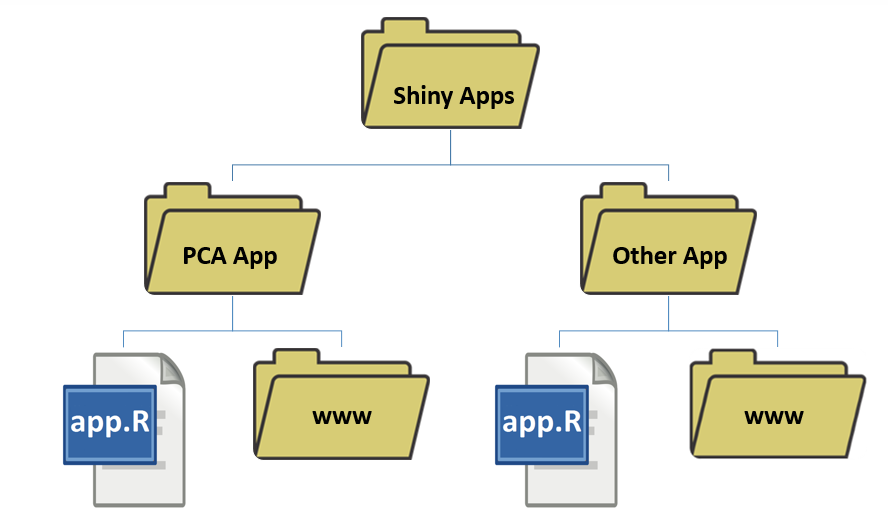


**Figure 1:** Installing R packages within the RStudio IDE.

*2.2 Downloading the PCA Application Files*

All R code files needed to run the PCA Application are publically available online on GitHub. The work is shared under an MIT License, meaning that it may be used without limitations, provided attribution is given to the author and the author is released from any liability relating to use of the code.

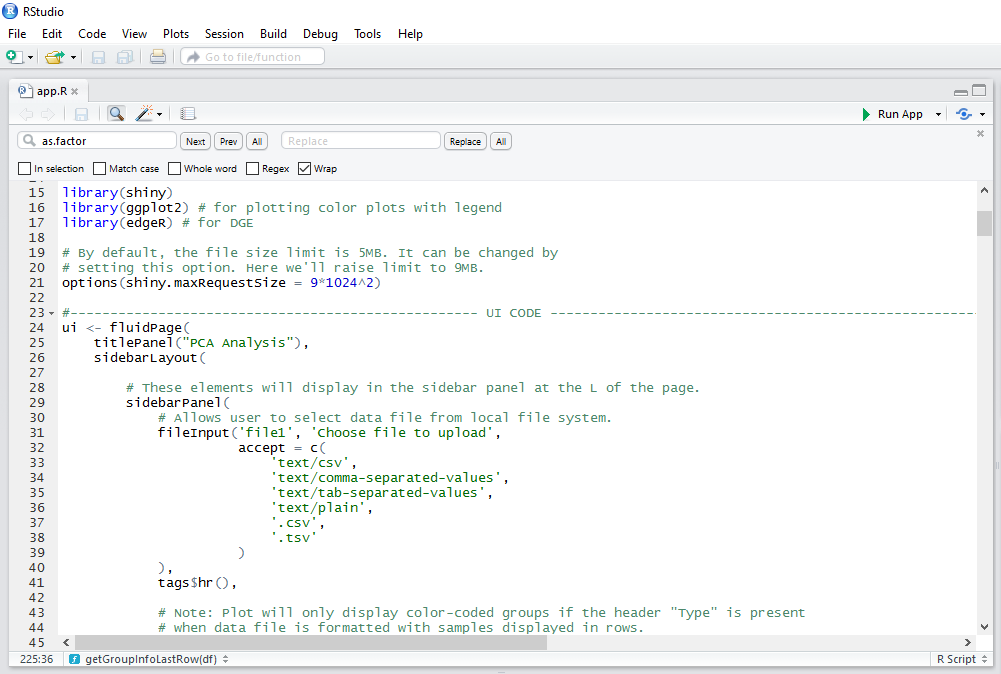
Once downloaded, the application files may be stored anywhere in your local file system. For the applications to work most efficiently, the downloaded files for the PCA Application should be kept together in a single folder. R requires that each Shiny application be names “app.R” in order to be recognized and run as an application, without the code needing to be manually selected and highlighted. In addition, files that contain material the Shiny application needs to access, such as local image files, need to be stored in a folder named “www.” Keeping each app.R file and its associated www folder in its own, informatively named directory makes organizing this, and other, Shiny applications possible. The file hierarchical diagram shown below illustrates one option for organizing Shiny application files.



**Figure 2:** File organization for Shiny applications.

*2.3 Running the PCA Application*

Once the R environment has been set up, the PCA Application can be run from within R Studio. To do this, open RStudio and open the PCA Application’s “app.R” file. After the code has opened in RStudio, the application can be deployed by clicking the “Run app” button as illustrated in the screen shot provided in Figure 3. Once the application has deployed, its functionality is accessed exclusively through the graphical user interface (GUI).



**Figure 3:** Deploying the PCA Application

Open the PCA Application’s app.R file in RStudio, then deploy the application from within RStudio by clicking the “Run App” arrow in the top right corner (outlined in red).

**3 Overview of features**

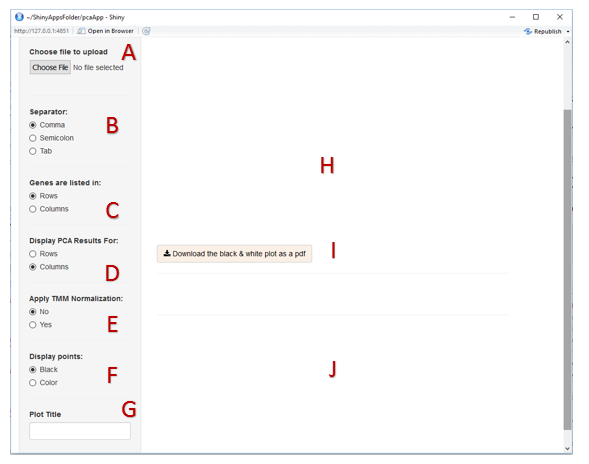
*3.1 PCA Application Layout*

Upon launching the PCA Application, the GUI opens. Figure 4 points out the various features of the PCA Application layout.

A. File upload box that allows the user to select an appropriately formatted file to upload from any location in the local file system.

B. Radio buttons to provide information on the file type to the application.

C. Radio buttons to provide information to the application on how the data are arranged in the data file. It is CRITICAL that the program know whether genes are arranged listed by row or by column. Inaccurate information will cause invalid results and, potentially, abrupt program termination.



**Figure 4:** PCA Application Layout

D. Radio buttons to indicate whether the PC 1 versus PC 2 plot should plot the elements that arranged in rows or in columns in the uploaded data file.

E. Radio buttons to enable the user to indicate whether or not TMM normalization should be applied to the data. TMM normalization is discussed in Section 3.6 of this document.

F. Radio buttons to indicate if the plot’s points should be displayed black or in color. Color is only an option when group (categorical) information is provided.

G. Text entry box that allows the user to input a title for the plot(s) created.

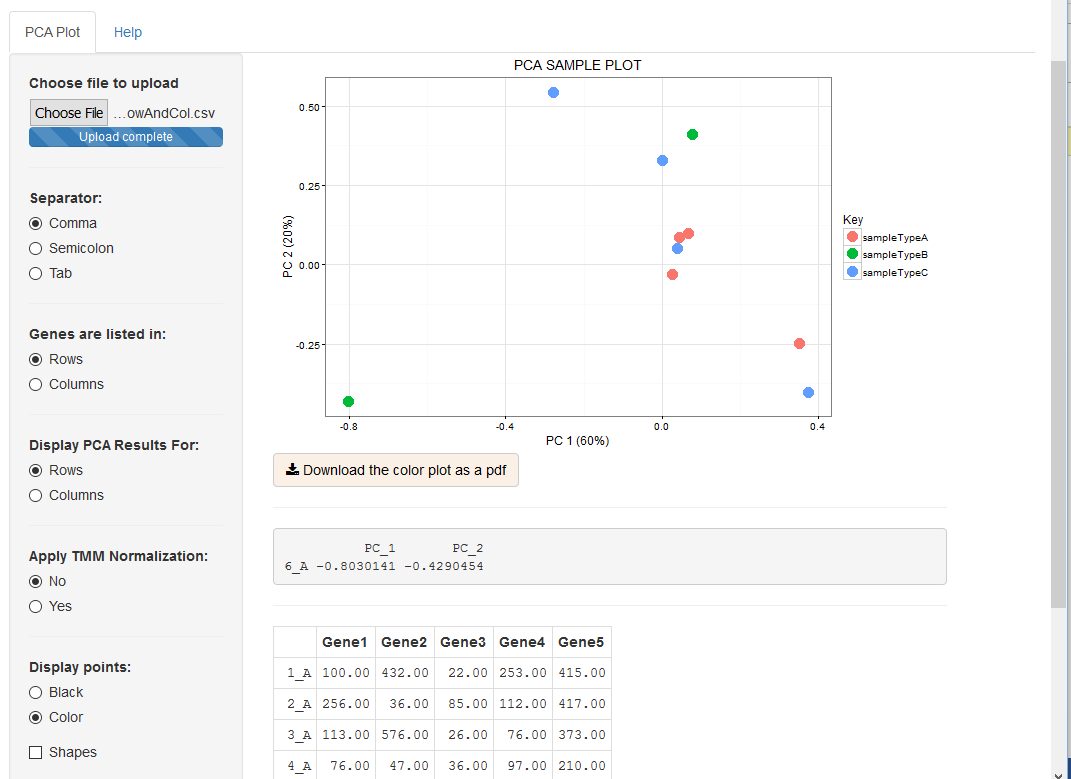
H. Main plot display area.

I. Download button to save the plot created as a .pdf file.

J. Area for displaying data file contents in tabular form.

(Not shown in the figure: Tab to a “Help” page that describes the format required of the data files uploaded to the application and provides usage instructions. Located off screen in this screenshot, at the top of the application.)

*3.2 Uploading Files and Selecting Features for Plotting*

Upon uploading an appropriately formatted file, a scatter plot of the first two principal components is displayed. The plot axes include information about the variance accounted for by each principal component.

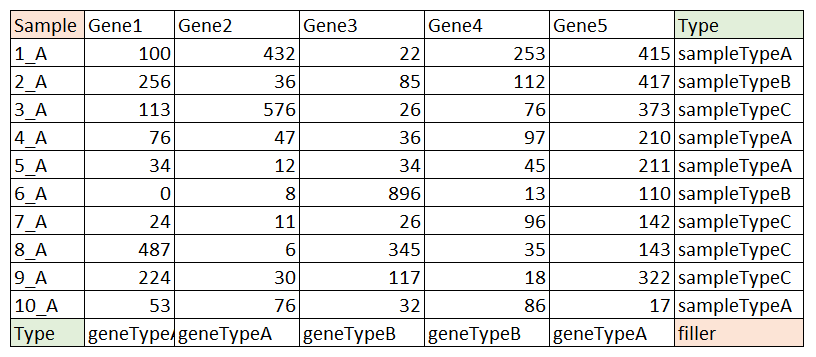
**Figure 5:** PCA Application with Uploaded Data File and Generated Plot.

Also visible is the tabular representation of the data file at the bottom of the display. The plot, itself, is interactive. Mousing over the plot causes the cursor to change to a “+” symbol, which can be moved over any point in the plot. Upon clicking, that point’s name and plot coordinates are displayed in the grey box below the plot. In Figure 5, the point selected was the point representing sample 6\_A, which is located in the lower left corner of the plot. A title may be entered via the text entry box labeled “Plot Title” in the sidebar panel. The data being examined can be easily changed. To select a new data file, all that is required is to return to the file upload box and select a new file from the local file system.

*3.3 Permitted File Formats*

The PCA Application can accept files in comma-separated value format (.csv or .txt), tab-separated value format (.tsv or .txt), or in semicolon-separated value format (.txt). The file should have both column names and row names for the data. In addition, the file may contain optional type information, if the elements (typically samples being sequences and genes) belong in different categories.

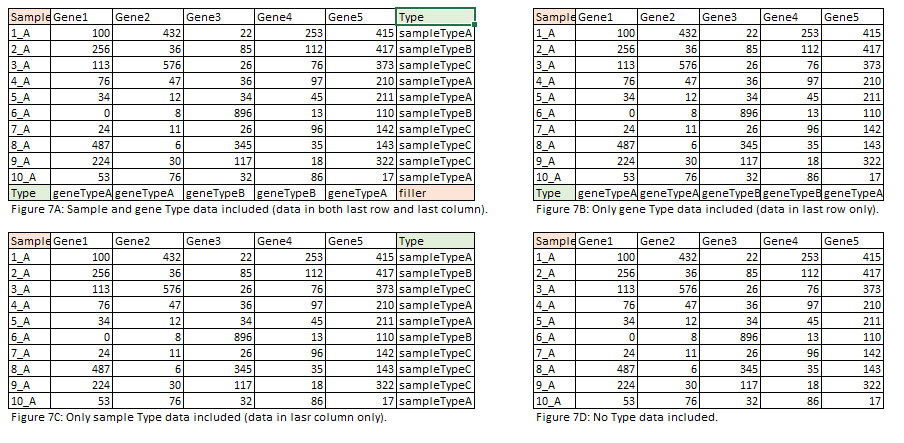
The formatting of the data file is critical to the proper operation of the PCA Application. Figure 6 provides an example of the format that should be followed when preparing data for use with the application.



**Figure 6:** Proper Formatting of Data

There must be text in the top-left and bottom-right cells. However, the actual words “Sample” and “filler” are not critical and may be changed. Including type data is optional, but if it is included for either samples or genes, the word “Type” is critical, as the data will not be processed appropriately if this keyword is missing.

Figure 6 displays a data file as it might appear in a program such as Excel, before being converted to a comma-separated value file, or other permitted file format. All data files are required to include both column names and row names. For the use the application was designed for, these names will identify both the samples and genes included in the data set. The cells highlighted in red MUST contain text, as the data file is not permitted to contain any empty cells, but the actual word chosen is not critical. Type (or categorizing) information is optional. However, when such data are included, the word “Type” must be used in the cells highlighted in green in Figure 6. Keeping this format in mind, the red-highlighted cell in the lower-right corner is only an issue when BOTH row elements and column elements have associated type data. Figure 7 displays thumbnail versions of each of the permitted formats.

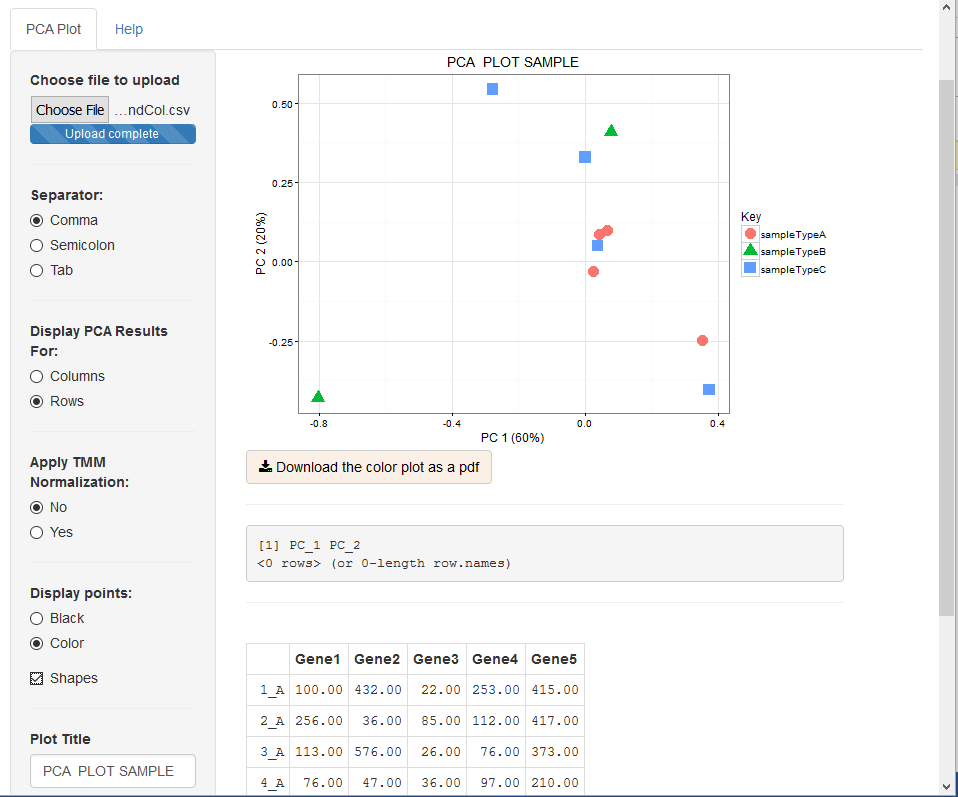


**Figure 7:** Permitted file formats

Examples of permitted file formats illustrating, once again, that column names and row names are required (first column and first row and that type (categorical) information is optional, but, if present, must be identified using the keyword “Type.” The decision to have genes in columns and samples in rows or to reverse them is up to the user. The application can process files in either orientation.

*3.4 Displaying Categorical (Type) Information*

If type/categorical information is included for the elements being plotted, selecting the “Color” radio button under “Display points” will color code points according to their category. A legend listing the categories appears to the right of the plot. In addition, upon selection the “Color” option a “Shapes” checkbox also appears, further emphasizing the different categories by assigning each category a unique shape. The “Shapes” option is offered to facilitate the creation of plots in which categories remain distinguishable even in black and white reproductions of the plot. Figure 8 illustrates each of these features.



**Figure 8:** Displaying Type (categorical) Information

If type/categorical information is included for the elements being plotted, these categories may be indicated using different colors and shapes.

*3.5 Interpreting the PC1 versus PC2 Plot*

Principal component analysis is a method that provides investigators with a low-dimensional representation of the data set that still captures much of the variation in the data set. The first principal component is the normalized linear combination of the set of features on which data were collected (for this application, generally the gene expression counts) which has the greatest variance. The second principal component is the linear combination with the highest variance that is also uncorrelated with the first principal component. In effect, the original data are being projected into a lower dimensional subspace -- a plane created by the two principal components. These projected points are used to create the plot.

The proportion of the data set’s variation that is being captured by each principal component is stated in the axes labels. In Figure 8, PC 1 accounts for 60% of the variation, with PC 2 explaining 20% more, bringing the cumulative variance captured by the plot to 80%. The visualization created using these two principal components has the potential to reveal interesting patterns in the data, and is a valuable tool for exploratory data analysis.

*3.6 Trimmed Mean of M-values Normalization*

Normalizing the RNA-Sequencing counts data is a standard step in the data analysis pipeline. However, there is no consensus on the “best” normalization method to use, and a number of different methods are commonly used. The PCA Application offers the option of normalizing the uploaded data using the Trimmed Mean of M-values (TMM) method described in the article by Robinson and Oshlack (6).

The TMM normalization method uses raw counts data to estimate scaling factors to ensure that genes being expressed at the same level in different samples are not erroneously flagged as being differentially expressed. The base assumption of the method is that the majority of genes are not differentially expressed. The method estimates the ratio of RNA production between samples by calculating a weighted trimmed mean of log expression ratios. The PCA Application uses the default setting in the edgeR package for the percentage of data trimmed before the mean calculation.

In acknowledgement of the existence of other normalization techniques and the possibility that the user of the application may wish to use some method other than TMM, whether or not the data being plotted undergo the normalization or not is user-controlled via the “Apply TMM Normalization” set of radio buttons. The user thus has the option of conducting the normalization step prior to uploading the data file for plotting, or bypassing this step altogether to look at raw counts data.

*3.7 Downloading Plots*

The plots generated by the PCA Application may be saved to the local file system as .pdf files by clicking on the “Download” button located under each plot.

**4 Troubleshooting and How to Get Help**

All attempts have been made to thoroughly test this application. Should features fail to work as described, the most common source of the problem is likely to be an improperly formatted data file. Please refer to the diagrams showing accepted formats.

It is also important to click the appropriate radio button for the file format being uploaded. The app will not process a comma-separated value file, for example, despite its being a permitted format, if the radio button clicked indicates that it should be uploaded a tab-separated values file.

This software is being made publicly available for all uses, including modification. It is thus offered with no guarantees or support provided. However, the software author may be contacted at [efgan@uw.edu](mailto:efgan@uw.edu), and will provide assistance when possible.

**5 References**

1. R Core Team (2015). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

2. RStudio Team (2015). RStudio: Integrated Development for R. RStudio, Inc., Boston, MA URL <http://www.rstudio.com/>.

3. Institute for Statistics and Math (Wirtschaftsuniversitat Wien). The Comprehensive R Archive Network, <https://cran.r-project.org/> , accessed 8 May 2016.

4. The Open Source Initiative. The MIT License (MIT), <https://opensource.org/licenses/MIT> , accessed 8 May 2016.

5. James, Gareth, et al. *An Introduction to Statistical Learning*. New York: Springer, 2014.

6. Robinson, Mark D and Alicia Oshlack. “A scaling normalization method for differential expression analysis of RNA-seq data.” *Genome Biology* (2010), 11:R225.

7. Robinson MD, McCarthy DJ and Smyth GK (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics 26, 139-140