**MANUAL:**

**VIASVA: A Shiny App**1 **for Visual Iteratively Adjusted Surrogate Variable Analysis**

**Author: Nathan Lawlor**

**Contact:** [**Nathan.lawlor03@gmail.com**](mailto:Nathan.lawlor03@gmail.com)

**Other Resources:**

**Manuscript on bioRxiv:** [**https://www.biorxiv.org/content/early/2017/06/18/151217**](https://www.biorxiv.org/content/early/2017/06/18/151217)

**Bioconductor package (iasva):** [**https://www.bioconductor.org/packages/devel/bioc/html/iasva.html**](https://www.bioconductor.org/packages/devel/bioc/html/iasva.html)

**GitHub page:**

[**https://github.com/nlawlor/iasva\_shiny**](https://github.com/nlawlor/iasva_shiny)

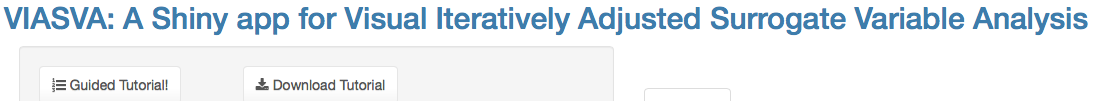
[**Interactive Tutorial**](#Interactive_Tutorial)

**Quick Start Guide:**

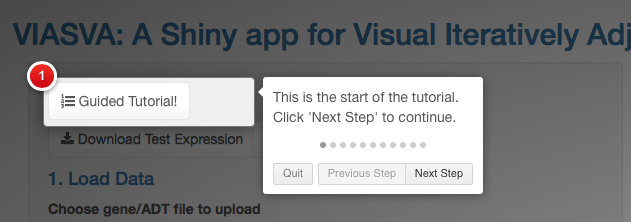
1. [**Load Data**](#Load_Data)
2. [**Data Preprocessing**](#Data_Preprocess)
3. [**Specify Known Factors to Adjust For**](#Specify_Known_Factors)
4. [**IA-SVA Analysis**](#IASVA_Analysis)
5. [**Data/QC**](#Data_QC)
6. [**Surrogate Variables**](#Surrogate_Variables)
7. [**Pairwise Surrogate Variable**](#Pairwise_Surrogate_Variable)
8. [**Identify Marker Genes**](#Identify_Marker_Genes)
9. [**Dimension Reduction and Visualization**](#Dimension_Reduction_and_Visualization)
10. [**Gene Enrichment Analysis**](#Gene_Enrichment_Analysis)
11. [**References**](#References)

**Interactive Tutorial**

In addition to this manual, you can choose to have an interactive tutorial of this shiny1 app by launching the app and clicking this button near the top left of your webpage:



Upon clicking this button, the tutorial should begin and you should see this:

****

Navigate through this tutorial using the “Next Step” and “Previous Step” buttons. To exit out of this tutorial, click the “Quit” button or anywhere on your screen outside of the white boxes.

1. **Load Data**

In the first section of the app, the user must load their data:

1. **(Required)** A file containing gene expression/ADT level data, typically in matrix format. The matrix should be formatted so that rows represent the genes/markers and the columns represent the samples. For information about the exact file format of the expression data, please see the example data provided (information about downloading below).
2. (**Optional)** A file containing sample metadata, typically in matrix format. The matrix should be formatted so that each row represents a sample and the columns represent different metadata variables (e.g., age, gender, disease state, etc.). For information about the exact file format of the sample metadata, please see the example data provided (information about downloading below).

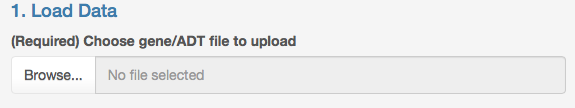
For convenience, an example cell gene expression and sample metadata set (single cell RNA-seq data; human peripheral blood mononuclear cells (PBMCs) treated with interferon beta from Kang et al. 20172: <https://www.nature.com/articles/nbt.4042>) are provided and are available for download by clicking the two buttons near the top left corner of the app:



These two example files are .csv files and may be uploaded in the next steps to test out the app.

**Loading Gene Expression (Required)**

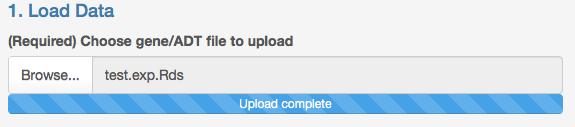
Users may upload their own expression file (or the downloaded test file) by clicking the “Browse” buttons pictured below:



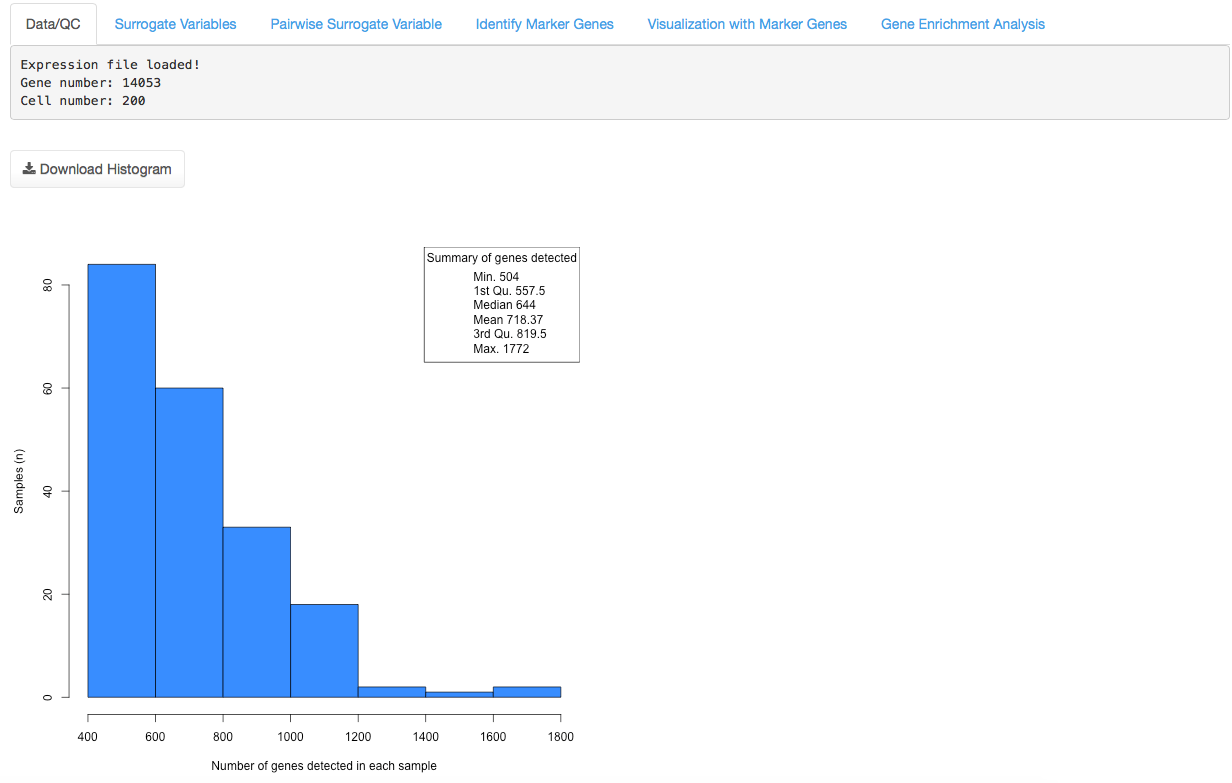
Input files must be in the format of either: .Rds, tab-delimited text, or .csv. Currently, the maximum size for input files is 2GB.

**Note:** The app will fail if you input file is larger than 2GB.

Upon successful loading of your input gene expression file, the user interface should look like this:



Additionally, the main panel of the app should display a message detailing the file has successfully been loaded and the dimensions of the data (number of genes and number of cells) and the following histogram in the **Data/QC** main panel area:



The histogram portrays the number of genes detected in each sample. A gene is defined as detected in a sample if it has at least one read count in that sample. A PDF version of this histogram can be downloaded by clicking the “Download Histogram” button.

**Loading Sample Metadata (Optional)**

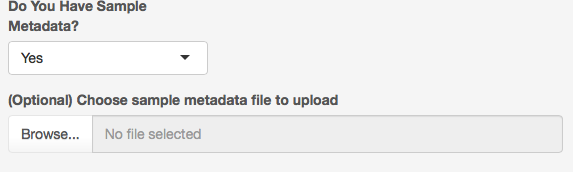
Afterwards, users may upload a sample metadata file if they have it.

**Note:** A sample metadata file is not required to use this app, but is strongly recommended. Providing a sample metadata file with detailed characteristics (age of donor, cell type identity, treatment/disease status, donor of origin) for each sample will improve accuracy of down-stream analyses.

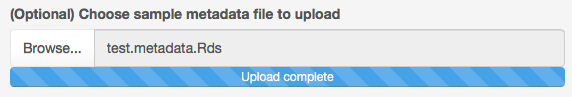
If users do not have a metadata file, leave the drop-down menu at “No”.



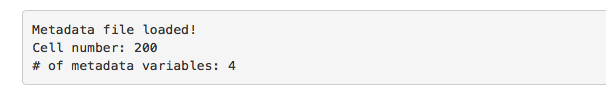
Otherwise, if you do have a metadata file, click the drop-down menu and change this to “Yes”:



The user interface should be updated to allow upload of the file. Upload the file:



Afterwards, you should see the following message below the histogram:

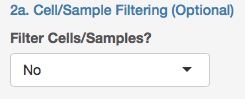


This indicates the dimensions of the metadata file (number of samples/cells and number of metadata variables).

1. **Data Preprocessing**

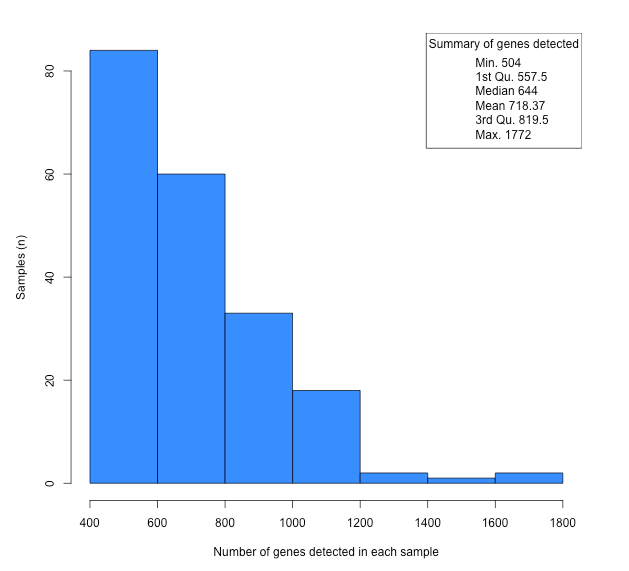
After providing input files, the users have the choice of further pre-processing their data. Specifically, there are **three** optional steps.

**2a. Cell/Sample Filtering**

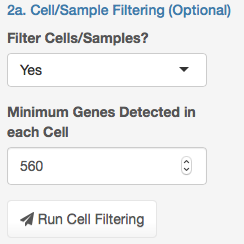
****

The first optional step is to remove cells with a low number of detected genes. In scRNA-seq data, cells with few detected genes may be indicating the cell is of low viability. If this option is desired, click the drop-down box and change to “Yes”, enter a numeric value for the minimum number of genes that should be detected in each cell, then click the “Run Cell Filtering” button.

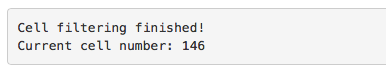
**Note:** For assistance in choosing this number, consult the previous histogram showing the number of detected genes in each cell.



The previous histogram informs us that the lower quartile of detected gene number is roughly 557 genes. For this example, we will choose 560 as our minimum detected gene threshold.

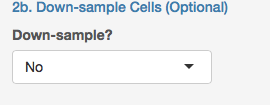


After clicking this button, you should see the following message displayed in the main **Data/QC** panel:



This indicates that the cell filtering has finished. Of our original 200 cells, 54 were removed and now 146 remain.

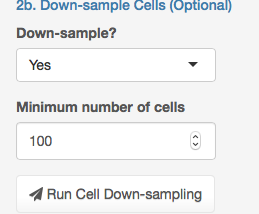
**2b. Down-sampling of Cells**

****

Often, scRNA-seq datasets are quite large consisting of thousands of individual cells. This can lead to long and frustrating computation times! For this reason, users may choose to randomly obtain a subset of their data to make computation time quicker.

**Note:** Down-sampling to a specific cell number may reduce the statistical power of down-stream analyses.

In this example, we will randomly down-sample our data from 146 cells, to 100 random cells:

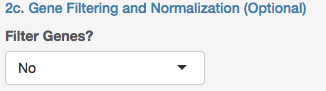


After clicking the “Run Cell Down-sampling” button, the following message will be displayed in the main **Data/QC** panel:



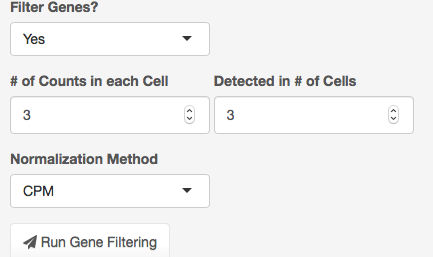
Which will indicate down-sampling has finished and will provide the updated sample/cell number.

**2c. Gene Filtering and Normalization**



Lastly, users have the choice of removing genes that are lowly expressed across multiple cells/samples. In scRNA-seq data, lowly expressed genes often do not provide interesting biological information and can be discarded.

Upon clicking and changing the drop-down menu to “Yes” the user interface should look like this:



First, designate the criteria to filter genes: (i) specify the # of read counts a gene must have in each cell to be considered as detected, and (ii) the number of cells the gene needs to be detected in. In the example above, we are specifying to keep genes with 3 or more counts in 3 or more cells. Second, specify one of 4 normalization methods:

1. CPM = counts per million. This method from the “edgeR”3 R package normalizes gene counts by the sample’s total counts and a scaling factor of one million.
2. Quantile = quantile normalization as implemented in the “preprocessCore”4 R package via the normalize.quantiles() function.
3. scran = single cell RNA-seq normalization method by deconvolving size factors from cell pools. This method from the “scran”5 R package normalizes gene counts using the computeSumFactors() function.
4. None = no normalization applied to the data.

Lastly, click the “Run Gene Filtering” button, and the following message should be displayed in the main **Data/QC** panel:



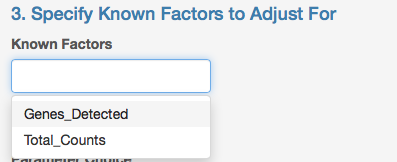
This indicates that the gene filtering and normalization is done. From our original matrix of 14,053 genes, now only 414 genes remain.

1. **Specify Known Factors to Adjust For**

**No Metadata file provided**

In this next step, users will choose known sources of variation that they would like to adjust for prior to estimation of surrogate variables.

If the user did not upload a metadata file, the following “Known Factors” will be displayed:

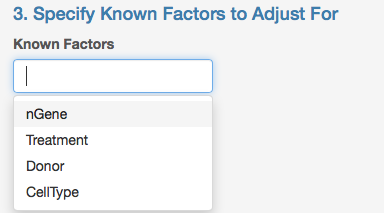
****

1. Genes\_Detected: the number of detected genes (at least one read count in the cell) in each sample.
2. Total\_Counts: the total number of read counts in the sample.

**Note:**Users must specify at least one of these factors before proceeding to the next step.

**User provided a Metadata file**

If the user provided their own metadata file, clicking on the “Known Factors” box should display all of the metadata variables specified in that file. In the example test.metadata.Rds file, the following Known Factors are as follows:



Note that the names of these known factors will be displayed as they are exactly provided in the metadata file.

Select a known factor to adjust for that variable in the downstream analysis. Users may choose multiple known factors (in this example, I chose the “Donor” and “nGene” variables:

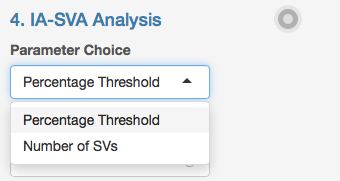


**Note: Users may remove variables from the “Known Factors” box using the “Backspace” or “Delete” key on your keyboard.**

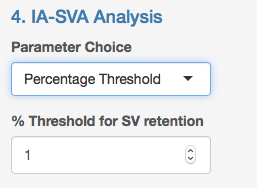
In the example above, I chose to adjust for the technical sources of variation: differences due to number of genes detected (“nGene”) and sample donor of origin (“Donor”), to hopefully maximize the biological variation we can discover in the data.

**IV.** **IA-SVA Analysis**

The fourth step involves adjusting parameters prior to running the IA-SVA algorithm. Users may choose to conduct their analysis by selecting one of two IA-SVA parameter choices: “Percentage Threshold” or “Number of SVs” from the drop down menu:

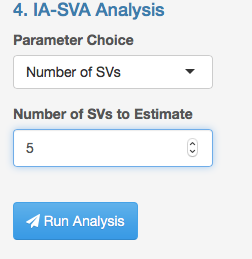


Upon selecting “Percentage Threshold”, the user interface should change to this:



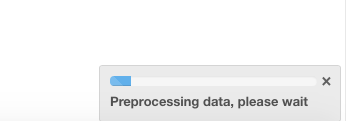
Where “% Threshold for SV retention” is the numeric percentage threshold for SV retention. IA-SVA computes the percentage of unmodeled variance explained by the putative hidden factor and compares it with the user-defined threshold. If the percentage is greater than the threshold, the SV is retained.

Alternatively, users may select the “Number of SVs” option to limit the number of significant SVs identified for retention. This option is useful when analyzing highly dimensional or complex data to avoid identifying a surplus of SVs.

****

Following parameter selection, click the blue “Run Analysis” button.

Upon clicking this button, the user should see a message prompt in the bottom right of the screen:



**V.** **Data/QC**

By default, the Data/QC panel will be the first tabbed panel that is displayed in the main panel area. Once the IA-SVA algorithm finishes running, the Data/QC panel should now display information about your input gene/ADT dataset. For example:

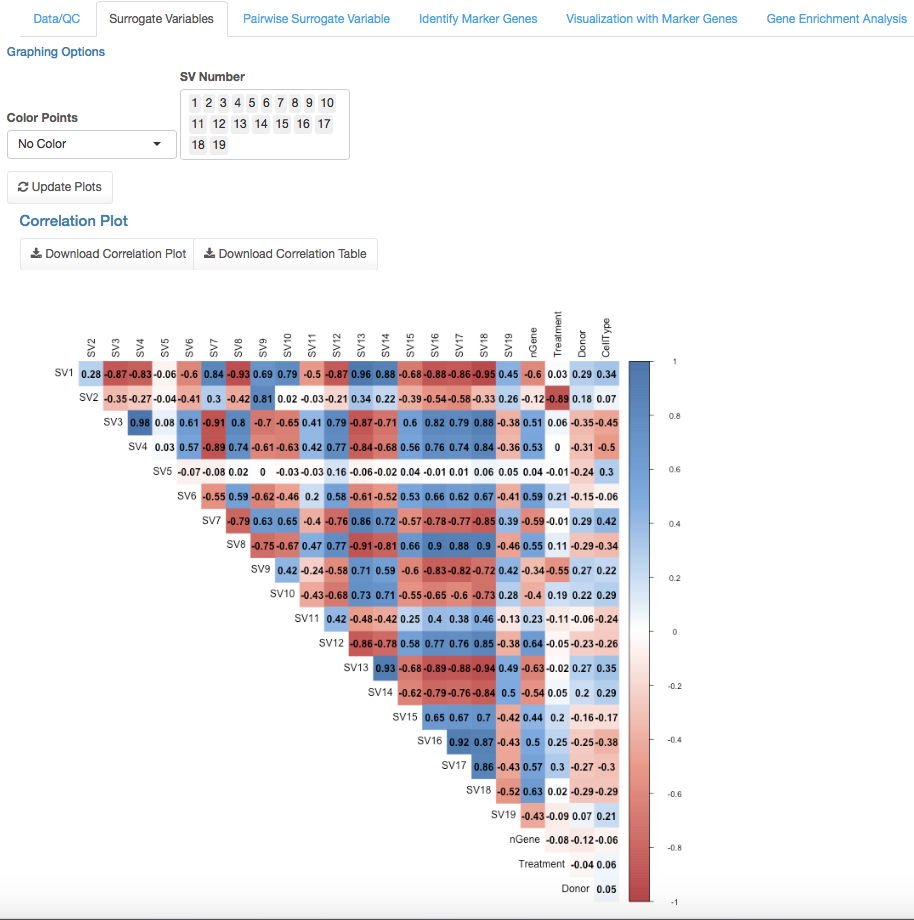


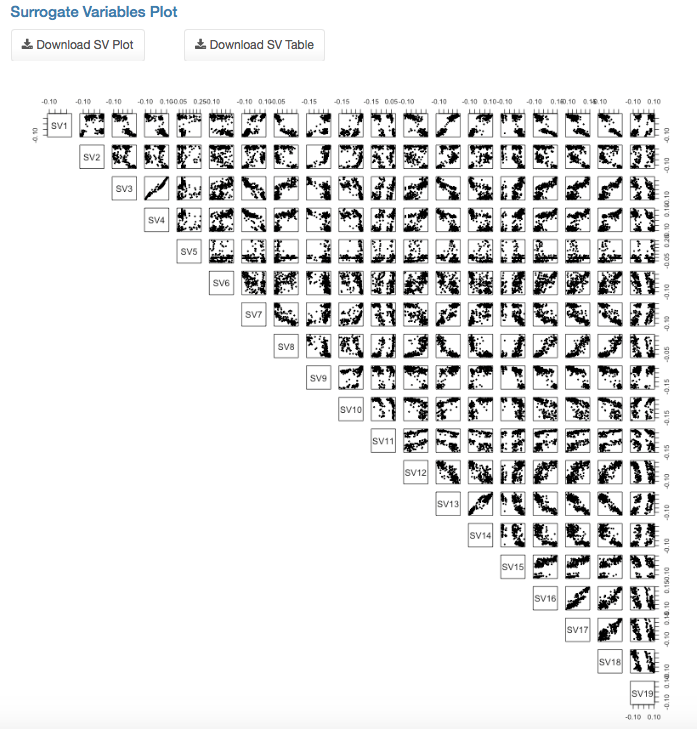
The output message will tell the user how many surrogate variables (in this case, 19) that were identified in the analysis. This

Now, navigate to the next tabbed panel, “Surrogate Variables”.

**VI.** **Surrogate Variables**

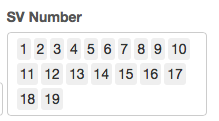
Upon clicking on this tab, by default, the user should see two plots:





The plot on the top, contains a correlation plot6 depicting the correlation between each SV and all Known Factors provided in the analysis. The plot on the bottom right shows pairwise scatter plots of each surrogate variable identified (19 in this case).

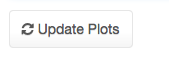
In this example, it is difficult to visualize all 19 surrogate variables effectively. So let’s just focus on the first five SVs. To do this, the user needs to go to the “SV Number” box shown below:



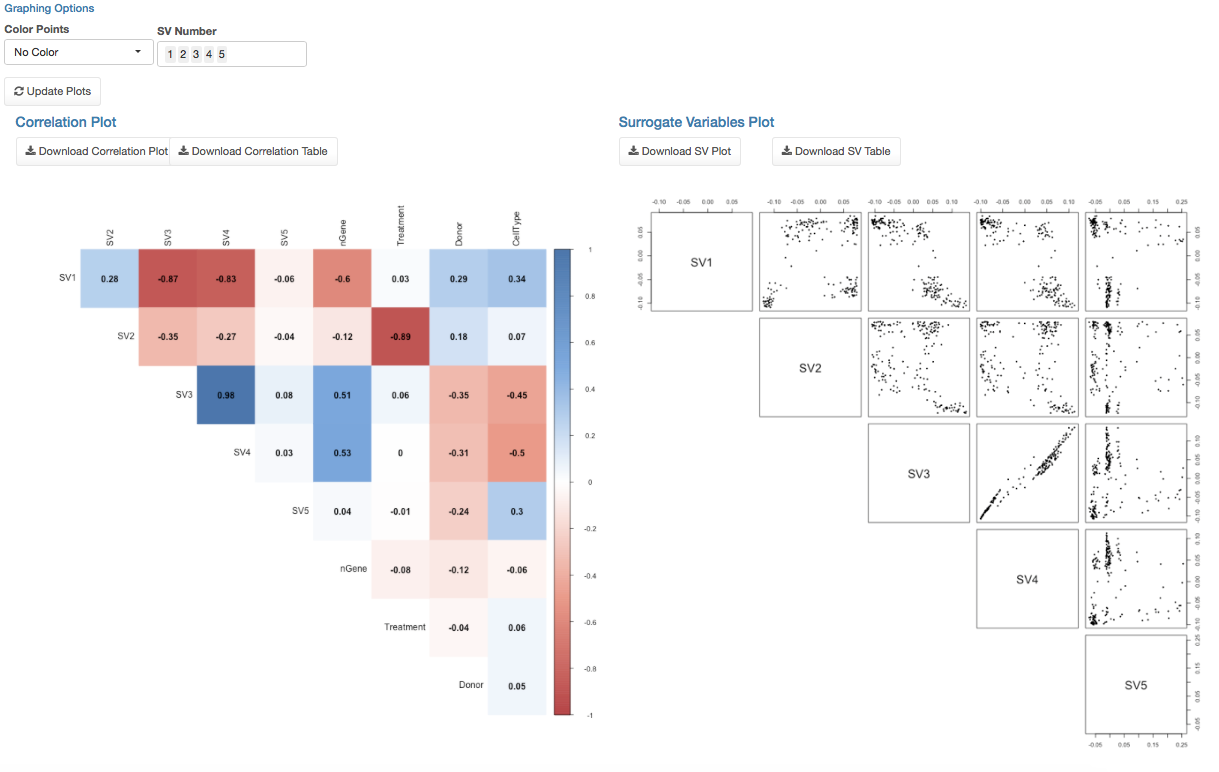
And merely remove all numbers except 1 through 5:



And then click the “Update Plots” button:



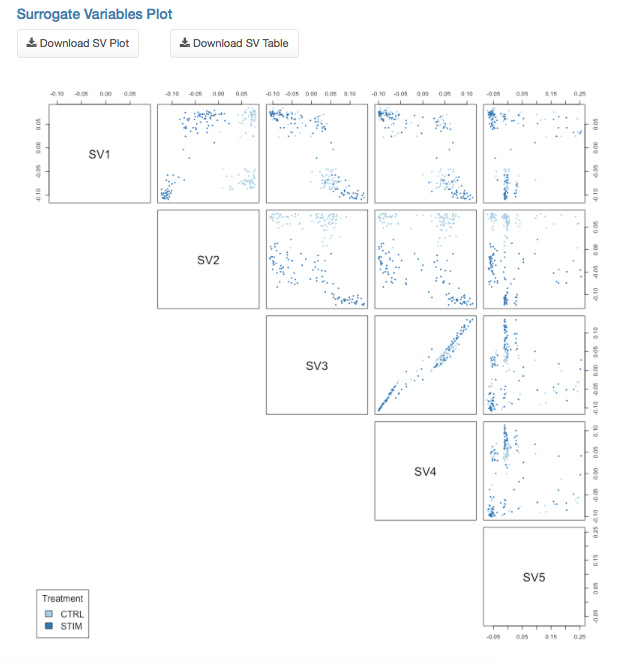
Which will re-display the two plots, this time only visualizing SVs 1 through 5. The updated plots in this example will look like this:



Focusing on the bottom right pairwise scatterplots, the user also has the choice to color cells/samples in this figure by different known factors. Users should specify a choice from the “Color Points” drop-down menu, and then click the update plots button.



In the example below, the cells/samples are colored by the “Treatment” variable provided in the metadata file:



After plotting this data, users may also download PDF file versions by clicking the buttons:



Where the “Download Correlation Plot” corresponds to the circular plot in the bottom left of the page and the “Download SV Plot” corresponds to the pairwise scatter plot in the bottom right of the page.

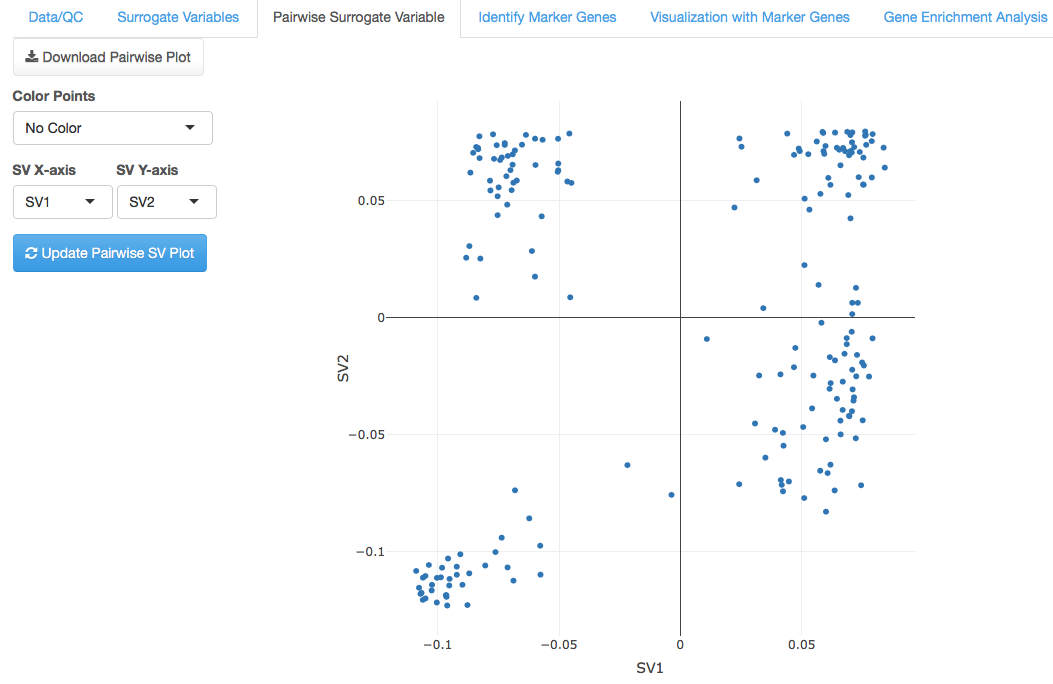
Users may also download CSV files containing the Pearson correlation values of each surrogate variable and known factor (Download Correlation Table button) and surrogate variable coordinates for each individual sample (Download SV Table button) respectively:



Next, navigate to the “Pairwise Surrogate Variable” tabbed panel.

**VII.** **Pairwise Surrogate Variable**

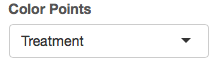
Upon clicking this tab, the user should see the following interactive7 plot:



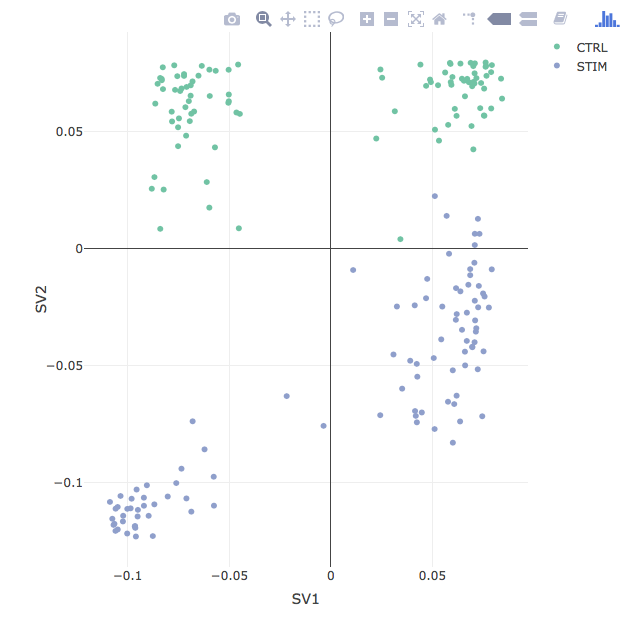
Within this tab, users may visualize two surrogate variables in a 2-D scatterplot. Users may specify which SVs to view using the drop down menus:



Users may also determine how to color the points (samples/cells) in the graph based on the known factors provided. Choose a selection in the “Color Points” drop down menu:



And then click the “Update Pairwise SV Plot” button will produce an updated plot:



For each interactive plot produced, users may hover the “mouse” over each point on the graph to see the Surrogate Variable coordinates as well as the cell/sample name information.

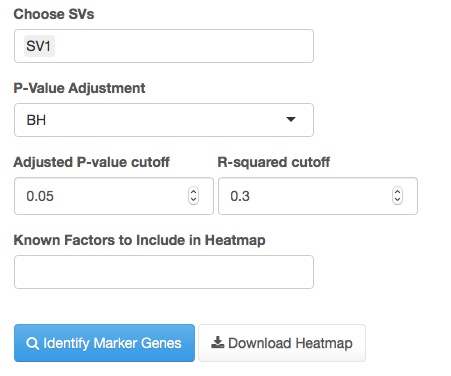
An HTML version of the plot may be downloaded by clicking the following button:



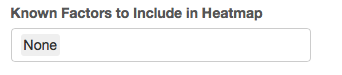
Next, navigate to the “Identify Marker Genes” tabbed panel.

**VIII.** **Identify Marker Genes**

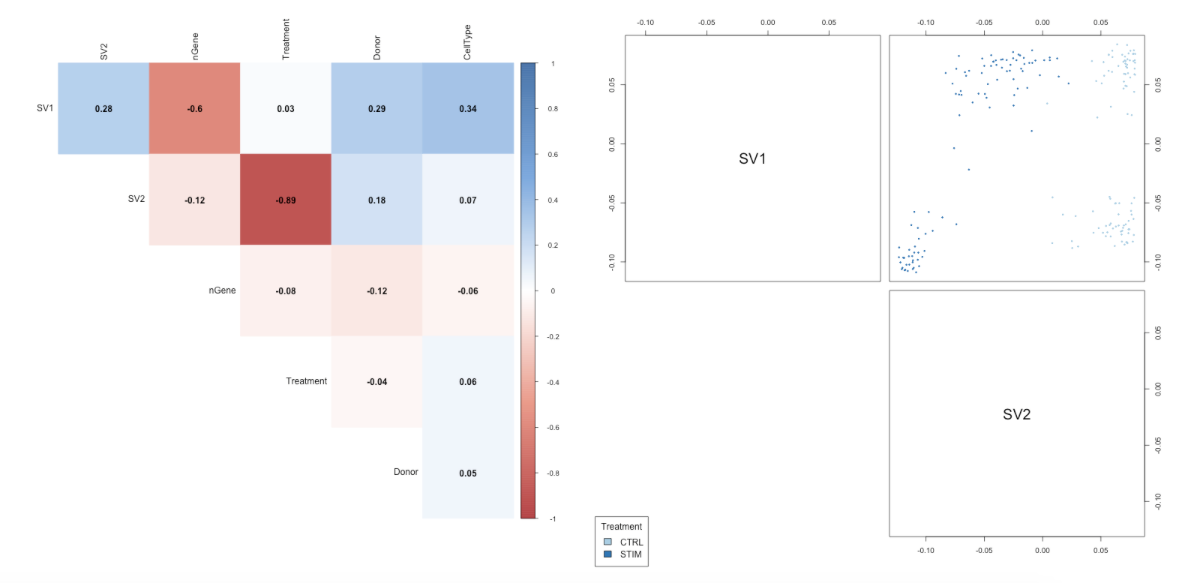
The next panel is where users can identify marker genes highly correlated with a Surrogate Variable. There are several parameters that the user must specify:



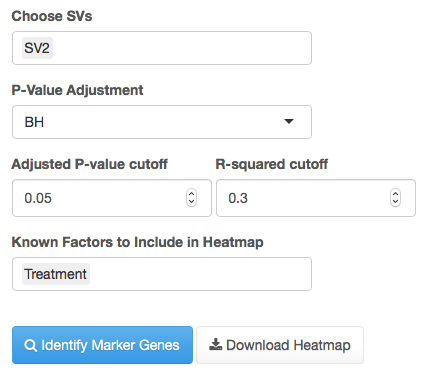
1. First, the user must indicate which surrogate variables in “Choose SVs” they wish to find marker genes for. Here, users may choose multiple or single surrogate variables.
2. Second, the user must specify which multiple hypothesis testing correction method should be applied. By default, BH (Benjamini – Hochberg procedure) is chosen, but users may also choose “bonferroni” or “none”.
3. Third, the users must provide an adjusted p-value cutoff (default is 0.05) between 0 and 1.
4. Fourth, the user must provide an R-squared cutoff (default 0.3) between 0 and 1 which is used to identify genes correlated with each SV of interest.
5. Lastly, the user must specify which known factor information should be included in the final visualization of marker genes. If no information is desired to be included, select the “None” option only:



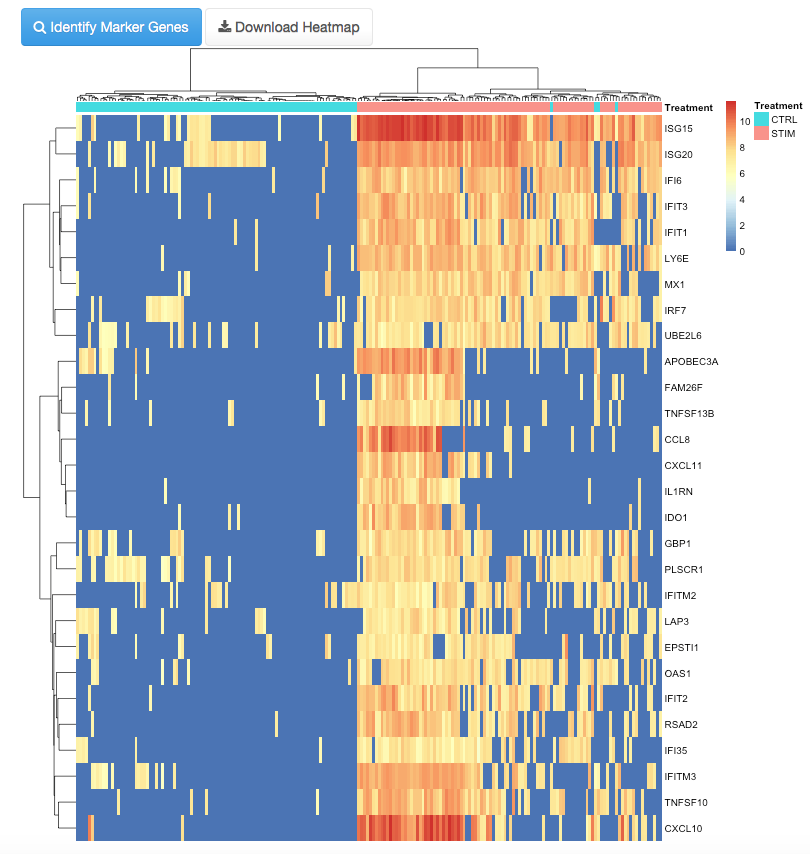
In the previous example, I notice that the SV2 surrogate variable is highly correlated with the “Treatment” known factor. In this data from Kang et. al 2017, human PBMCs were treated with interferon beta, an immune cell activator. Cells treated with interferon beta were labeled as “STIM” and those without treatment were labeled as “CTRL”. When we visualize the SV1 and SV2 projections for each single cell and overlay the interferon beta treatment information we see a separation of cells based on SV2 (interferon beta treatment):



As a result, I decide to find marker genes associated with SV2 using the following options and clicking the “Identify Marker Genes” button:



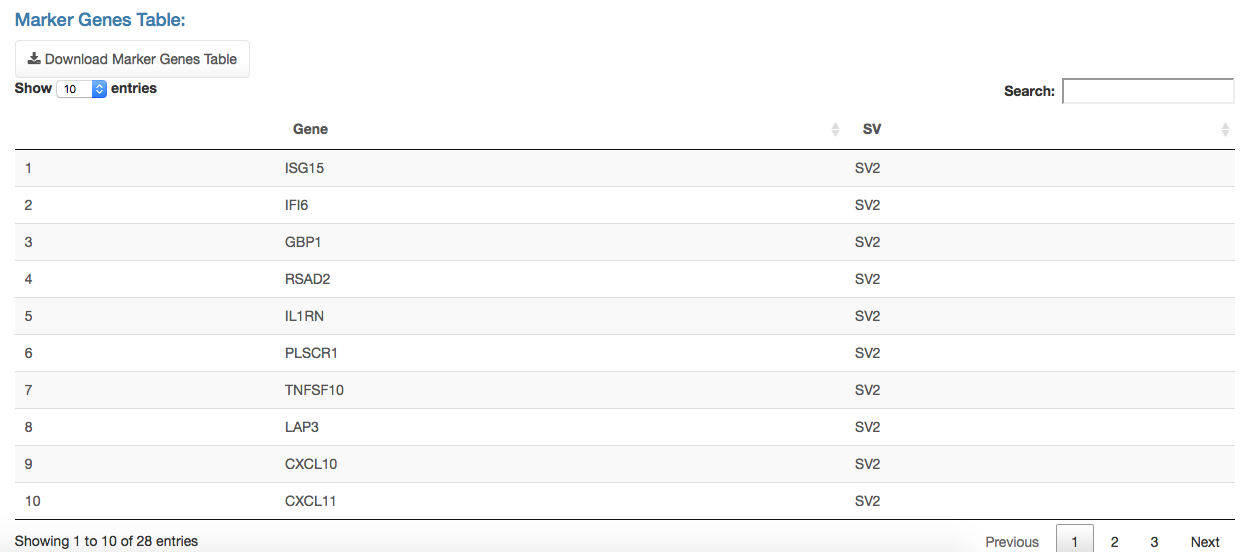
The resulting heatmap8 image is produced displaying the expression levels of each marker gene for each sample:



The resulting heatmap image may be downloaded in PDF format by clicking the button:



Below this above heatmap is an interactive table of the marker genes identified. The table will look like this:



Where each entry of the table will contain a gene name and the SV they are associated with. If multiple SVs are provided, a gene may be associated with one or multiple SVs.

At the top of the table, users may specify how many entries to view (default: 10) or search for a particular gene of interest to filter the table results:



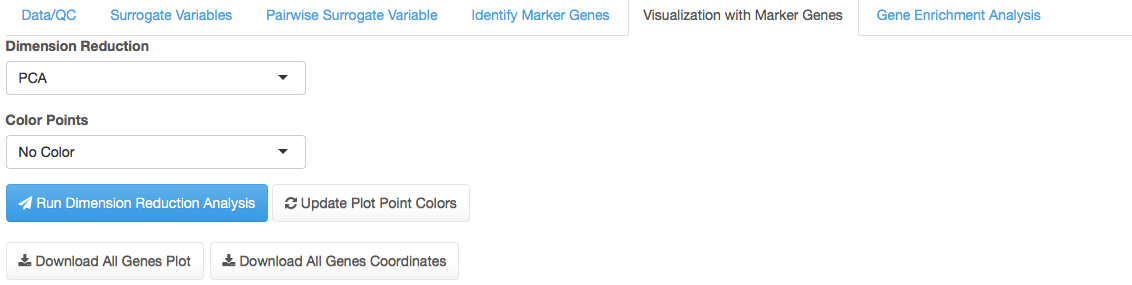
Lastly, the table may be downloaded in comma-separated value (CSV) format by clicking the following button:



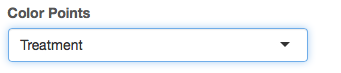
Next, navigate to the “Visualization with Marker Genes” tabbed panel.

**IX.** **Dimension Reduction and Visualization**

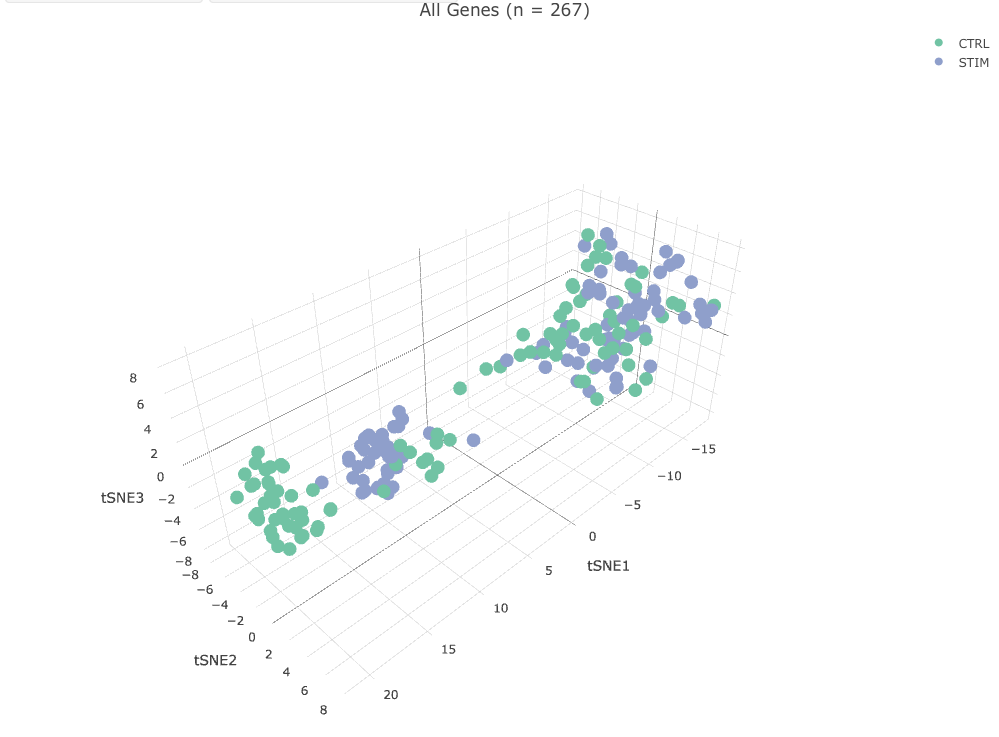
In the second to last tabbed panel, users can perform dimension reduction analyses (PCA = principal component analysis, or t-SNE9 = t-distributed stochastic neighbor embedding) on their datasets to better visualize the complex structure/biology within their datasets:



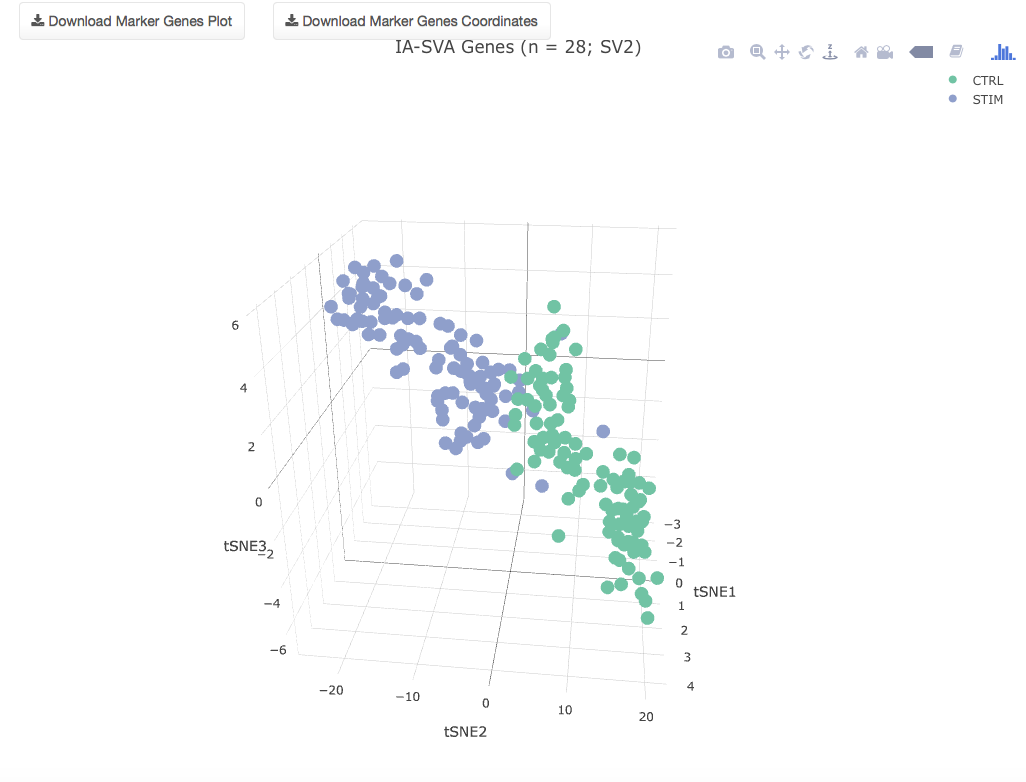
After selecting either “PCA” or “t-SNE”, users should then specify how the plots should be colored in the “Color Points” menu. Here, the known factors provided in the metadata file in the beginning stages should be displayed. In this example, we will color points/samples by their “Treatment” metadata variable:



Next, we run the dimension reduction analysis (this case, we chose t-SNE) and two 3-dimensional interactive7 plots are produced:



Where the top plot represents the final t-SNE dimensions determined based on the original preprocessed data (all genes after filtering and normalization) and the bottom plot:



represents the final t-SNE dimensions determined based on the marker genes identified via IA-SVA (28 genes associated with SV2 in this case).

In both plots, users may hover their mouse over graph points to obtain information about the t-SNE dimensions and cell/sample name. Additionally, users may click and drag on the graph space to rotate the plot. Lastly, users may use their mouse wheel to zoom in and out of the plot.

HTML versions of each plot may be downloaded using the following buttons:



Similarly, CSV tables of the dimension reduction coordinates for each plot may be downloaded using the following buttons:

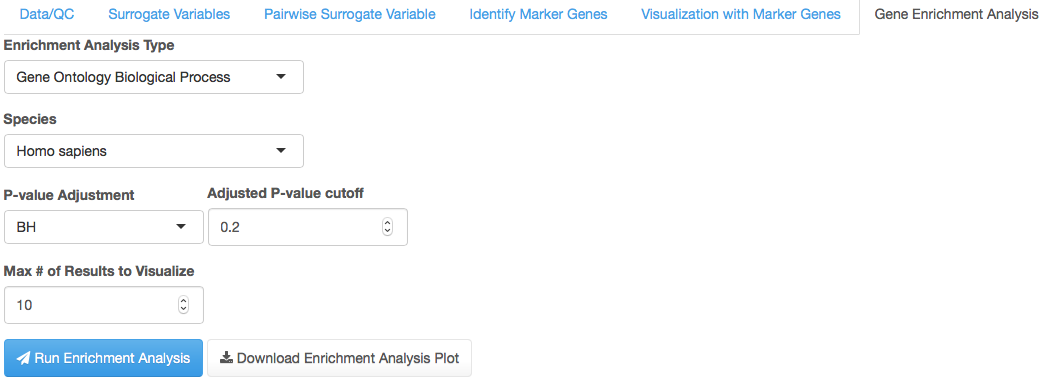
 

Next, navigate to the “Gene Enrichment Analysis” tabbed panel.

**X.** **Gene Enrichment Analysis**

In this tabbed panel, users can identify the Gene Ontology (GO) terms and pathways (KEGG, etc) that are associated with the marker genes identified in the previous tab “Identify Marker Genes”. Gene enrichment analyses are performed using the R package “clusterProfiler”10.

From the drop-down menu, users may specify the pathway/database (by default, Gene Ontology Biological Process is selected):



Other pathway/database options include:

1. Gene Ontology Cellular Component
2. Gene Ontology Molecular Function
3. Kyoto Encyclopedia of Genes and Genomes (KEGG)
4. Homo sapiens PBMC Cell Specific Modules
   1. This contains a set of human peripheral blood mononuclear cell (PBMC) specific gene lists as determined from public 10X Genomics single cell RNA-seq data.
5. Homo sapiens Immune Modules
   1. This contains blood transcriptional modules described by Chaussabel et al. (2008)11 and by Li et al. (2014)12 as well as metabolic profiling clusters from Weiner et al. (2012)13, obtained from the “tmod”14 R package: .

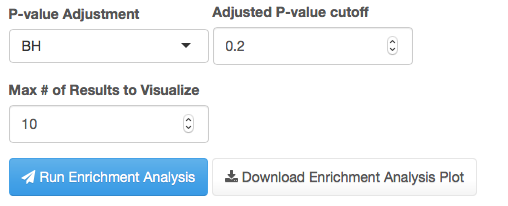
Next, the user should designate the “Species” type of their input data (by default: Homo sapiens is selected):



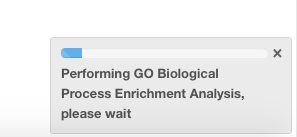
Currently, three species are supported:

1. Homo sapiens
2. Mus musculus
3. Rattus norvegicus

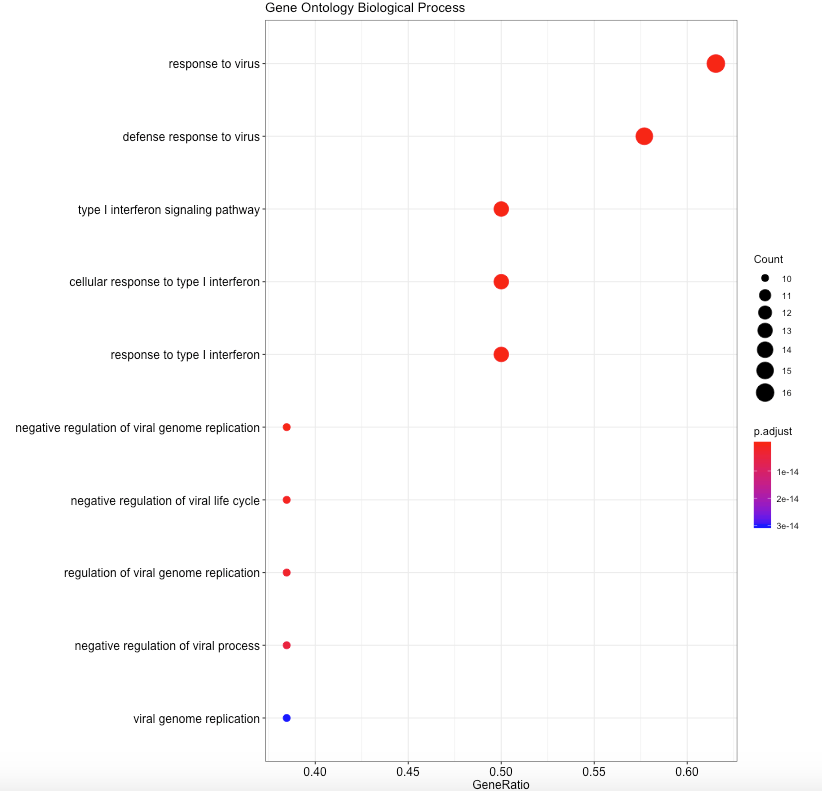
Lastly, before running the gene enrichment analysis, users should specify the multiple hypothesis testing correction method to use (either BH = benjamini and Hochberg procedure, bonferroni, or none), as well as the adjusted p-value cutoff (default set to 0.2), and the maximum number of results to view after the enrichment analysis finishes calculation (default set to 10 results).



Upon clicking the blue “Run Enrichment Analysis” button, users should see the message prompt in the bottom right of their screen:



After the analysis finishes, the users should see the following plot14 displaying the top 10 enriched pathways:

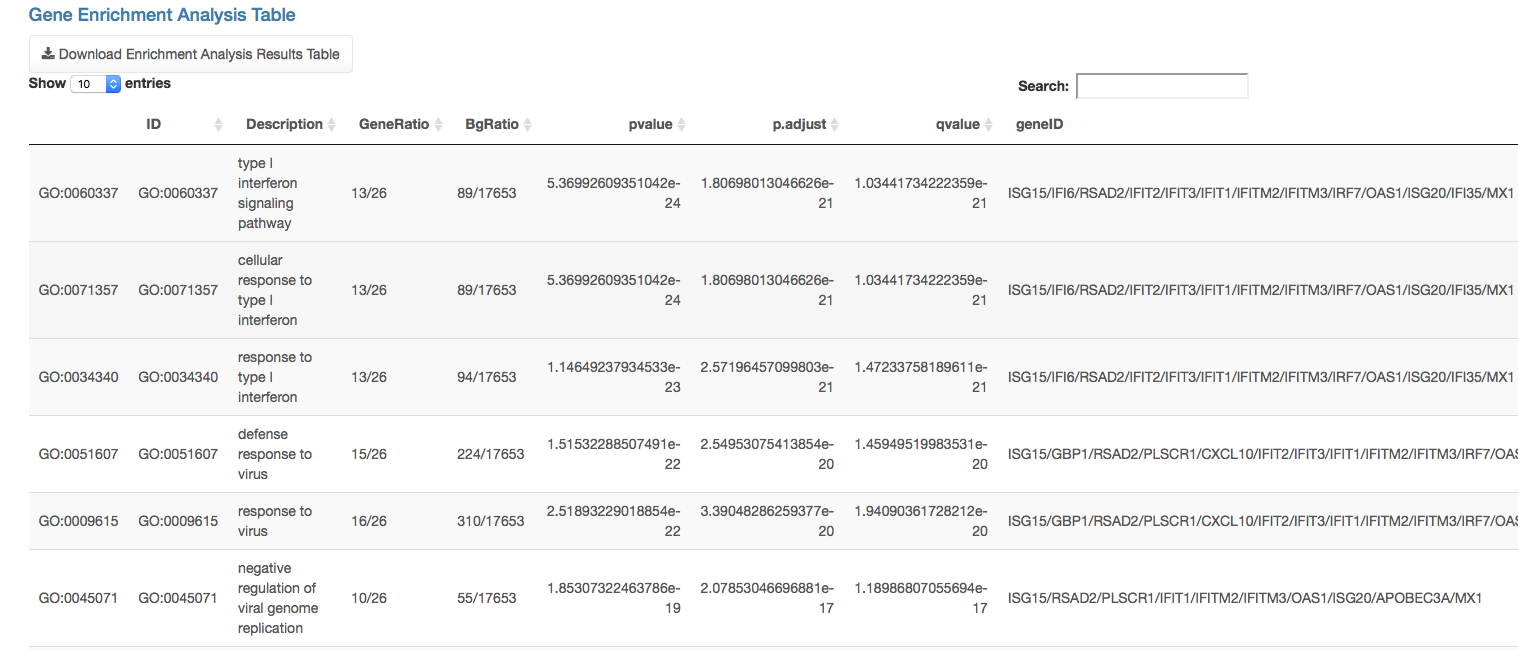


In the resultant plot, the y-axis displays the pathway/term name and the x-axis displays the “gene ratio” or the number of marker genes that fall within each pathway/term divided by the total number of genes in that pathway/term. The sizes of the points also reflect the “Count” or number of marker genes that fall within each pathway/term. Points are colored by their adjusted p-value.

The resultant plot may be downloaded by clicking the button:



Below this plot will be an interactive table of the pathways/terms identified in the enrichment analysis. The table will look like this:



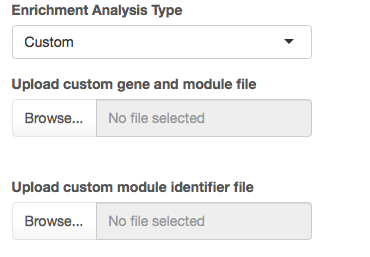
Where each entry of the table will contain a pathway term (ID), description, GeneRatio (number of marker genes in the pathway divided by the total genes in the pathway), p-value, adjusted p-value, q-value, and geneID (the marker genes that fell within the pathway).

The table may be downloaded in comma-separated value (CSV) format by clicking the following button:



**Enrichment Analysis using Custom Pathway/Module Terms**

Users also have the option to provide their own/custom pathway/module terms for gene enrichment analysis. To do this, users must first select “Custom” from the “Enrichment Analysis Type” drop down menu. The user interface will update to the following:



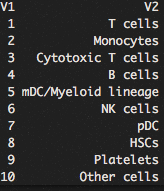
Afterwards, the user must provide their custom data in the form of two tab-delimited text files:

1. A custom gene and module file: a 2 column file with the first column corresponding to a pathway/module identifier (e.g., 1, 2, 3, etc) and the second column corresponding to a single gene symbol. An example is as follows:



Where each of the gene symbols in column two (V2) are a part of the pathway/module (currently named 1, in column V1).

1. A module identifier file: a 2 column file with the first column corresponding to a pathway/module identifier (e.g., 1, 2, 3, etc). Note, these identifiers should be the same as those found in column 1 of the gene and module file. The second column of this file will contain the pathway/module description/name. An example is as follows:



Where pathway/module identifier number 1 corresponds to a module named “T cells”.

**XI.** **References**

1. Chang, W. *et al.* *shiny: Web Application Framework for R*. (2018).

2. Kang, H. M. *et al.* Multiplexed droplet single-cell RNA-sequencing using natural genetic variation. *Nat. Biotechnol.* **36,** 89–94 (2018).

3. Robinson, M. D., McCarthy, D. J. & Smyth, G. K. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinforma. Oxf. Engl.* **26,** 139–140 (2010).

4. preprocessCore. *Bioconductor* Available at: http://bioconductor.org/packages/preprocessCore/. (Accessed: 17th September 2018)

5. L. Lun, A. T., Bach, K. & Marioni, J. C. Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome Biol.* **17,** 1–14 (2016).

6. Wei, T. *et al.* *corrplot: Visualization of a Correlation Matrix*. (2017).

7. Sievert, C. *et al.* *plotly: Create Interactive Web Graphics via ‘plotly.js’*. (2018).

8. Kolde, R. *pheatmap: Pretty Heatmaps*. (2018).

9. Maaten, L. van der. Accelerating t-SNE using Tree-Based Algorithms. *J. Mach. Learn. Res.* **15,** 3221–3245 (2014).

10. Yu, G., Wang, L.-G., Han, Y. & He, Q.-Y. clusterProfiler: an R Package for Comparing Biological Themes Among Gene Clusters. *OMICS J. Integr. Biol.* **16,** 284–287 (2012).

11. Chaussabel, D. *et al.* A modular analysis framework for blood genomics studies: application to systemic lupus erythematosus. *Immunity* **29,** 150–164 (2008).

12. Li, S. *et al.* Molecular signatures of antibody responses derived from a systems biological study of 5 human vaccines. *Nat. Immunol.* **15,** 195–204 (2014).

13. Weiner 3rd, J. *et al.* Biomarkers of Inflammation, Immunosuppression and Stress Are Revealed by Metabolomic Profiling of Tuberculosis Patients. *PLOS ONE* **7,** e40221 (2012).

14. Weiner, J. *tmod: Feature Set Enrichment Analysis for Metabolomics and Transcriptomics*. (2018).