Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Application Note

OXFORD

Genome Analysis

NASQAR: A web-based platform for High-throughput sequencing data analysis and visualization

Ayman Yousif [1], Nizar Drou [1], Jillian Rowe [1], Mohammed Khalfan [2], and Kristin C. Gunsalus [1,2,*]

[1]NYU Abu Dhabi Center for Genomics & Systems Biology, Division of Biological Sciences, Abu Dhabi, United Arab Emirates and [2]Center for Genomics & Systems Biology, Department of Biology, New York University, New York, 10003, United States.

*To whom correspondence should be addressed.

Abstract

Summary: As high-throughput sequencing applications continue to evolve, the rapid growth in quantity and variety of sequence-based data calls for the development of new software libraries and tools for data analysis and visualization. Often, effective use of these tools requires computational skills beyond those of many researchers. To lower this computational barrier, we have created a dynamic web-based platform, NASQAR (Nucleic Acid SeQuence Analysis Resource). NASQAR provides an intuitive interface that allows users to easily and efficiently explore their data in an interactive way using popular tools for a variety of applications, including Transcriptome Data Preprocessing, RNAseq Analysis (including Single-cell RNAseq), Metagenomics, and Gene Enrichment.

Availability and Implementation: NASQAR offers a collection of custom and publicly available open-source web applications that make extensive use of the Shiny R package to provide interactive data visualization. The platform is publicly accessible at `http://nasqar.abudhabi.nyu.edu/`. NASQAR is open-source and the code is available through GitHub at `https://github.com/nasqar/NASQAR`. NASQAR is also available as a Docker image at `https://hub.docker.com/r/aymanm/nasqarall`. NASQAR is a collaboration between the core bioinformatics teams of the NYU Abu Dhabi and NYU New York Centers for Genomics and Systems Biology.

Contact: kcg1@nyu.edu

1 Introduction

Genomic data has experienced tremendous growth in recent years due to the rapid advancement of Next Generation Sequencing (NGS) technologies (Goodwin *et al.* 2016; `https://www.genome.gov/sequencingcostsdata/`). Common applications include transcriptome profiling; de novo genome sequencing; metagenomics; and mapping of genomic variation, transcription factor binding sites, chromatin modifications, chromatin accessibility, and 3D chromatin conformation. Single-cell versions of these (e.g. Zheng *et al.* 2019) and newer methods — such as spatial transcriptomics (e.g. Ståhl *et al.* 2016), CRISPR-based screens (e.g. Canver *et al.* 2018), and multi-modal profiling (simultaneous quantification of proteins and mRNAs, e.g. Stoeckius *et al.* 2017) — are rapidly evolving, resulting in continuous churn in the field as new techniques and applications come on the scene (e.g. Stuart and Satija 2019; Mimitou *et al.* 2019). As the volume of data and diversity of applications continue to grow, so does the number of software libraries and tools for the analysis and visualization of these datasets. Many of the available tools for genomic data analysis require computational experience and lack a graphical user interface (GUI), making them inaccessible to many researchers whose work depends on them. Some of the common challenges include:

- Knowledge and experience in various programming/scripting languages (R, Python, Shell, etc.)
- Data munging: pre-processing and reformatting for use with specific tools
- Limited computational resources (cpu, memory, and disk storage)

- Installation of software packages and dependencies. Many required tasks can be time consuming and tedious due to issues such as satisfying software or hardware requirements and resolving software dependencies. Moreover, the rapid churn of operating system updates and hardware configurations contributes to the decline of a tool's usability and lifetime (Mangul *et al.* 2019).

To address these challenges, we have implemented NASQAR: Nucleic Acid SeQuence Analysis Resource, a web-based platform that that wraps popular high-level analysis and visualization tools in an intuitive and appealing interface. In addition to providing the ability to explore data using various tools and visualization methods, NASQAR also enables users to download analysis results such as normalized count data and a wide array of figures (PCA plots, heatmaps, dendograms, etc.). NASQAR thus lowers the entry barrier and provides greater independence for researchers with little or no computational experience to carry out standard bioinformatics analysis, visualize their data, and produce publication-ready data files and figures.

2 NASQAR Platform and Implementation

The architectural framework of the NASQAR web platform is illustrated in Figure 1. NASQAR has been deployed on a cluster of virtual machines and is publicly accessible at `http://nasqar.abudhabi.nyu.edu/`. Docker (Merkel 2014) and Swarm (Soppelsa and Kaewkasi 2017) provide containerization and cluster management, and the Traefik reverse proxy / load balancer (`https://traefik.io/`) manages requests and maintains sticky user sessions, which is essential for hosting Shiny applications. This framework allows access to multiple users concurrently while providing sufficient resources (RAM/CPU) for the applications. In anticipation of growing computational demand and the addition of more applications, the scalable design makes it relatively easy to increase dedicated resources simply by adding more nodes to the cluster.

A Docker image of NASQAR is publicly available through DockerHub and can be used to deploy the application seamlessly on any system with Docker installed, whether a local computer or a public server. In addition, each application can be installed and launched on its own, saving users from the hassle of satisfying the different software and hardware requirements. The source code is available publicly on GitHub and is actively maintained. All applications have clear user guides with example data sets to help users get started and acclimate quickly.

NASQAR comprises a collection of applications primarily implemented in R, a widely used and freely available statistical programming language (R Core Team 2017). Most of the analysis workflows are built using R libraries for genomics and computation. The front-end design utilizes the R Shiny (Chang *et al.* 2018) library and is supported by JavaScript/CSS to enhance usability and improve overall user experience.

In addition to previously published software, we introduce here several new applications we have developed that wrap around popular analysis packages, such as DESeq2 (Love *et al.* 2014) and Seurat (Butler *et al.* 2018) for bulk and single-cell RNA-seq analysis and visualization, respectively. Since most of the analysis applications in NASQAR require a matrix of gene counts as input, we have also built a convenient tool to assist with preprocessing, GeneCountMerger. Some of the applications have been integrated to provide a seamless transition from data preprocessing to downstream analysis. This implementation gives users the option of using multiple analysis applications without having to modify/reformat the input data set, thus allowing them to easily benchmark and compare the performance of different analysis software packages.

3 Results

The following is a description of each application currently hosted by NASQAR:

3.1 GeneCountMerger

This preprocessing tool is used to merge individual raw gene count files produced from software such as htseq-count (Pyl *et al.* 2014) or featurecounts (Smyth *et al.* 2013). Options include:

- Merge individual sample count files into one matrix
- Merge multiple raw count matrices
- Convert Ensembl gene IDs to gene names
- Select from available genomes / versions
- Add pseudocounts
- Rename sample column headers
- Download merged counts file in .csv format
- Seamless transcriptome analysis following count merger (Seurat Wizard for single-cell RNA analysis; DESeq2Shiny or START (Sklenar *et al.* 2016) for bulk RNA analysis)

3.2 Seurat Wizards

Seurat Wizards are wizard-style web-based interactive applications to perform guided single-cell RNA-seq data analysis and visualization. They are based on Seurat, a popular R package designed for QC, analysis, and exploration of single-cell RNAseq data. The wizard style makes it intuitive to go back and forth between steps and adjust parameters based on the results/feedback of different outputs/plots/steps, giving the user the ability to interactively tune the analysis. SeuratWizard and SeuratV3Wizard implementations provide support for Seurat versions 2 and 3 (Stuart *et al.* 2019), respectively.

Another web-based tool for scRNA-seq analysis, IS-CellR(Patel 2018), has recently been described that also utilizes Seurat v2. The SeuratWizard and SeuratV3Wizard take a different approach to design and implementation and follow closely the Seurat Guided Clustering Tutorials devised by the authors (`https://satijalab.org/seurat/v3.0/pbmc3k_tutorial.html`). Users can follow the tutorials while using the Wizards and can edit parameters at almost every step, which is instrumental in producing accurate results. A unique feature of the Seurat Wizards is that they can accept as input processed 10X Genomics data files in place of a matrix of gene counts, which eliminates the need for this additional pre-processing step. SeuratV3Wizard integrates several additional features like the UCSC Cell Browser (`https://github.com/maximilianh/cellBrowser`), enabling users to interactively visualize clusters and gene markers, and the newly published sctransform method (Hafemeister and Satija 2019), which gives users the ability to run the analysis using two slightly different workflows and compare the results. These differences in features and design give the Seurat Wizards more versatility and improve usability in comparison with other publicly available implementations of Seurat.

3.3 Deseq2Shiny

The Deseq2Shiny app is a Shiny wrapper around DESeq2 (Love *et al.* 2014), a popular R package for performing differential mRNA expression analysis of RNA-seq data. This web-based application implements the standard default workflow outlined in the DESeq2 Bioconductor tutorial (`https://bioconductor.org/packages/devel/bioc/vignettes/DESeq2/inst/doc/DESeq2.html`). This includes normalization, data transformation (e.g., rlog and vst for clustering), and estimation for dispersion and log fold change. This

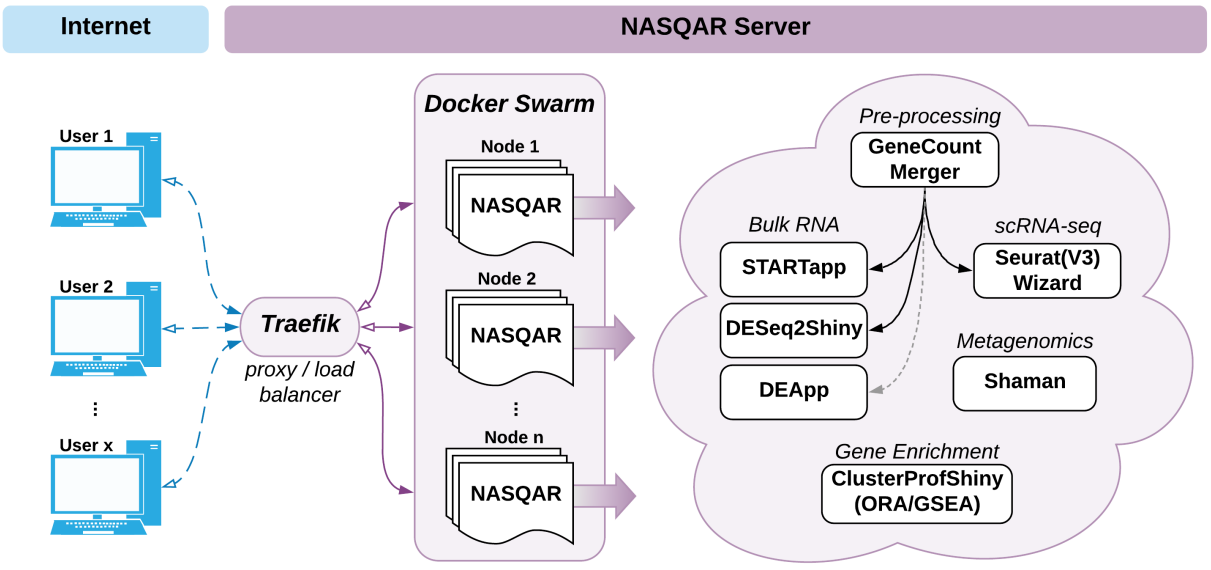


Fig. 1. NASQAR Platform Architecture. A cluster of virtual machines at NYU Abu Dhabi serves NASQAR applications to multiple concurrent users. Applications are containerized and managed on the cluster using Docker and Swarm, while Traefik load-balances requests among available server nodes. Functionality includes merging gene counts, conversion of gene IDs to gene names, analysis of differential mRNA expression, metagenomics analysis, and gene set and functional enrichment analysis. Applications for bulk expression analysis include DESeq2, limma, and EdgeR. Single-cell RNAseq analysis with Seurat Wizards is built on top of the Seurat R package and includes options for filtering, normalization, dimensionality reduction (PCA), clustering, and t-SNE. Enrichment analysis includes applications for Gene Set Enrichment Analysis (GSEA) and Over-representation Analysis (ORA) built using the clusterProfiler R package.

app follows the same implementation as other apps on NASQAR, whereby users can fine tune the analysis parameters interactively.

3.4 ClusterProfilerShiny

The ClusterProfilerShiny apps wrap the popular clusterProfiler (Yu *et al.* 2012) package, which implements methods to analyze and visualize functional profiles of genomic coordinates, genes, and gene clusters. Users can upload their own data from DESeq2 or import data from the upstream Deseq2Shiny app. These apps allow for quick and easy over-representation analysis (ORA) and gene set enrichment analysis (GSEA) of GO terms and KEGG pathways. Visuals produced include dot plots, word clouds, category net plots, enrichment map plots, GO induced graphs, GSEA plots, and enriched KEGG pathway plots using the Pathview package (Luo and Brouwer 2013).

3.5 Other open-source apps

- START: a web-based RNA-seq analysis and visualization resource (Sklenar *et al.* 2016). We have modified this application slightly from the open-source version to add options to some plots. We have also integrated it with GeneCountMerger so that once merging gene counts is complete, users can launch the START app and have their merged matrix data loaded automatically.
- DEApp (Li and Andrade 2017): an interactive web application for differential expression analysis.
- Shaman (Quereda *et al.* 2016): a Shiny application that enables the identification of differentially abundant genera within metagenomic datasets. It wraps around the Generalized Linear Model implemented in DESeq2. It includes multiple visualizations, and is compatible with common metagenomic file formats.

4 Conclusion

The NASQAR platform offers a publicly available, comprehensive suite of interactive bioinformatics analysis and visualization tools that is accessible to all researchers with or without computational experience. Our aim is to continue expanding this toolbox to include new analysis and visualization categories. NASQAR is actively developed and will continue to offer user support. Future work will also focus on deploying NASQAR on cloud computing platforms including Kubernetes on AWS.

Acknowledgements

The authors would like to acknowledge all the faculty and researchers in the NYU Abu Dhabi CGSB and Division of Biology for their excellent feedback, which has motivated the development of NASQAR. The authors would also like to acknowledge David Gresham and Siyu Sun (NYU New York CGSB) for their guidance during the development of the enrichment applications.

This research was carried out on the High Performance Computing resources at New York University Abu Dhabi. We extend special thanks to Fayizal Kunhi, NYU Abu Dhabi HPC.

Funding

This work was supported by a grant from the NYU Abu Dhabi Research Institute to the NYU Abu Dhabi Center for Genomics and Systems Biology (CGSB).

References

Butler, A., Hoffman, P., Smibert, P., Papalexi, E., and Satija, R. (2018). Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature Biotechnology*.

Canver, M. C., Haeussler, M., Bauer, D. E., Orkin, S. H., Sanjana, N. E., Shalem, O., Yuan, G.-C., Zhang, F., Concordet, J.-P., and Pinello, L. (2018). Integrated design, execution, and analysis of arrayed and pooled crispr genome-editing experiments. *Nature Protocols*, **13**, 946 EP –.

Chang, W., Cheng, J., Allaire, J., Xie, Y., and McPherson, J. (2018). *shiny: Web Application Framework for R*. R package version 1.1.0.

Goodwin, S., McPherson, J. D., and McCombie, W. R. (2016). Coming of age: ten years of next-generation sequencing technologies. *Nature Reviews Genetics*, **17**, 333 EP –. Review Article L3 -.

Hafemeister, C. and Satija, R. (2019). Normalization and variance stabilization of single-cell rna-seq data using regularized negative binomial regression. *bioRxiv*.

Li, Y. and Andrade, J. (2017). Deapp: an interactive web interface for differential expression analysis of next generation sequence data. *Source Code for Biology and Medicine*, **12**(1), 2.

Love, M. I., Huber, W., and Anders, S. (2014). Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome Biology*, **15**, 550.

Luo, W. and Brouwer, C. (2013). Pathview: an R/Bioconductor package for pathway-based data integration and visualization. *Bioinformatics*, **29**(14), 1830–1831.

Mangul, S., Martin, L. S., Eskin, E., and Blekhman, R. (2019). Improving the usability and archival stability of bioinformatics software. *Genome Biology*, **20**(1), 47.

Merkel, D. (2014). Docker: Lightweight linux containers for consistent development and deployment. *Linux J.*, **2014**(239).

Mimitou, E. P., Cheng, A., Montalbano, A., Hao, S., Stoeckius, M., Legut, M., Roush, T., Herrera, A., Papalexi, E., Ouyang, Z., Satija, R., Sanjana, N. E., Koralov, S. B., and Smibert, P. (2019). Multiplexed detection of proteins, transcriptomes, clonotypes and crispr perturbations in single cells. *Nature Methods*, **16**(5), 409–412.

Patel, M. V. (2018). iS-CellR: a user-friendly tool for analyzing and visualizing single-cell RNA sequencing data. *Bioinformatics*, **34**(24), 4305–4306.

Pyl, P. T., Anders, S., and Huber, W. (2014). HTSeq—a Python framework to work with high-throughput sequencing data. *Bioinformatics*, **31**(2), 166–169.

Quereda, J. J., Dussurget, O., Nahori, M.-A., Ghozlane, A., Volant, S., Dillies, M.-A., Regnault, B., Kennedy, S., Mondot, S., Villoing, B., Cossart, P., and Pizarro-Cerda, J. (2016). Bacteriocin from epidemic listeria strains alters the host intestinal microbiota to favor infection. *Proceedings of the National Academy of Sciences*, **113**(20), 5706–5711.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Sklenar, J., Nelson, J. W., Minnier, J., and Barnes, A. P. (2016). The START App: a web-based RNAseq analysis and visualization resource. *Bioinformatics*, **33**(3), 447–449.

Smyth, G. K., Shi, W., and Liao, Y. (2013). featureCounts: an efficient general purpose program for assigning sequence reads to genomic features. *Bioinformatics*, **30**(7), 923–930.

Soppelsa, F. and Kaewkasi, C. (2017). *Native Docker Clustering with Swarm*. Packt Publishing.

Ståhl, P. L., Salmén, F., Vickovic, S., Lundmark, A., Navarro, J. F., Magnusson, J., Giacomello, S., Asp, M., Westholm, J. O., Huss, M., Mollbrink, A., Linnarsson, S., Codeluppi, S., Borg, Å., Pontén, F., Costea, P. I., Sahlén, P., Mulder, J., Bergmann, O., Lundeberg, J., and Frisén, J. (2016). Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science*, **353**(6294), 78–82.

Stoeckius, M., Hafemeister, C., Stephenson, W., Houck-Loomis, B., Chattopadhyay, P. K., Swerdlow, H., Satija, R., and Smibert, P. (2017). Simultaneous epitope and transcriptome measurement in single cells. *Nature Methods*, **14**, 865 EP –.

Stuart, T. and Satija, R. (2019). Integrative single-cell analysis. *Nature Reviews Genetics*, **20**(5), 257–272.

Stuart, T., Butler, A., Hoffman, P., Hafemeister, C., Papalexi, E., Mauck, W. M. I., Hao, Y., Stoeckius, M., Smibert, P., and Satija, R. (2019). Comprehensive integration of single-cell data. *Cell*, **177**(7), 1888–1902.e21.

Yu, G., Wang, L.-G., Han, Y., and He, Q.-Y. (2012). clusterprofiler: an r package for comparing biological themes among gene clusters. *OMICS: A Journal of Integrative Biology*, **16**(5), 284–287. PMID: 22455463.

Zheng, M., Tian, S. Z., Capurso, D., Kim, M., Maurya, R., Lee, B., Piecuch, E., Gong, L., Zhu, J. J., Li, Z., Wong, C. H., Ngan, C. Y., Wang, P., Ruan, X., Wei, C.-L., and Ruan, Y. (2019). Multiplex chromatin interactions with single-molecule precision. *Nature*, **566**(7745), 558–562.

SUPPLEMENTARY MATERIALS FOR:

NASQAR: A web-based platform for High-throughput sequencing data analysis and visualization

Ayman Yousif[1], Nizar Drou[1], Jillian Rowe[1], Mohammed Khalfan[2] and Kristin C. Gunsalus[1,2,*]
[1]NYU Abu Dhabi Center for Genomics & System Biology, Division of Biological Sciences, Abu Dhabi, United Arab Emirates and
[2]Center for Genomics & System Biology, Department of Biology, New York University, New York, 10003, United States.
[*]To whom correspondence should be addressed.

1) Launch NASQAR using Docker (Recommended):

The recommended way to get NASQAR running is using Docker. The reason is that applications hosted within NASQAR have many package dependencies (R and OS) that might be tedious and very time consuming for the average user especially when trying to support different OS's (Windows/Linux/OSX).
Prerequisite: Make sure Docker (version >= 17.03.0-ce, https://docs.docker.com/install/) is installed.

Run NASQAR docker image as follows:

a) docker run -p 80:80 aymanm/nasqarall:nasqar (runs on port 80)
   If you are running this in your personal laptop/PC, access NASQAR using a modern web browser at the following URL http://localhost/
b) docker run -p 8083:80 aymanm/nasqarall:nasqar (runs on port 8083)
   If you are running this in your personal laptop/PC, access NASQAR using a modern web browser at the following URL http://localhost:8083/
c) If you are hosting this as a service at your organization, make sure the specified port is not blocked with a firewall so users can access the service. Users can access NASQAR using a modern web browser at the following URL http://<server_ip>:port/

Note: All apps in NASQAR can be launched individually. Visit each app's GitHub page for relevant instructions. For 3rd party apps hosted on NASQAR, please refer to their Github repositories.

App GitHub pages:

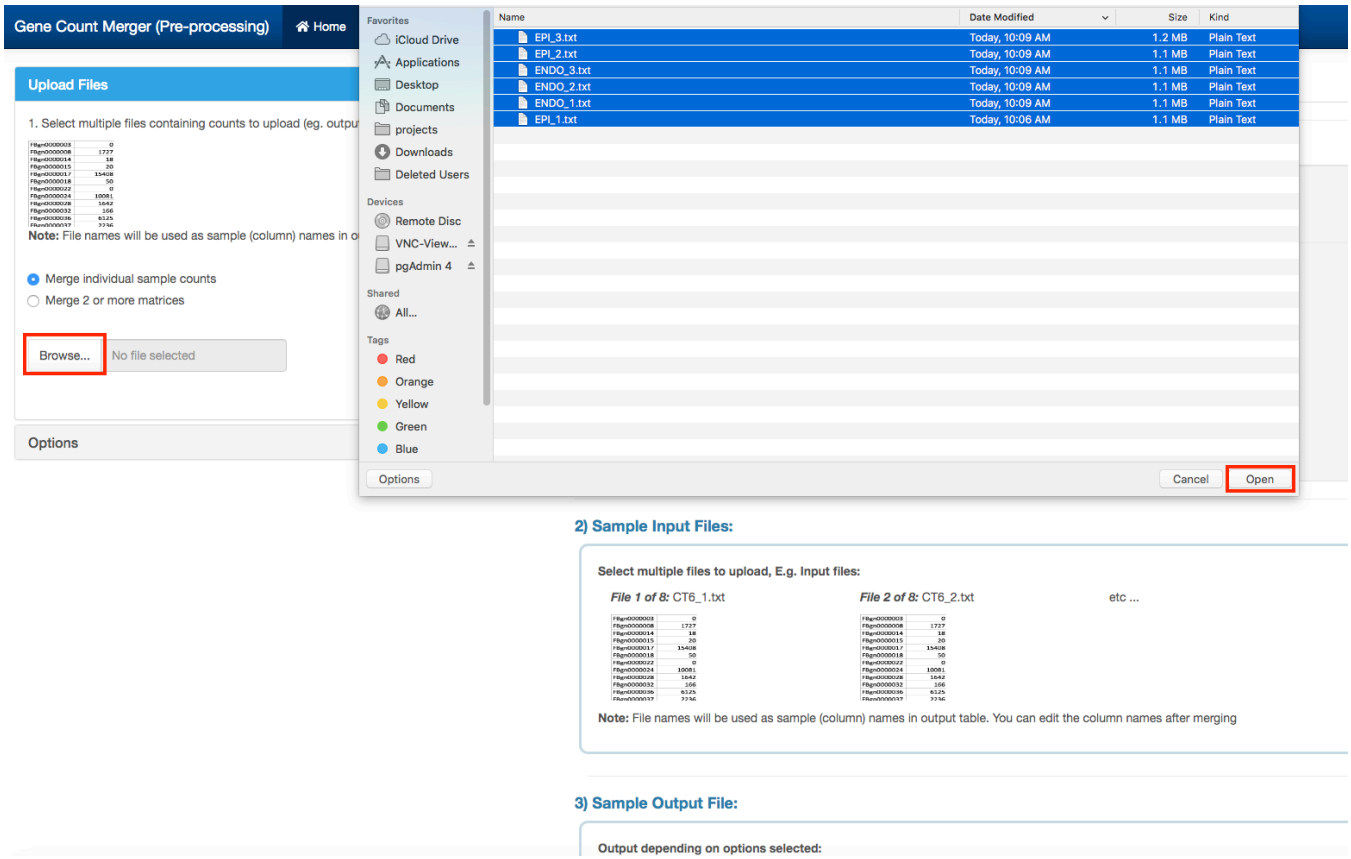
- SeuratV3Wizard (scRNA): https://github.com/nasqar/seuratv3wizard
- SeuratWizard (scRNA): https://github.com/nasqar/SeuratWizard
- deseq2shiny (Bulk RNA): https://github.com/nasqar/deseq2shiny
- GeneCountMerger (Pre-processing): https://github.com/nasqar/GeneCountMerger
- ClusterProfShinyGSEA (Enrichment): https://github.com/nasqar/ClusterProfShinyGSEA
- ClusterProfShinyORA (Enrichment): https://github.com/nasqar/ClusterProfShinyORA
- NASQAR (main page): https://github.com/nasqar/NASQAR

2) Example Use Case 1 (DGE and GSEA):

In this example we will show, using only your web browser, how you can start with individual sample gene count files (eg output of htseq counts) and carry out Differential Gene Expression (DGE) and Gene Set Enrichment analysis using DESeq2 and clusterProfiler R packages respectively.
The datasets provided in this example use case have been download from the ENSEMBL expression ATLAS (https://www.ebi.ac.uk/gxa/experiments/E-MTAB-970/Results). They belong to the ENSEMBL expression ATLAS experiment title "***Transcription profiling by high throughput sequencing of Sox17.Epi and Endo cells from mouse embryos***"[1]. The data comprises of 6 mouse RNA-seq samples, across two conditions. They were selected as an example dataset only, no other criteria were used in their selection.

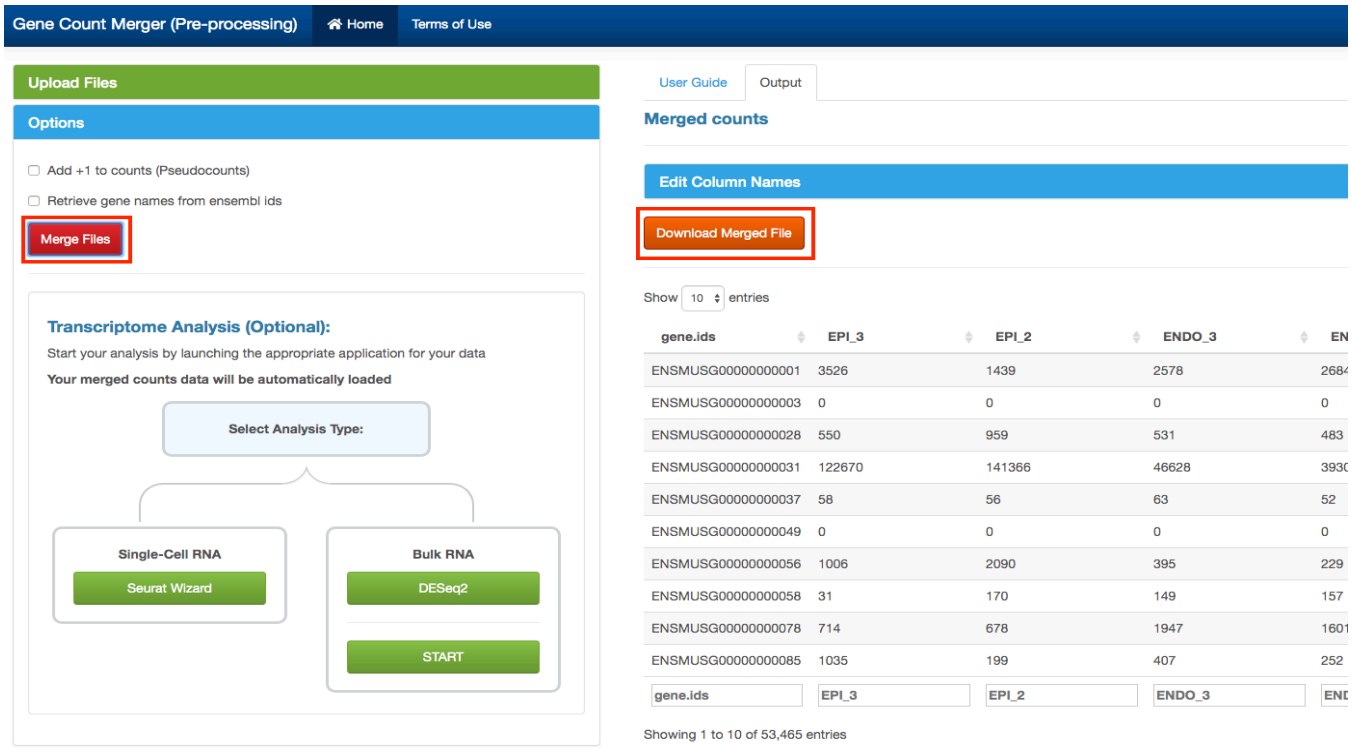
Step 1: Merge counts (GeneCountMerger):

a) Download example count files zip from here (https://drive.google.com/file/d/1OB2vojHqscLAHZWGETHdGK0yaZ9r06PT/iew?usp=sharing)
b) Extract zip file
c) Launch GeneCountMerger http://nasqar.abudhabi.nyu.edu/GeneCountMerger/
d) Click browse and select all gene count files:



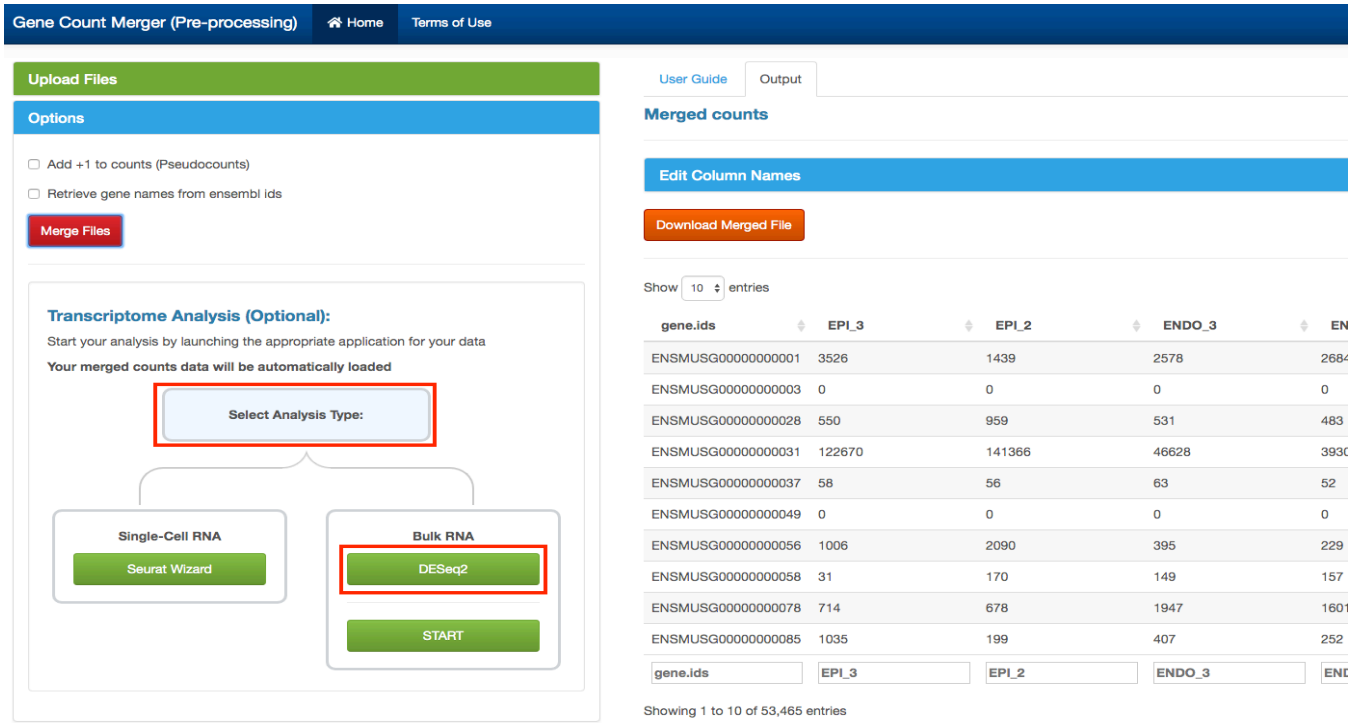
[1] Choi, E. , Kraus, M. R., Lemaire, L. A., Yoshimoto, M. , Vemula, S. , Potter, L. A., Manduchi, E. , Stoeckert, C. J., Grapin-Botton, A. and Magnuson, M. A. (2012), Dual Lineage-Specific Expression of Sox17 During Mouse Embryogenesis. STEM CELLS, 30: 2297-2308. doi:10.1002/stem.1192

e) Once loaded, click the red Merge button. Now all counts files have been consolidated into one table. You can now download and save it as a .csv file

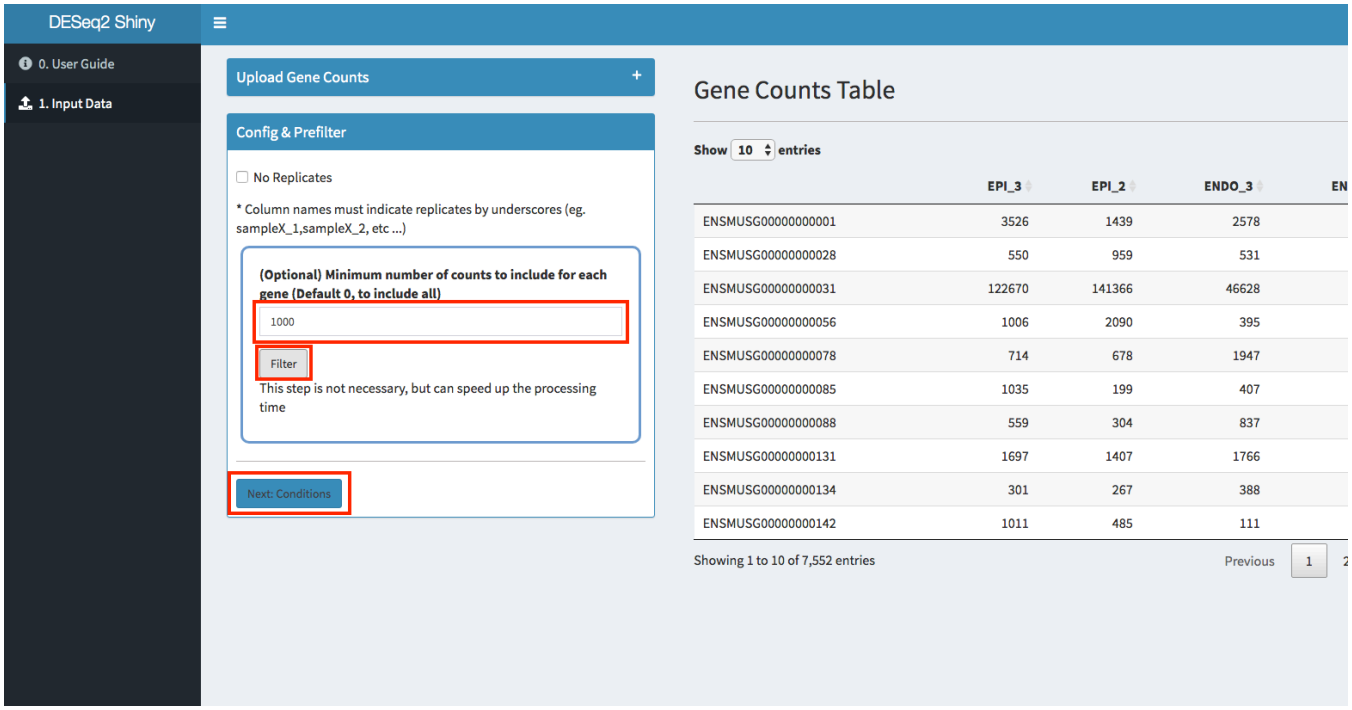


Step 2: Differential Gene Expression analysis (Deseq2Shiny):

a) Carrying on from the previous step, under "**Select Analysis Type**" select "**DESeq2**"

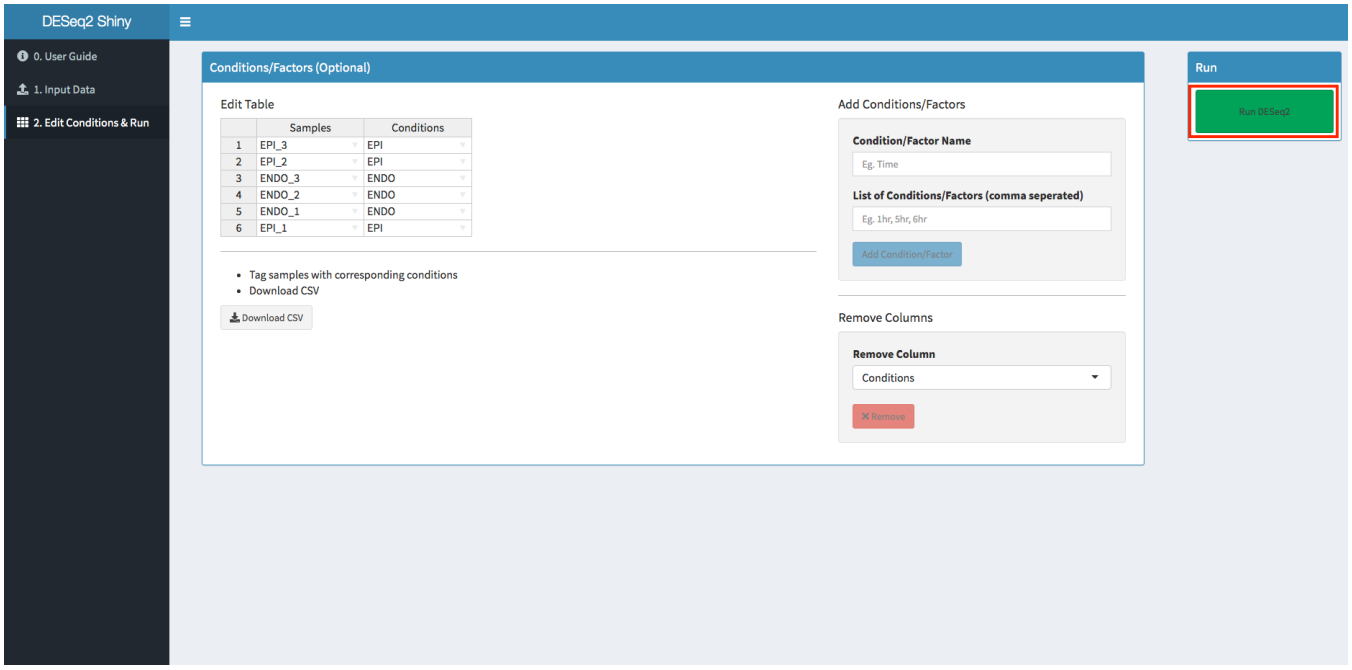


b) The Deseq2Shiny app will be launched in a new tab, with all counts data already loaded onto it. You can set the minimum number of counts to 1000 for example (this is mainly for speed up, you can continue with out it) and click "**Filter**". Then click "**Next: Conditions**"



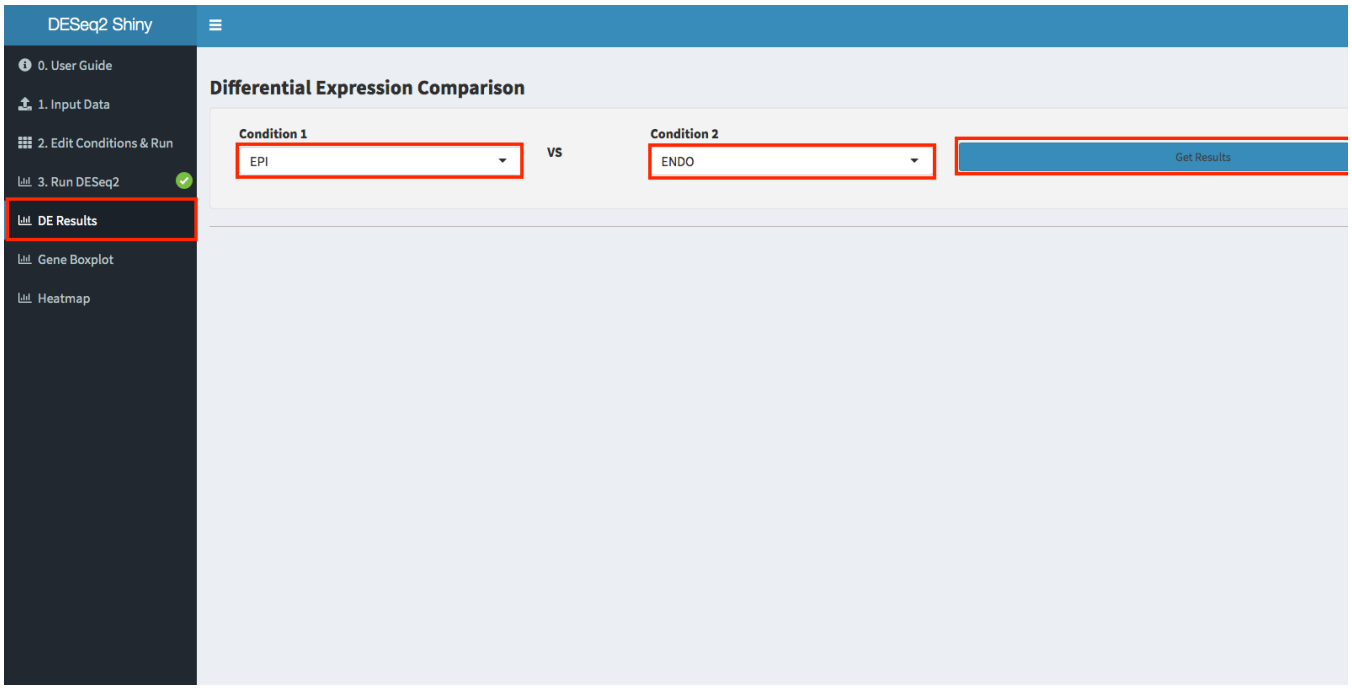
c) You can verify the sample/conditions table on the left, then click "**Run DESeq2**". The replicate/condition IDs are inferred from the naming of the raw counts files. For example, ENDO_1 will be automatically determined to mean that the sample belongs to the condition "ENDO", and that it is the first replicate. In case a user supplies filenames that are named

differently, this page will allow users to tag their replicates to the appropriate conditions.

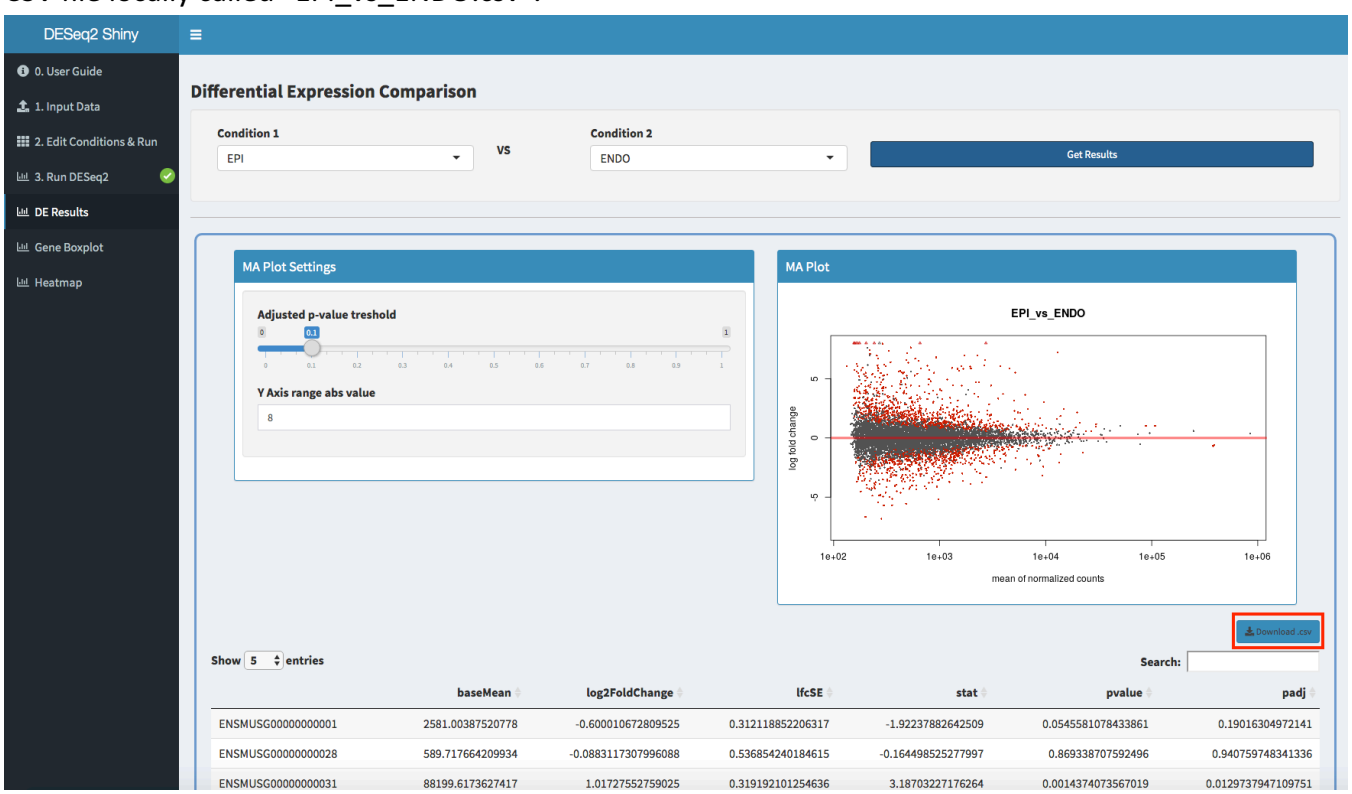


d) Once DESeq2 has completed, you will be able to see rlog/vst transformation matrices and PCA and distance heatmap plots.

e) Next, go to "**DE Results**" to be able to run comparisons between different sample conditions. Select two different conditions, for example "EPI" and "ENDO" and then click "**Get Results**"



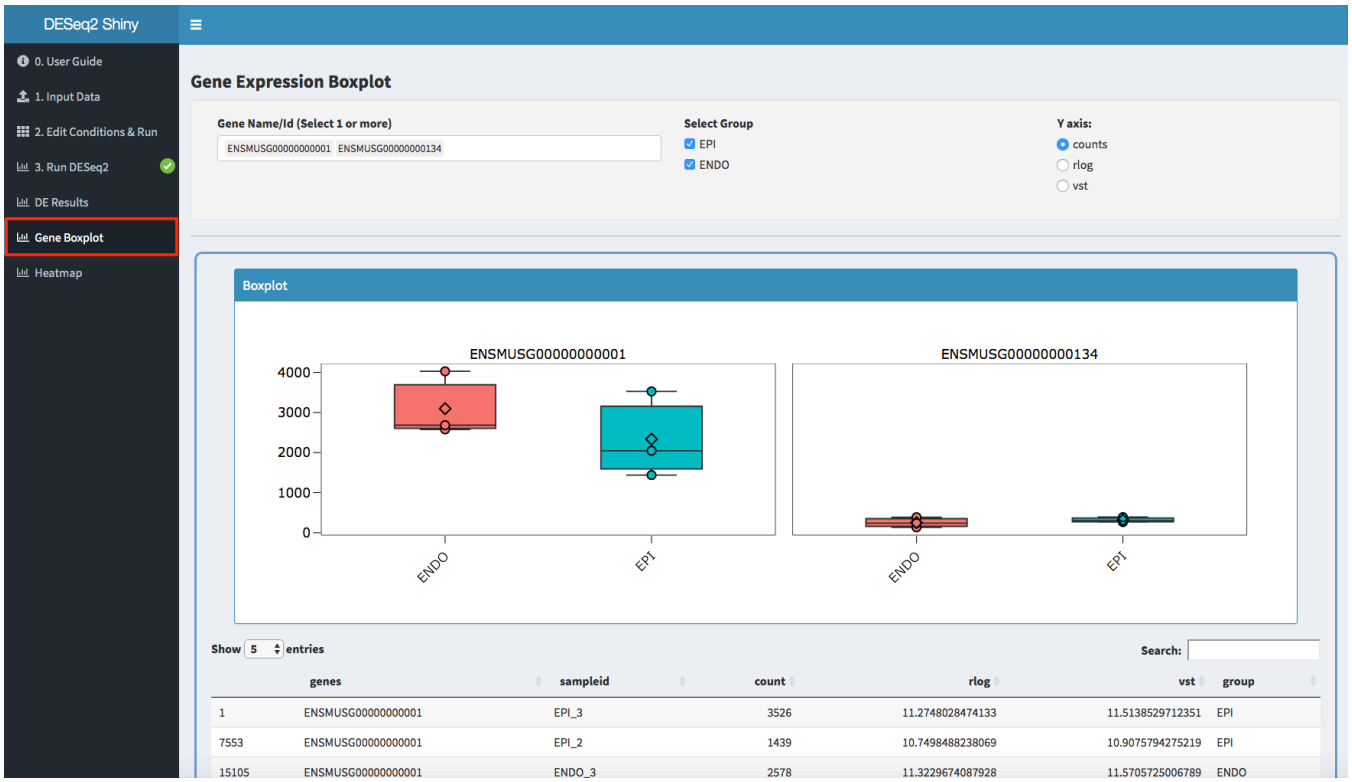
f) Scroll down and you can see the DE comparison results. Click "**Download .csv**". This will save a CSV file locally called "EPI_vs_ENDO.csv".



	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj
ENSMUSG00000000001	2581.00387520778	-0.600010672809525	0.312118852206317	-1.92237882642509	0.0545581078433861	0.19016304972141
ENSMUSG00000000028	589.717664209934	-0.0883117307996088	0.536854240184615	-0.164498525277997	0.869338707592496	0.940759748341336
ENSMUSG00000000031	88199.6173627417	1.01727552759025	0.319192101254636	3.18703227176264	0.0014374073567019	0.0129737947109751

g) There are other visualization plots like gene box plots and clustered heatmap

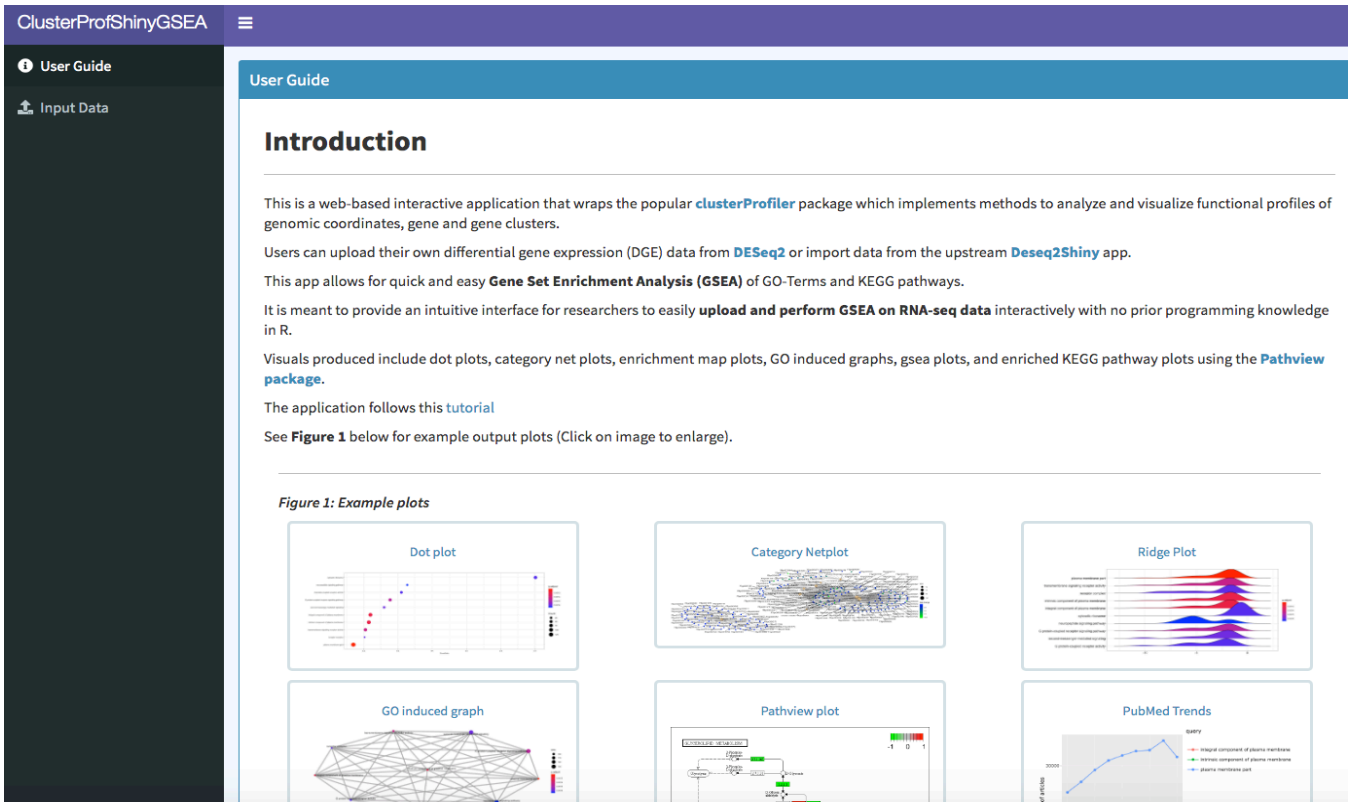
Box Plots:



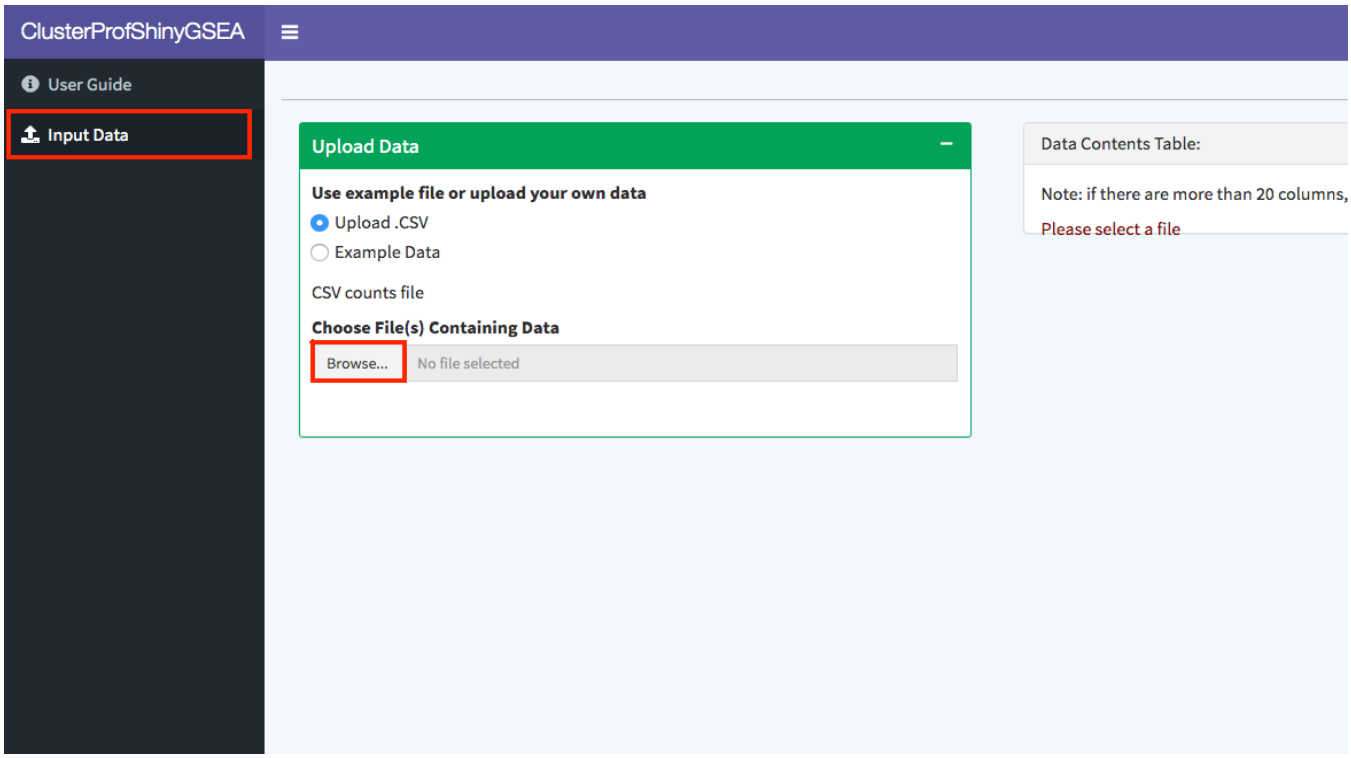
Clustered Heatmap:

Step 3: Gene Set Enrichment Analysis (GSEA):

a) Launch ClusterProfShinyGSEA http://nasqar.abudhabi.nyu.edu/ClusterProfShinyGSEA/

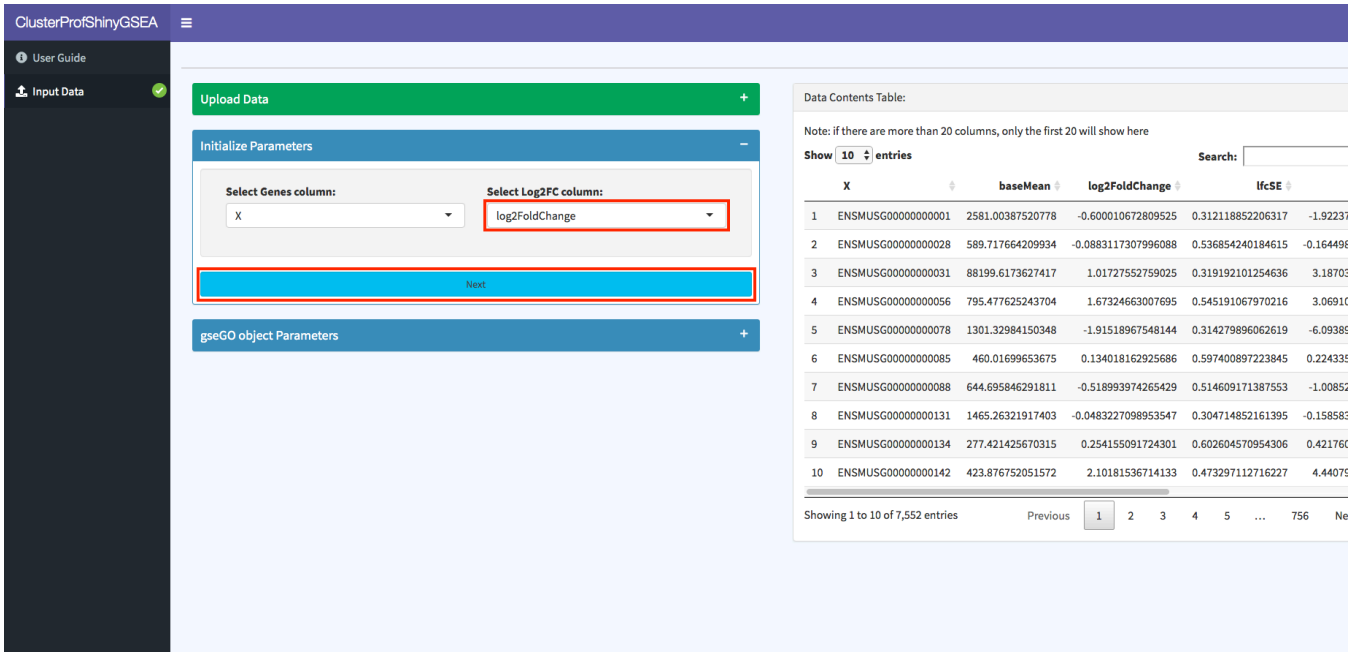


b) Go to "**Input Data**" tab and click "**Browse**"

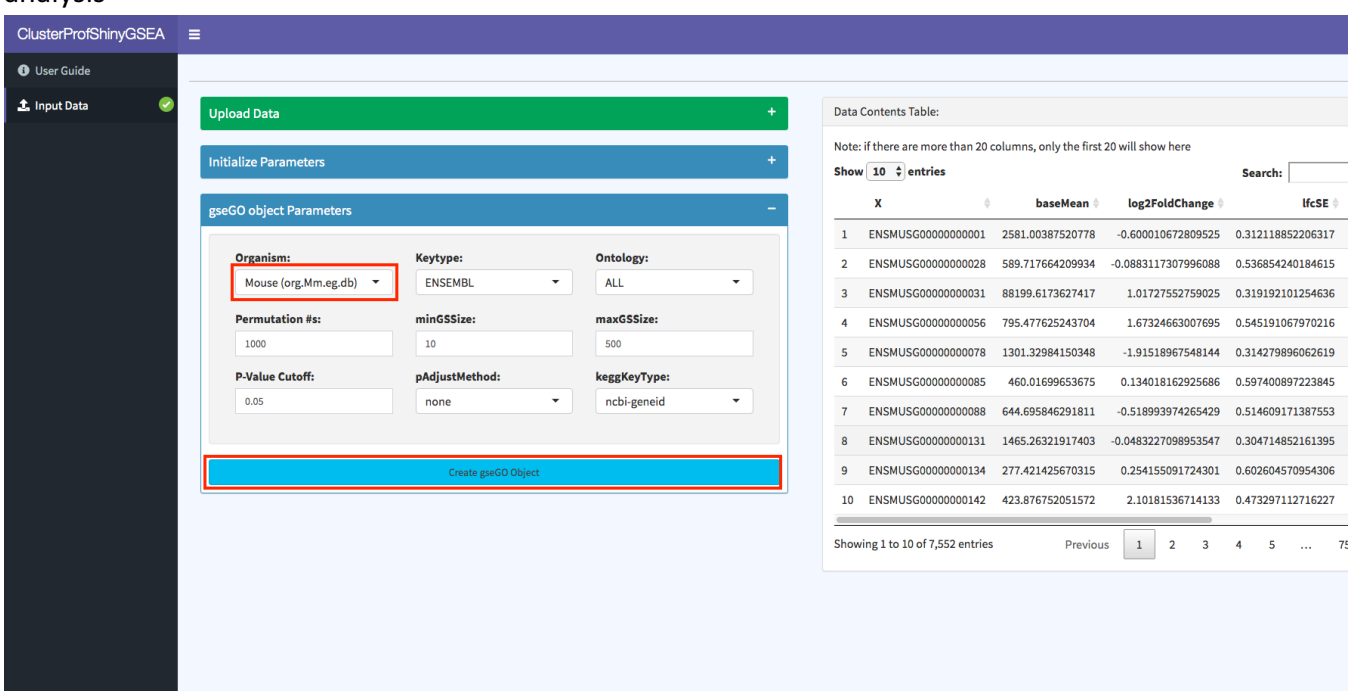


c) Select the DE results file downloaded in the previous section (Section 2, Step f) from Deseq2Shiny (**EPI_vs_ENDO.csv**) to upload.

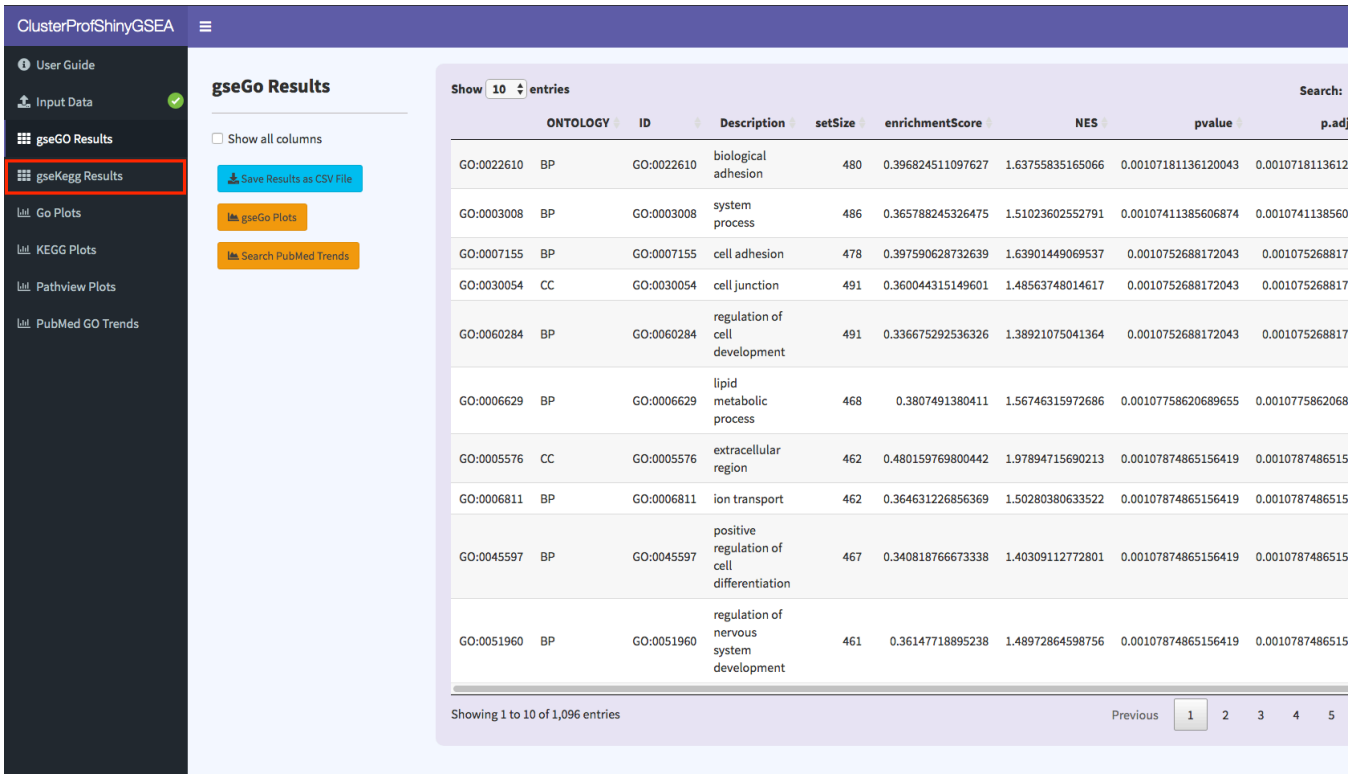
d) In the tab "Initialize Parameters", make sure to select the correct column name that corresponds to the LogFC column and click "**Next**"



e) Select "**Mouse (org.Mm.eg.db)**" as the organism and click "**Create gseGO Object**" to start the analysis

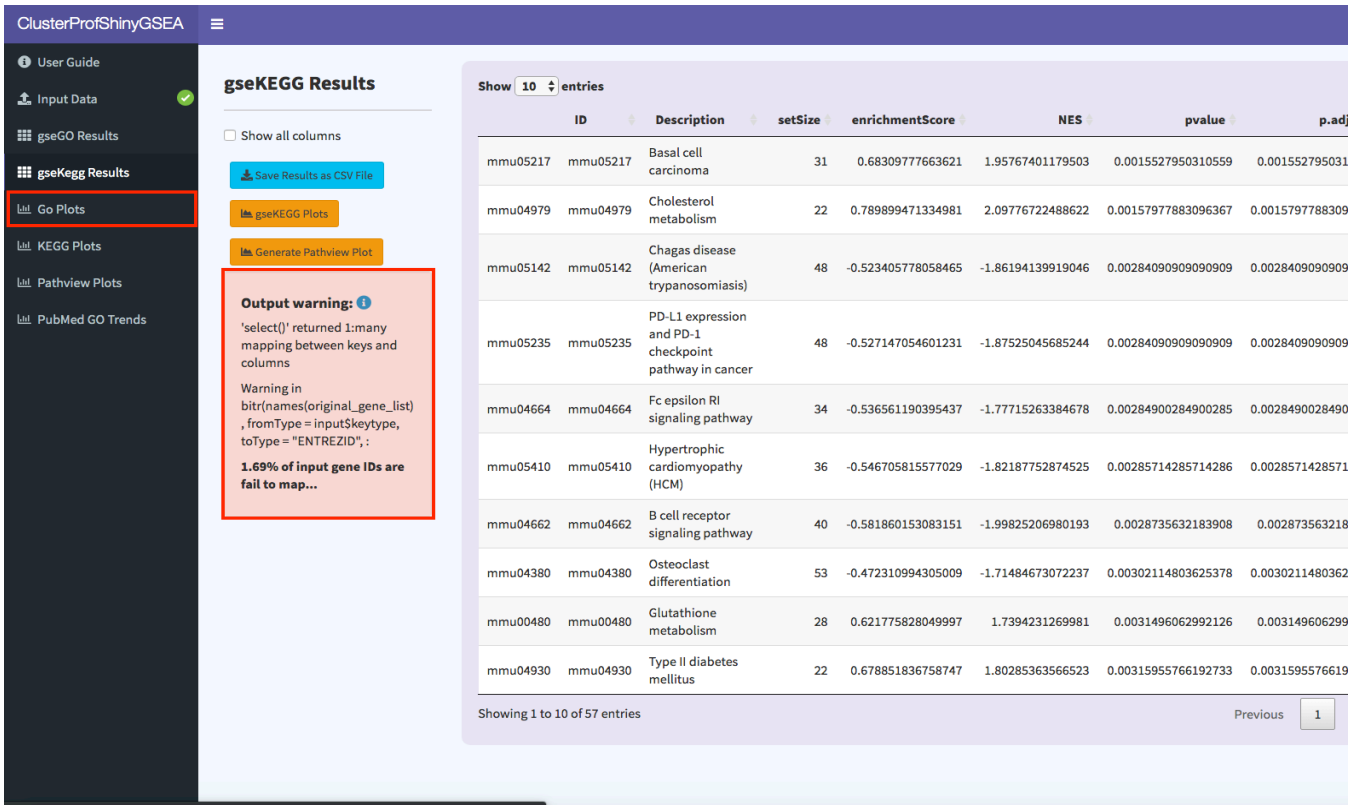


f) Now you can see the gseGO results table. Next go to "**gseKegg Results**" tab to view gseKEGG results table.



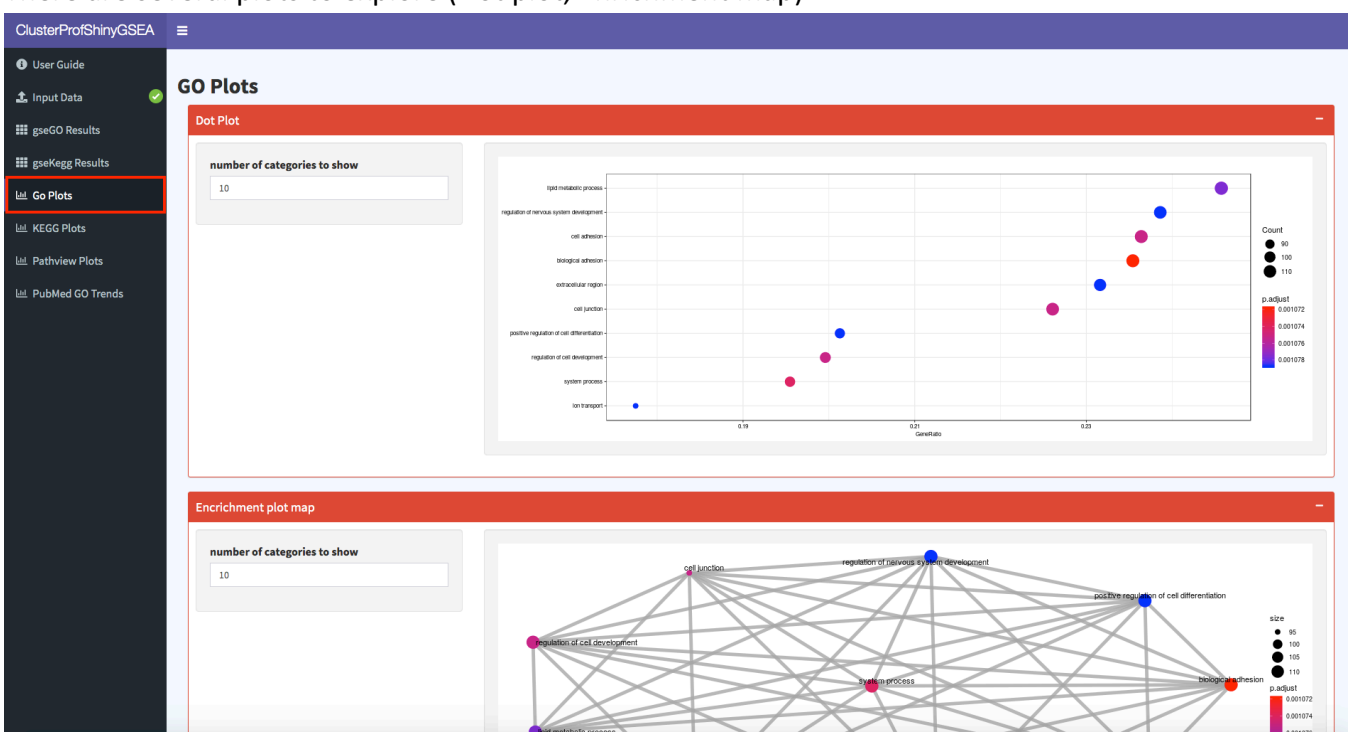
	ONTOLOGY	ID	Description	setSize	enrichmentScore	NES	pvalue	p.adj
GO:0022610	BP	GO:0022610	biological adhesion	480	0.396824511097627	1.63755835165066	0.00107181136120043	0.0010718113612
GO:0003008	BP	GO:0003008	system process	486	0.365788245326475	1.51023602552791	0.00107411385606874	0.0010741138560
GO:0007155	BP	GO:0007155	cell adhesion	478	0.397590628732639	1.63901449069537	0.0010752688172043	0.001075268817
GO:0030054	CC	GO:0030054	cell junction	491	0.360044315149601	1.48563748014617	0.0010752688172043	0.001075268817
GO:0060284	BP	GO:0060284	regulation of cell development	491	0.336675292536326	1.38921075041364	0.0010752688172043	0.001075268817
GO:0006629	BP	GO:0006629	lipid metabolic process	468	0.3807491380411	1.56746315972686	0.00107758620689655	0.0010775862068
GO:0005576	CC	GO:0005576	extracellular region	462	0.480159769800442	1.97894715690213	0.00107874865156419	0.0010787486515
GO:0006811	BP	GO:0006811	ion transport	462	0.364631226856369	1.50280380633522	0.00107874865156419	0.0010787486515
GO:0045597	BP	GO:0045597	positive regulation of cell differentiation	467	0.340818766673338	1.40309112772801	0.00107874865156419	0.0010787486515
GO:0051960	BP	GO:0051960	regulation of nervous system development	461	0.36147718895238	1.48972864598756	0.00107874865156419	0.0010787486515

g) Here are the Kegg results with an output that indicates the percentage of genes that were not mapped. Next go to "**GO Plots**" tab

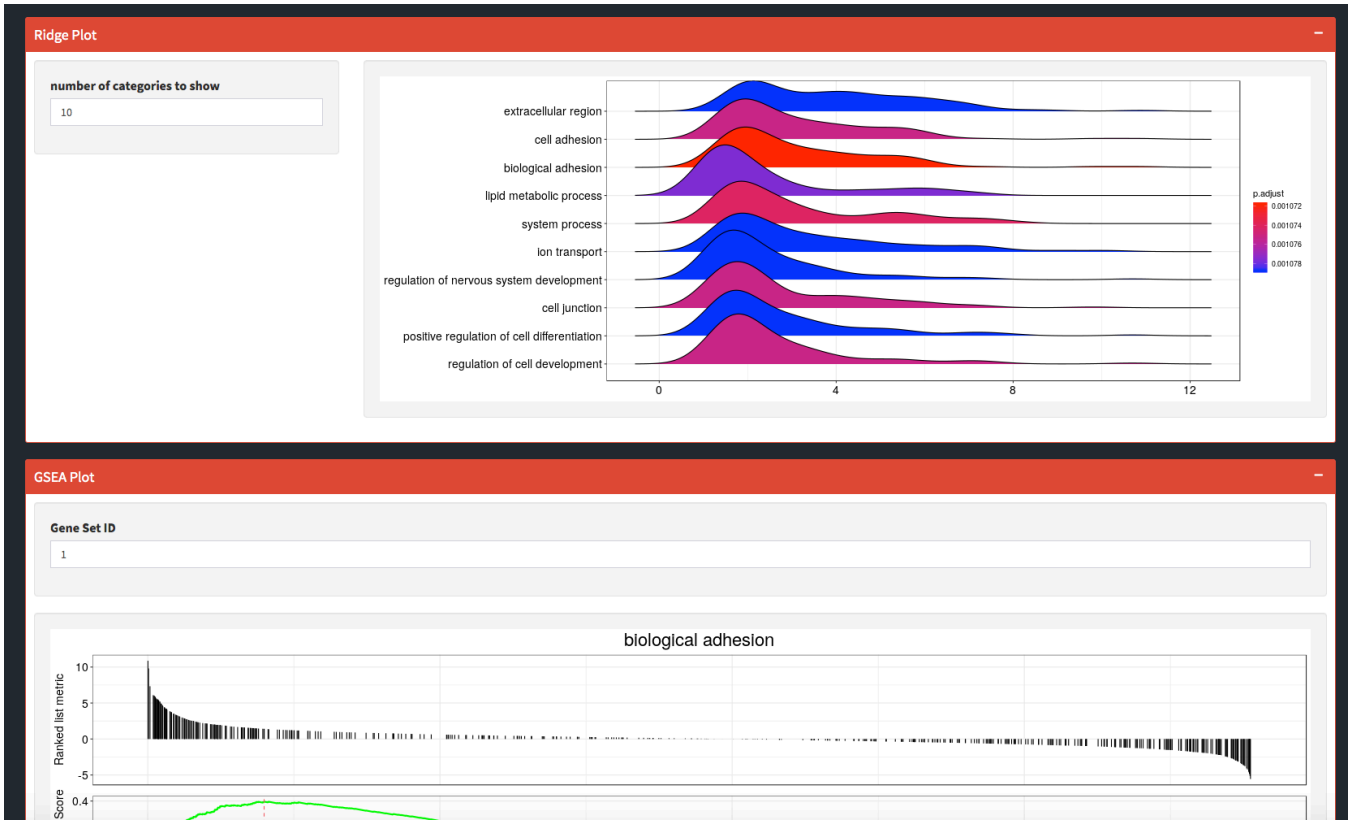


	ID	Description	setSize	enrichmentScore	NES	pvalue	p.adj
mmu05217	mmu05217	Basal cell carcinoma	31	0.68309777663621	1.95767401179503	0.0015527950310559	0.001552795031
mmu04979	mmu04979	Cholesterol metabolism	22	0.789899471334981	2.09776722488622	0.00157977883096367	0.0015797788309
mmu05142	mmu05142	Chagas disease (American trypanosomiasis)	48	-0.523405778058465	-1.86194139919046	0.00284090909090909	0.0028409090909
mmu05235	mmu05235	PD-L1 expression and PD-1 checkpoint pathway in cancer	48	-0.527147054601231	-1.87525045685244	0.00284090909090909	0.0028409090909
mmu04664	mmu04664	Fc epsilon RI signaling pathway	34	-0.536561190395437	-1.77715263384678	0.00284900284900285	0.0028490028490
mmu05410	mmu05410	Hypertrophic cardiomyopathy (HCM)	36	-0.546705815577029	-1.82187752874525	0.00285714285714286	0.0028571428571
mmu04662	mmu04662	B cell receptor signaling pathway	40	-0.581860153083151	-1.99825206980193	0.0028735632183908	0.002873563218
mmu04380	mmu04380	Osteoclast differentiation	53	-0.472310994305009	-1.71484673072237	0.00302114803625378	0.0030211480362
mmu00480	mmu00480	Glutathione metabolism	28	0.621775828049997	1.7394231269981	0.0031496062992126	0.003149606299
mmu04930	mmu04930	Type II diabetes mellitus	22	0.678851836758747	1.80285363566523	0.00315955766192733	0.0031595576619

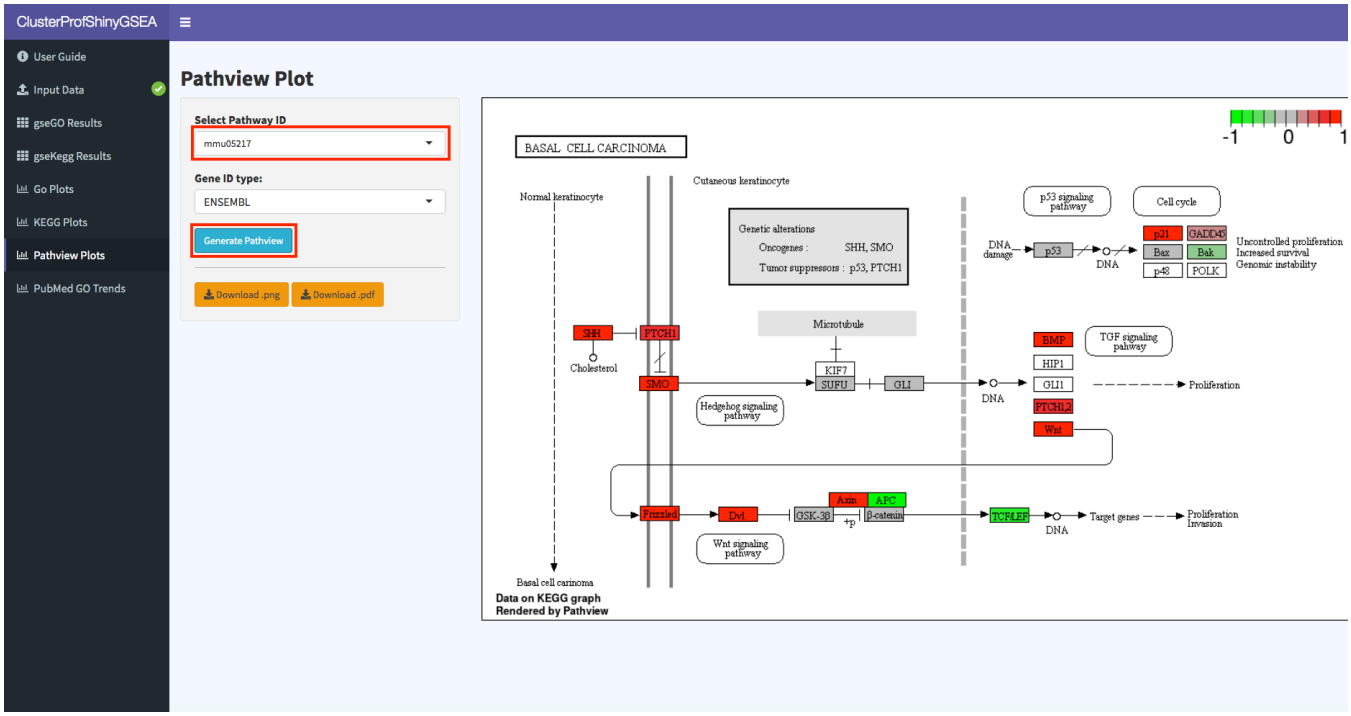
h) There are several plots to explore (Dot plot, Enrichment map)



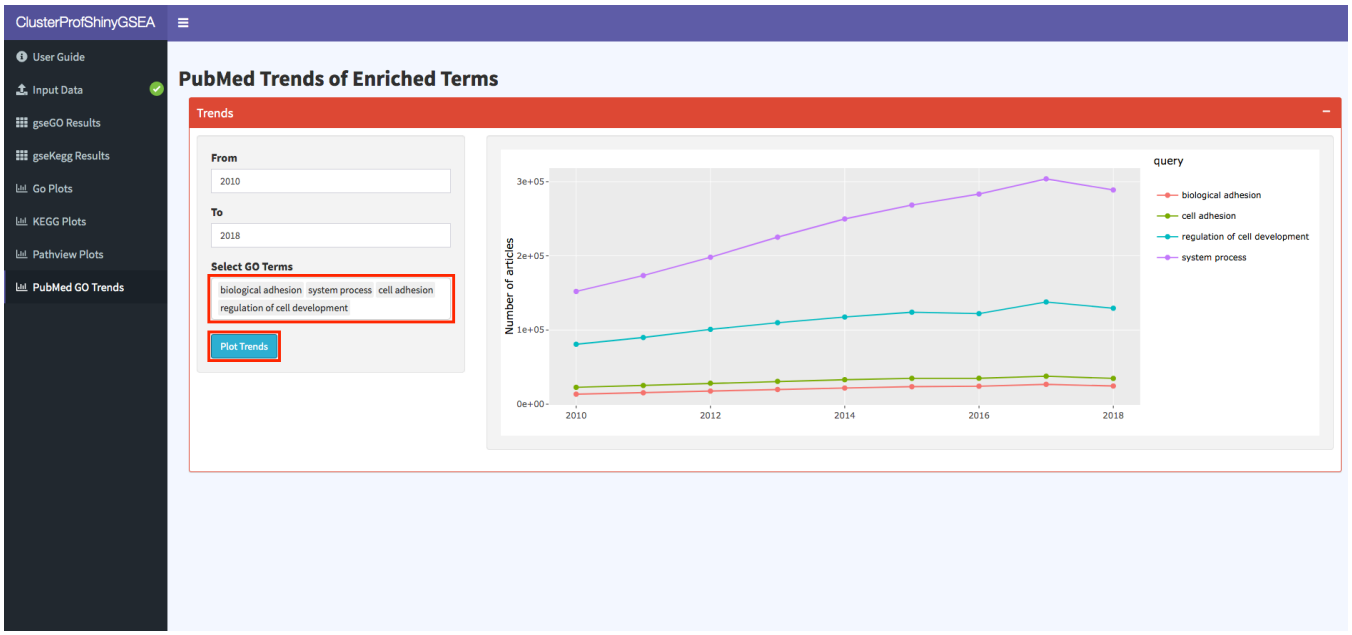
i) Ridge plot and GSEA plot



j) Next go to "**Pathview Plots**" tab and select gene "**mmu05217**" and click "**Generate Pathview**"



k) Last step is to check pubmed trends for enriched categories. Select a few GO terms and click "**Plot Trends**"

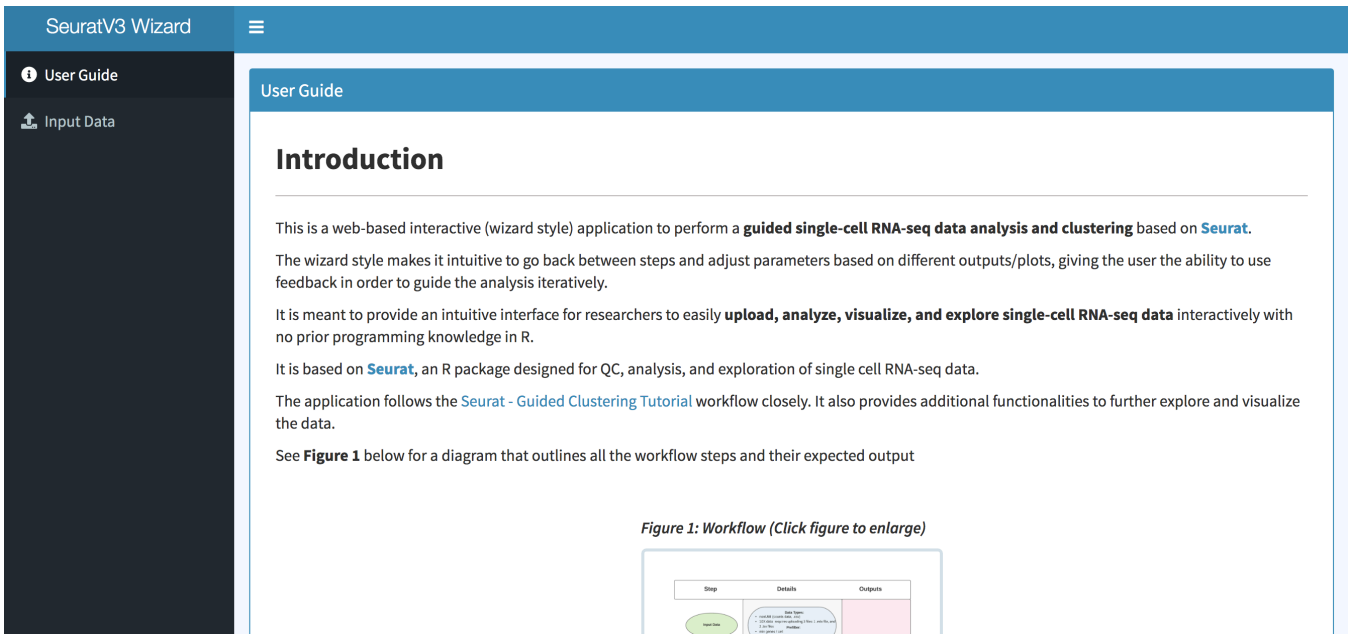


3) Example Use Case 2 (scRNA-seq analysis using Seurat):

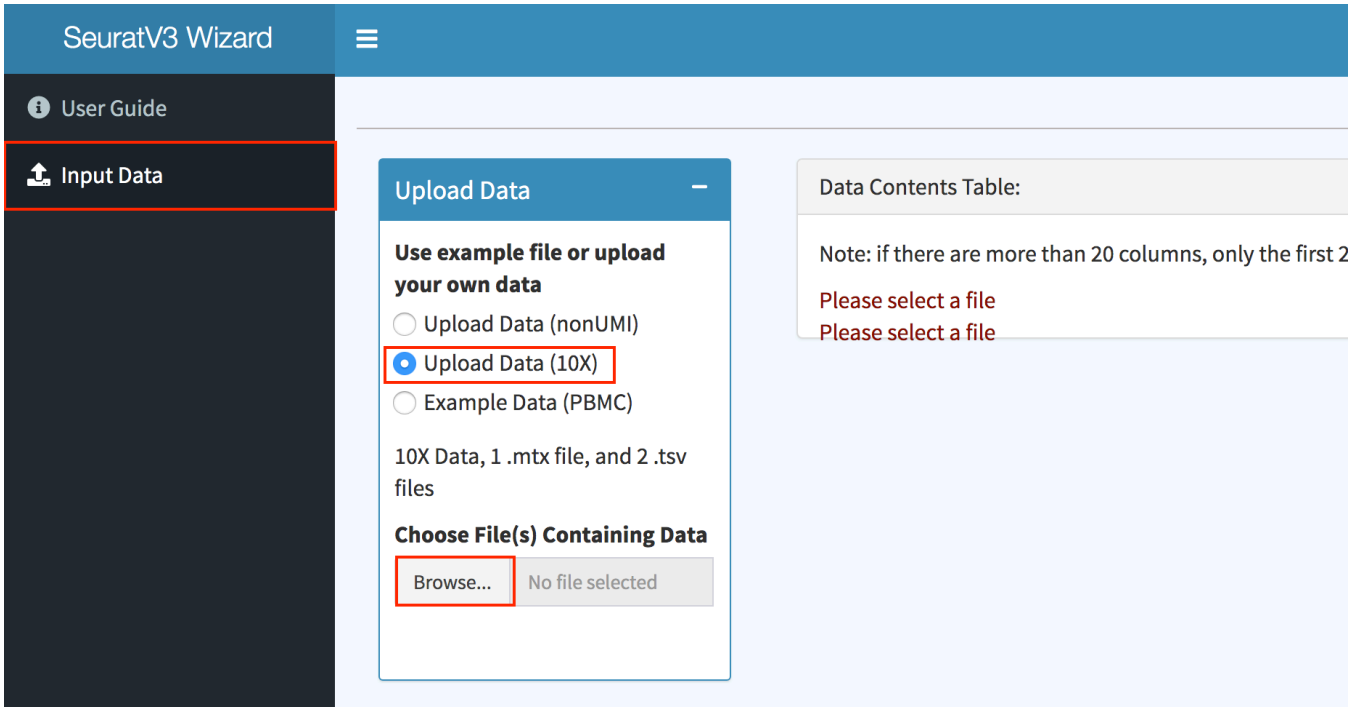
In this example we will show how you can easily upload your single-cell RNA seq sample data (10X format, using the 10x genomics cellranger pipeline) and perform a guided single-cell data analysis and visualization using the popular Seurat library. You can follow along using the Seurat tutorial (https://satijalab.org/seurat/v3.0/pbmc3k_tutorial.html), which the SeuratV3Wizard mirrors closely.

a) Download and extract data file from https://s3-us-west-2.amazonaws.com/10x.files/samples/cell/pbmc3k/pbmc3k_filtered_gene_bc_matrices.tar.gz

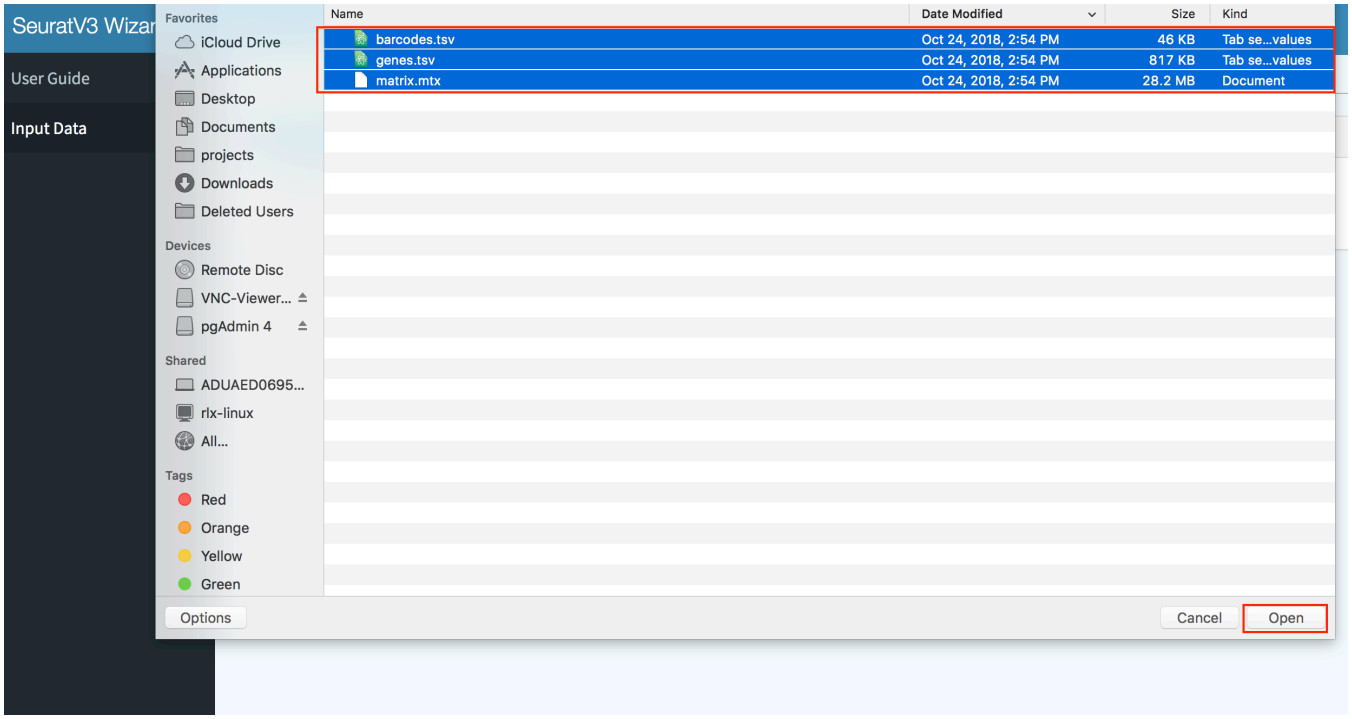
b) Launch SeuratV3Wizard (http://nasqar.abudhabi.nyu.edu/SeuratV3Wizard/)



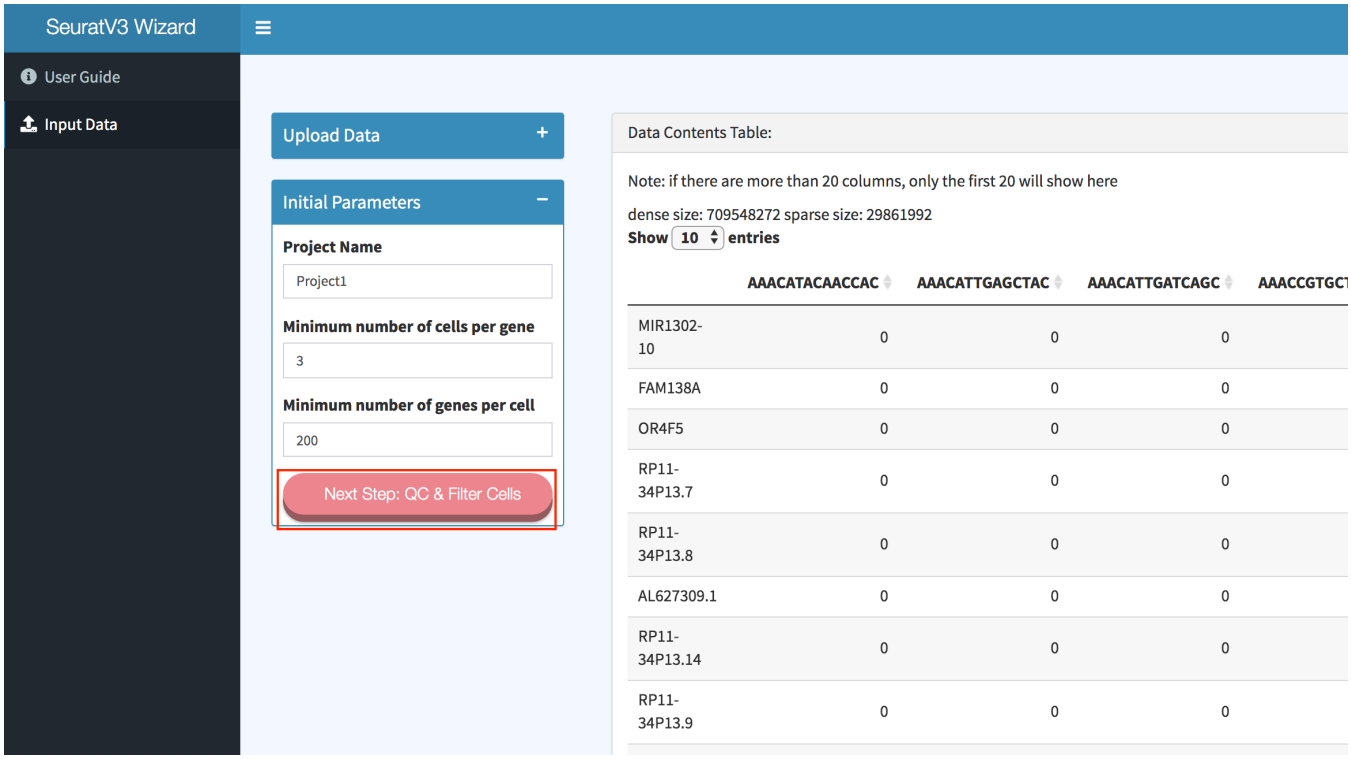
c) Go to "**Input Data**" tab and select "**Upload Data (10X)**"



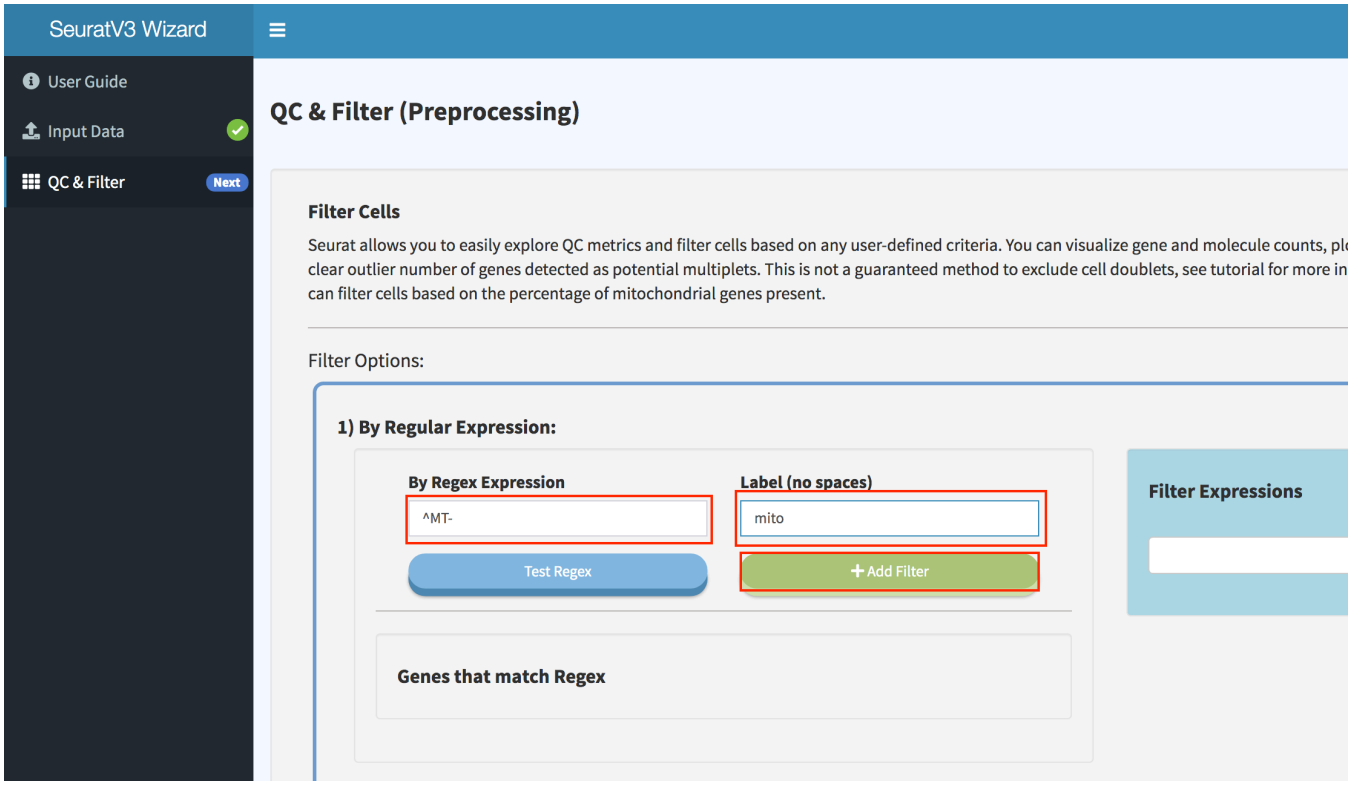
d) Browse for the extracted data files downloaded earlier. Select all 3 files to upload



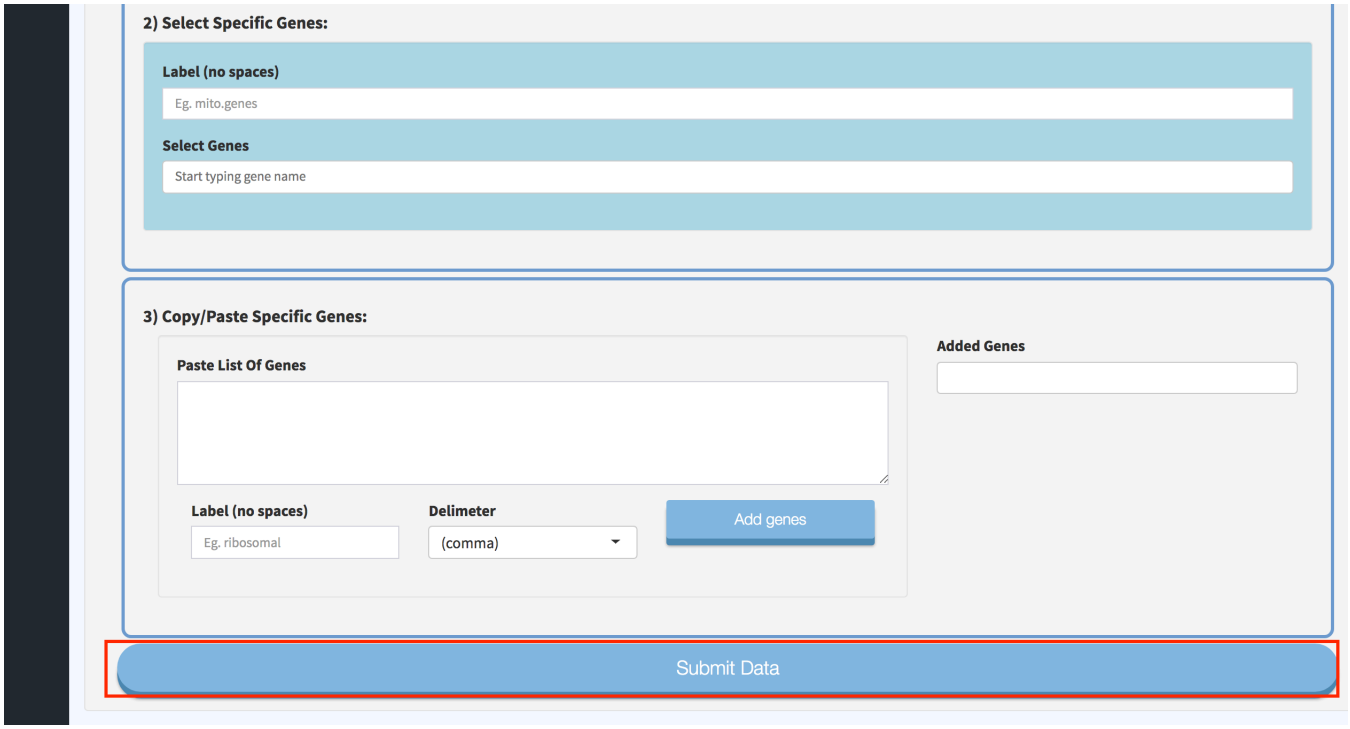
e) Click "**Next Step: QC & Filter Cells**"



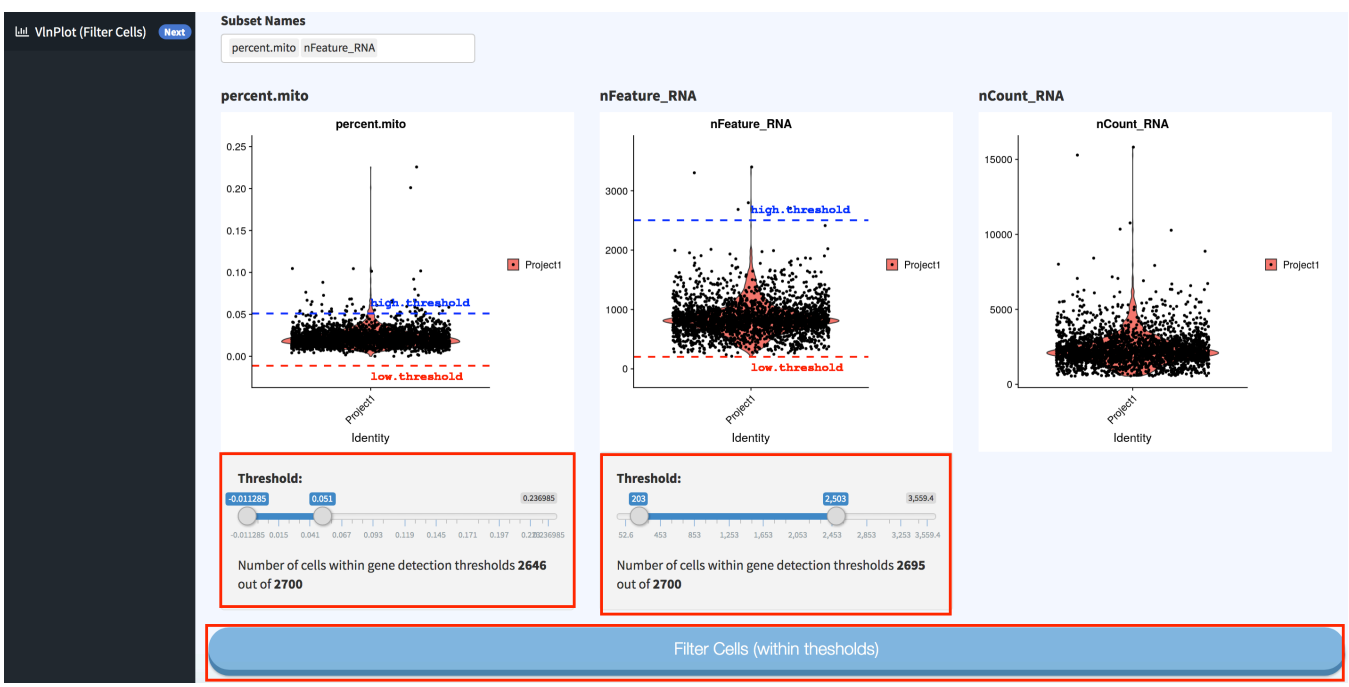
f) Add a regular expression for mitochondrial genes by adding the **regex** and **label** as seen below, and then click "**Add Filter**". You can use the "**Test Regex"** tab in order to verify that the regular expression the user supplies works as expected.



g) Scroll down and click "**Submit Data**"

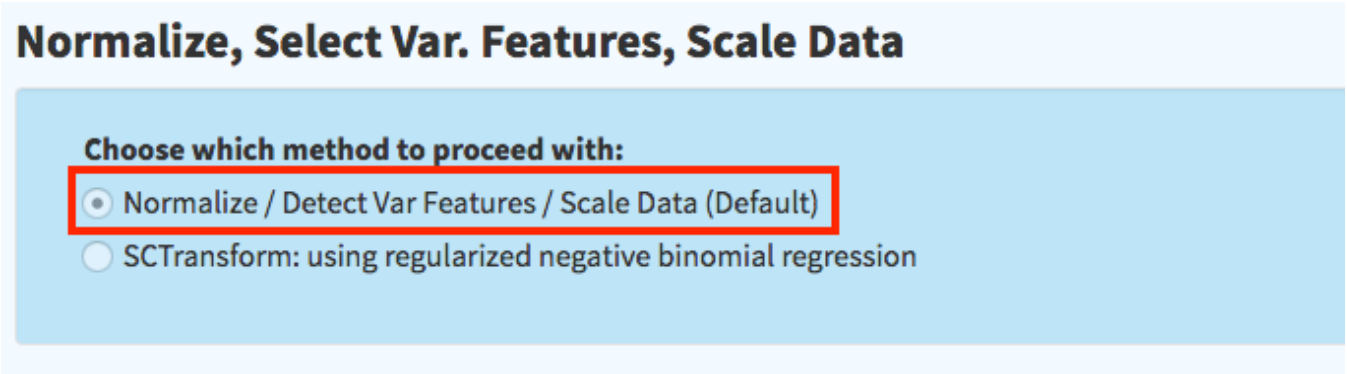


h) Select the low and high thresholds to filter out cells. Click "**Filter Cells (within thresholds)**". This is an interactive filter that will display the effects of applying different filtering thresholds on the data.

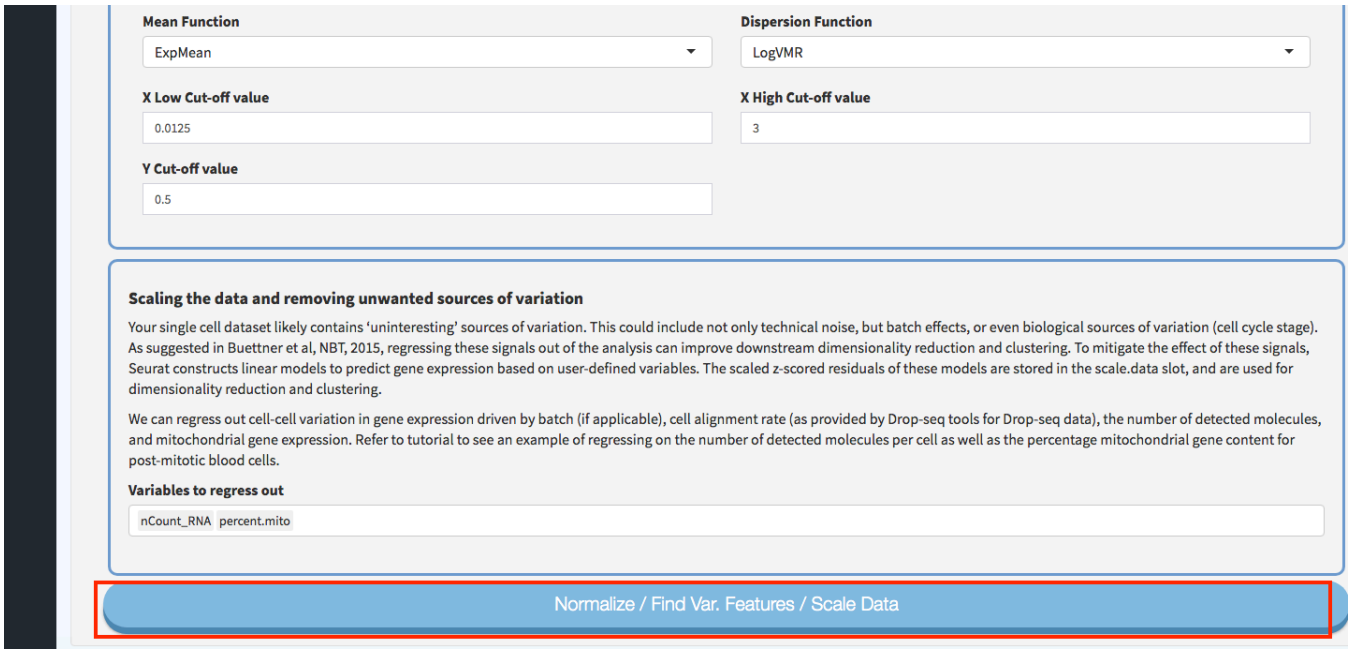


i) There are now two options to proceed within the analysis. We will choose the default one so we can mirror the tutorial. The second option is the SCTransform method

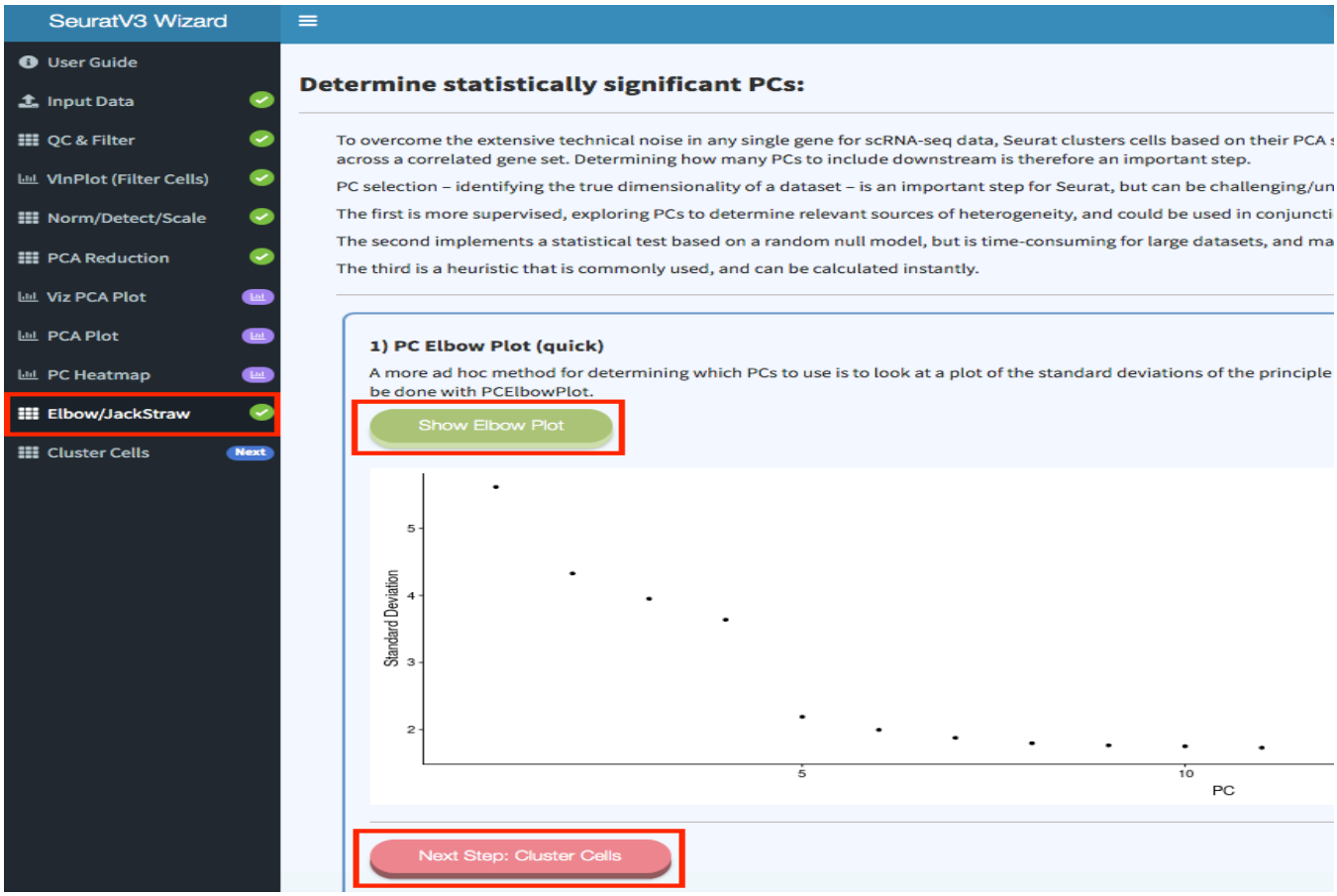
(https://rawgit.com/ChristophH/sctransform/master/inst/doc/seurat.html)



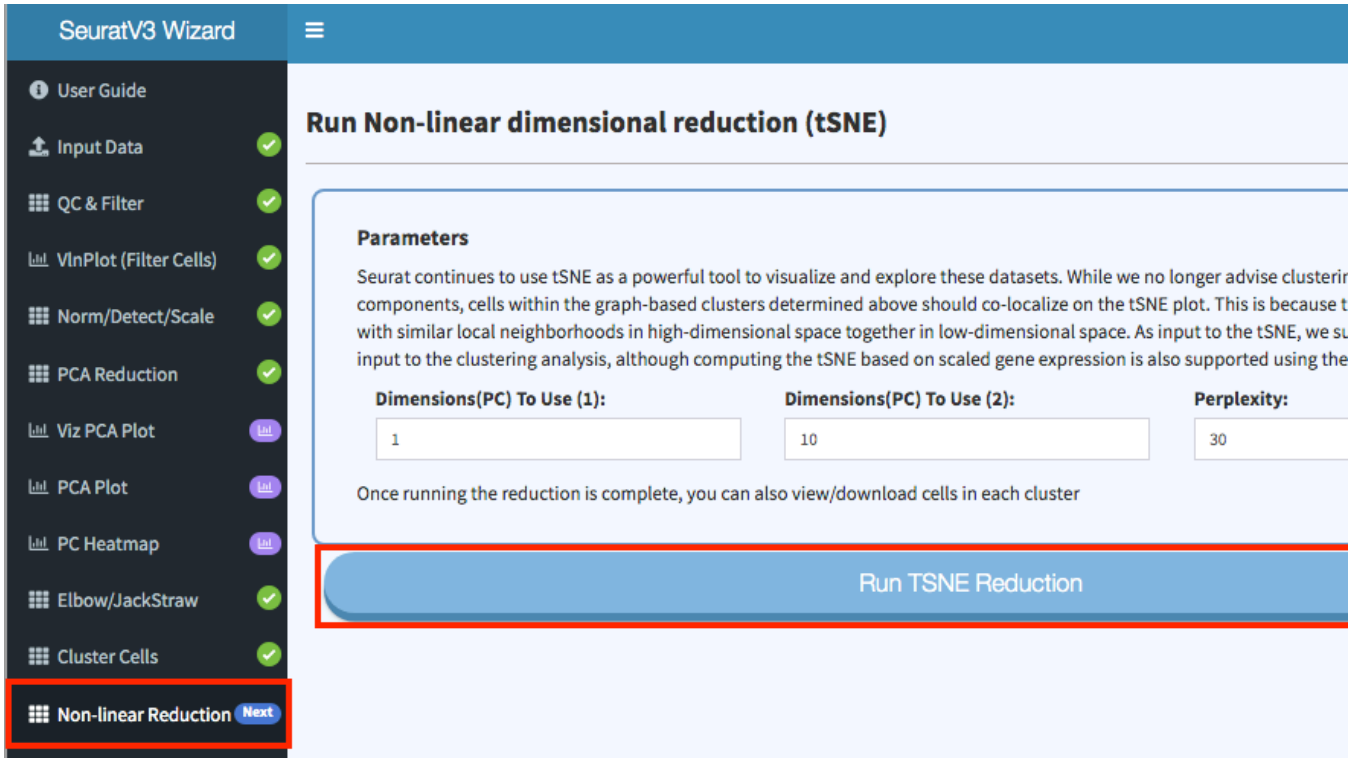
j) Proceed by clicking "**Normalize / Find Var. Feaatures / Scala Data**"



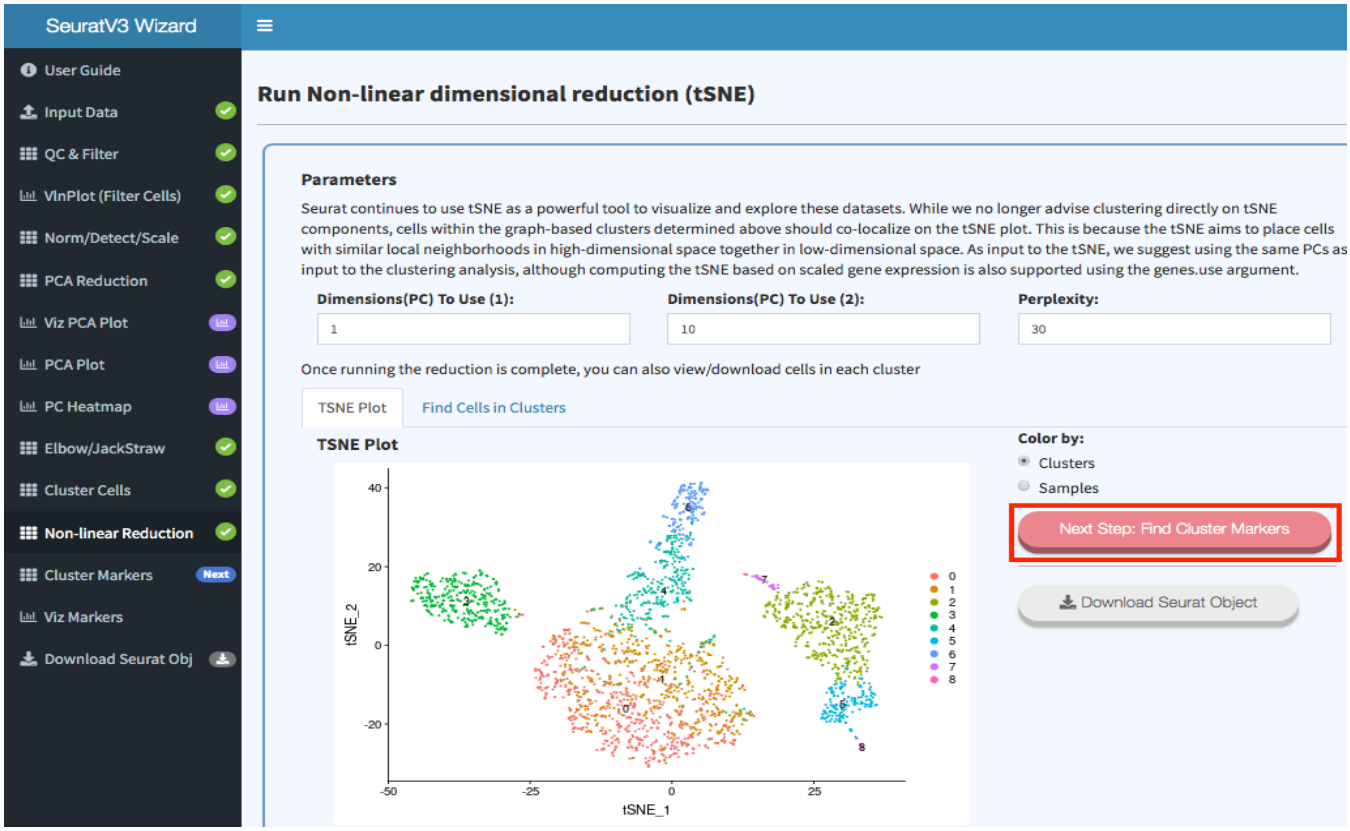
k) Now you can follow the default settings in the wizard to go through the steps until "**Elbow/Jackstraw**". Click "**Show Elbow Plot**" and click "**Next Step: Cluster Cells**"



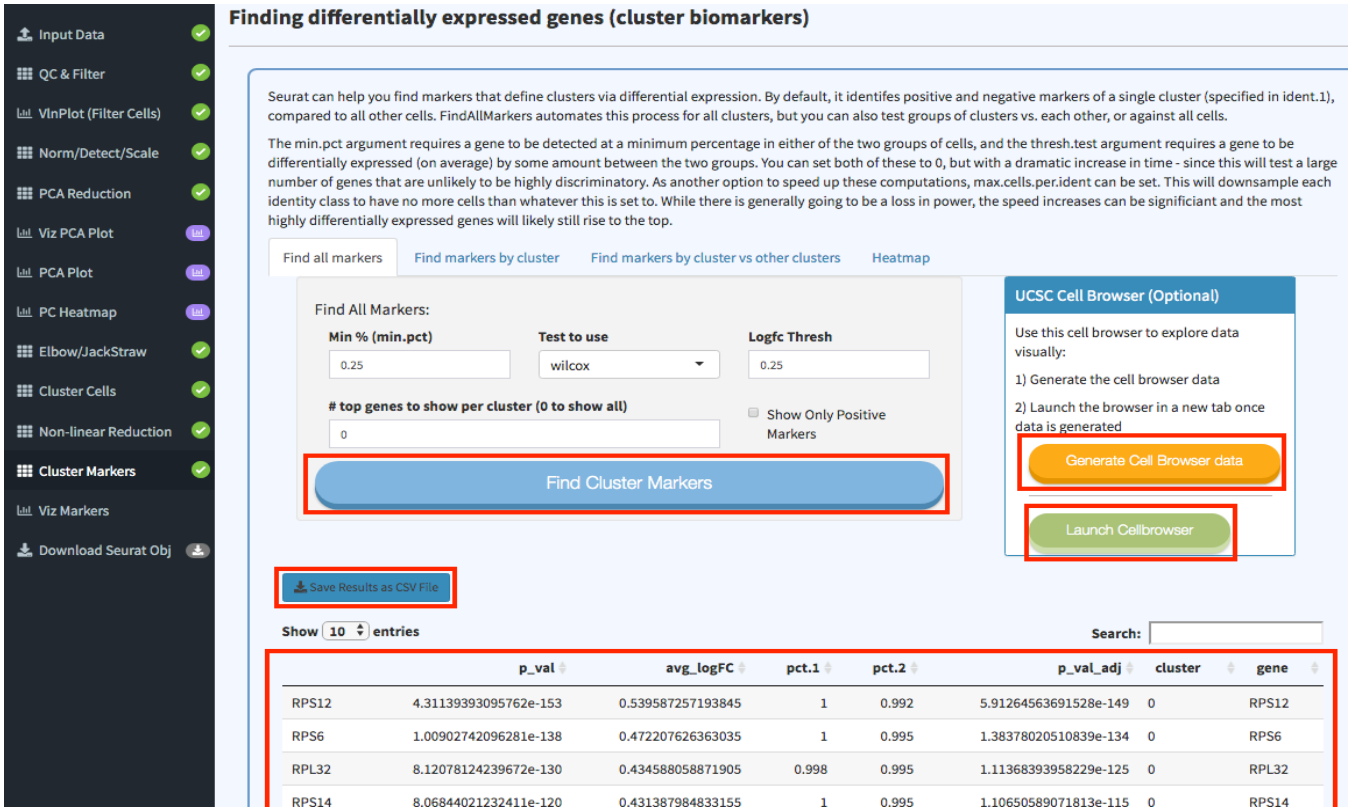
l) Next run "**Cluster Cells**" and go to "**Non-linear Reduction**" tab and "**Run TSNE Reduction**"



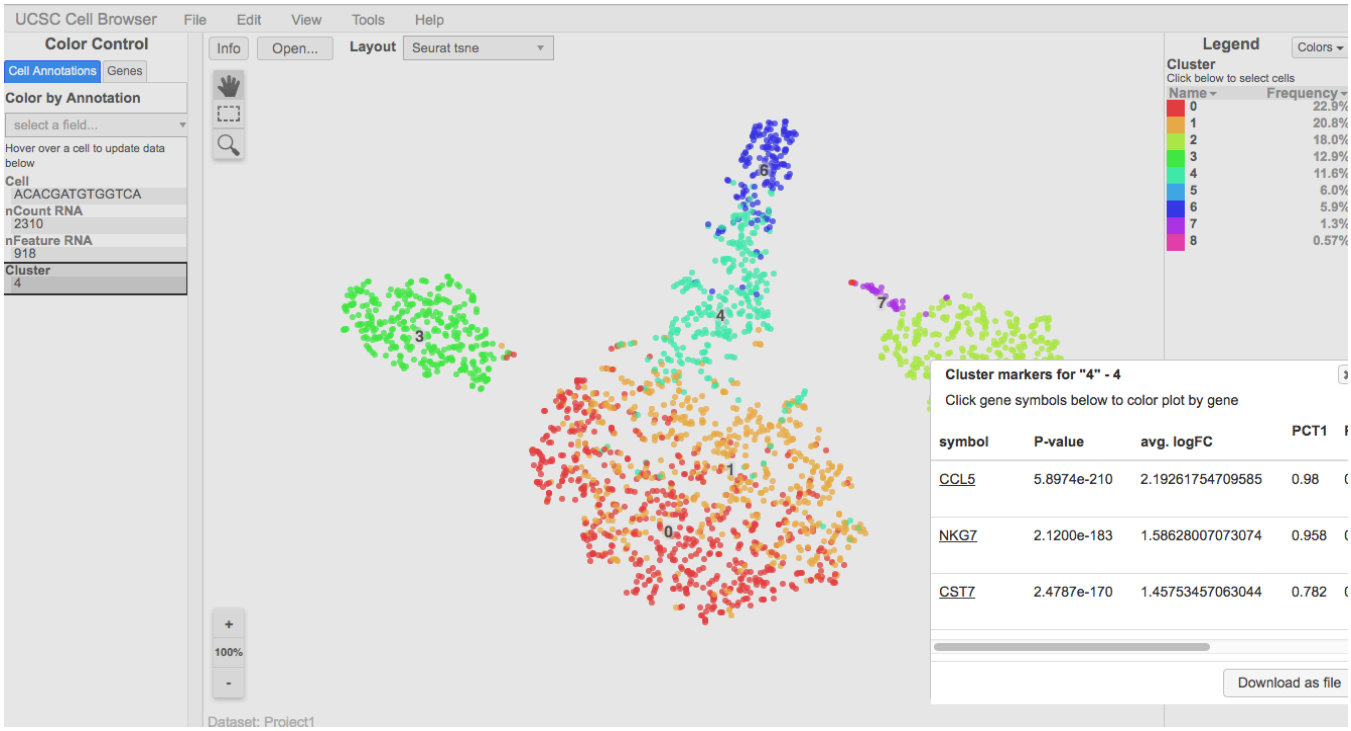
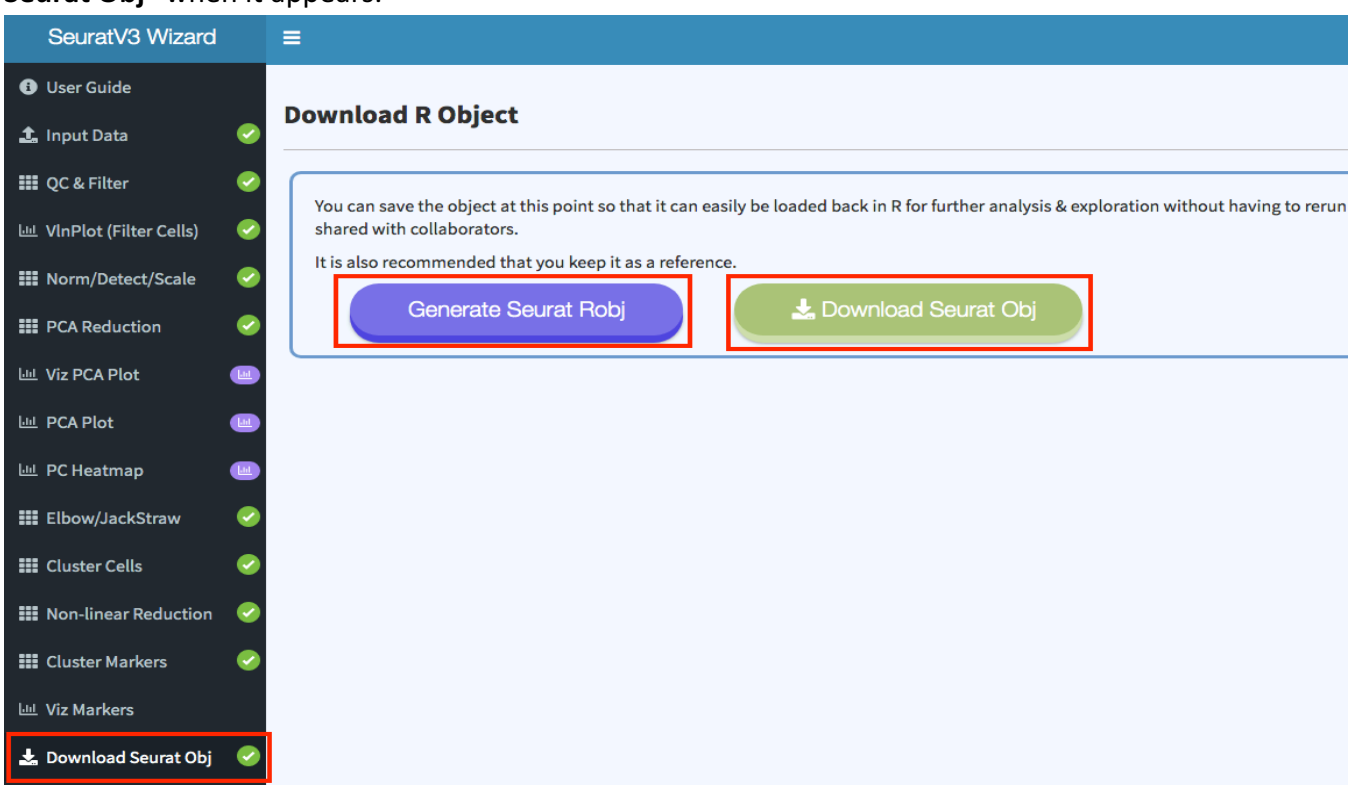
m) Once tSNE step is done, we can now go to the next step "**Next Step: Find Cluster Markers**" and click "**Find Cluster Markers**"



n) Now you can see the markers table (which is downloadable). There is also an option to explore the clusters and markers visually using UCSC Cellbrowser (https://github.com/maximilianh/cellBrowser). Click "**Generate Cell browser data**". Once done you can launch the cellbrowser by clicking on "**Launch Cellbrowser**".



o) A tab is opened with UCSC cellbrowser



p) Last step would be to download the Seurat object for reference and further analysis in R. Go to the last tab "**Download Seurat Obj**". Click "**Generate Seurat Obj**", once done click "**Download Seurat Obj**" when it appears.