DEBrowser Documentation

Release 1.14.0

Alper Kucukural, Manuel Garber, Onur Yukselen

Nov 07, 2019

Contents

1 Quick-start Guide **3**
1.1 Getting Started 3
1.2 Low Count Filtering 6
1.3 Batch Effect Correction and Normalization 7
1.4 DE Analysis 9
1.5 The Main Plots of DE Analysis 10
1.6 The Heatmap of DE Analysis 12
1.7 GO Term Plots 12
1.8 Data Tables 13
1.9 Quality Control Plots 15
1.10 Autoload Data via Hyperlink 22

2 Installation Guide **25**
2.1 Operating System Dependencies 26

3 DE Analysis **27**
3.1 Introduction 27
3.2 DESeq2 27
3.3 Un-normalized counts 27
3.4 Used parameters for DESeq2 28
3.5 EdgeR 28
3.6 Used parameters for EdgeR 28
3.7 Limma 28
3.8 Used parameters for Limma 29
3.9 ComBat 29
3.10 DEBrowser 29
3.11 References 30

4 Heatmaps **33**
4.1 Used clustering and linkage methods in heatmap 33
4.2 Used distance methods in heatmap 34
4.3 Interactive Heatmap 34
4.4 The Heatmap of DE Analysis 35
4.5 The Scale Option of Heatmap 35

5 DEBrowser Modules **37**
5.1 Demo Data 37

5.2 Example Modules 38

6 DEBrowser Additions 45

7 Examples 47
7.1 QC plots without Batch Effect Correction 47
7.2 QC plots after Batch Effect Correction 51
7.3 The Differential Expression Plots 54
7.4 GO Term Plots 59
7.5 Log2 fold change comparison for PPARα pathway 61

8 Frequently asked questions (FAQ) 67
8.1 Why un-normalized counts? 67
8.2 Why am I getting error while uploading files? 67
8.3 Why some columns not showed up after upload? 67
8.4 Why am I getting error while uploading CSV/TSV files exported from Excel? 68
8.5 Why can’t I see all the background data in Main Plots? 68
8.6 Why am I getting error when I click on **DE Genes** in Go Term Analysis? 68
8.7 How to download selected data from Main plots/QC Plots/Heatmaps? 68

9 Indices and tables 69

Contents:

CHAPTER 1

Quick-start Guide

This guide is walkthrough for the DEBrowser from start to finish.

1.1 Getting Started

First off, we need to install R package of DEBrowser from bioconductor:

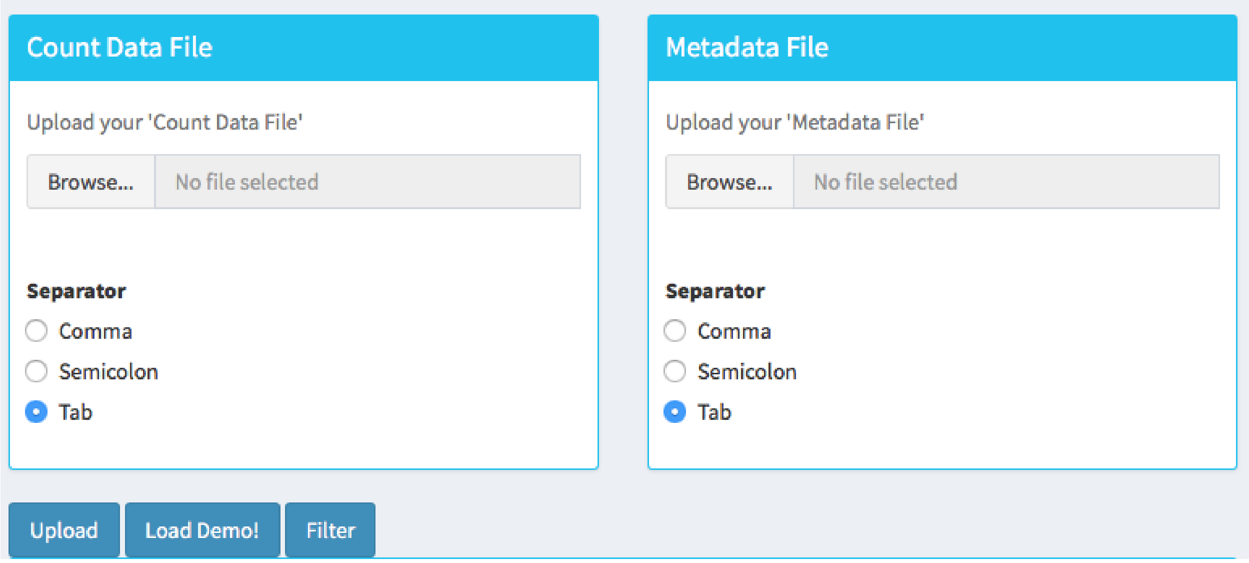
```
source("https://bioconductor.org/biocLite.R")
biocLite("debrowser")
```

One you have installed the R package, you can call these R commands:

```
library(debrowser)
startDEBrowser()
```

Note: For more information on installing DEBrowser locally, please consult our Installation Guide.

Once you've made your way to the website, or you have a local instance of DEBrowser running, you will be greeted with data loading section:



Separator
○ Comma
○ Semicolon
● Tab

Separator
○ Comma
○ Semicolon
● Tab

To begin the analysis, you need to upload your *count data file* (comma or semicolon separated (CSV), and tab separated (TSV) format) to be analyzed and choose appropriate separator for the file (comma, semicolon or tab).

If you do not have a dataset to upload, you can use the built in demo data file by clicking on the 'Load Demo (Vernia et al.)!' button. To view the entire demo data file, you can download this demo set. For another example, try our full dataset (Vernia et. al) .

The structure of the count data files are shown below:

gene	exp1	exp2	cont1	cont2
DQ714	0.00	0.00	0.00	0.00
DQ554	0.00	0.00	0.00	0.00
AK028	2.00	1.29	0.00	0.00

Tip: DEBrowser also accepts count data files via hyperlink, for more information please see the autoload data via hyperlink section.

In addition to the count data file; you might need to upload metadata file to correct for batch effects or any other normalizing conditions you might want to address that might be within your results. To handle for these conditions, simply create a metadata file by using the example table at below or download sample file from this link. Metadata file also simplifies condition selection for complex data. The columns you define in this file can be selected in condition selection page. Make sure you have defined two conditions per column. If there are more than two conditions in a column, those can be defined empty. Please note that, if your data is not complex, metadata file is optional, you don't need to upload.

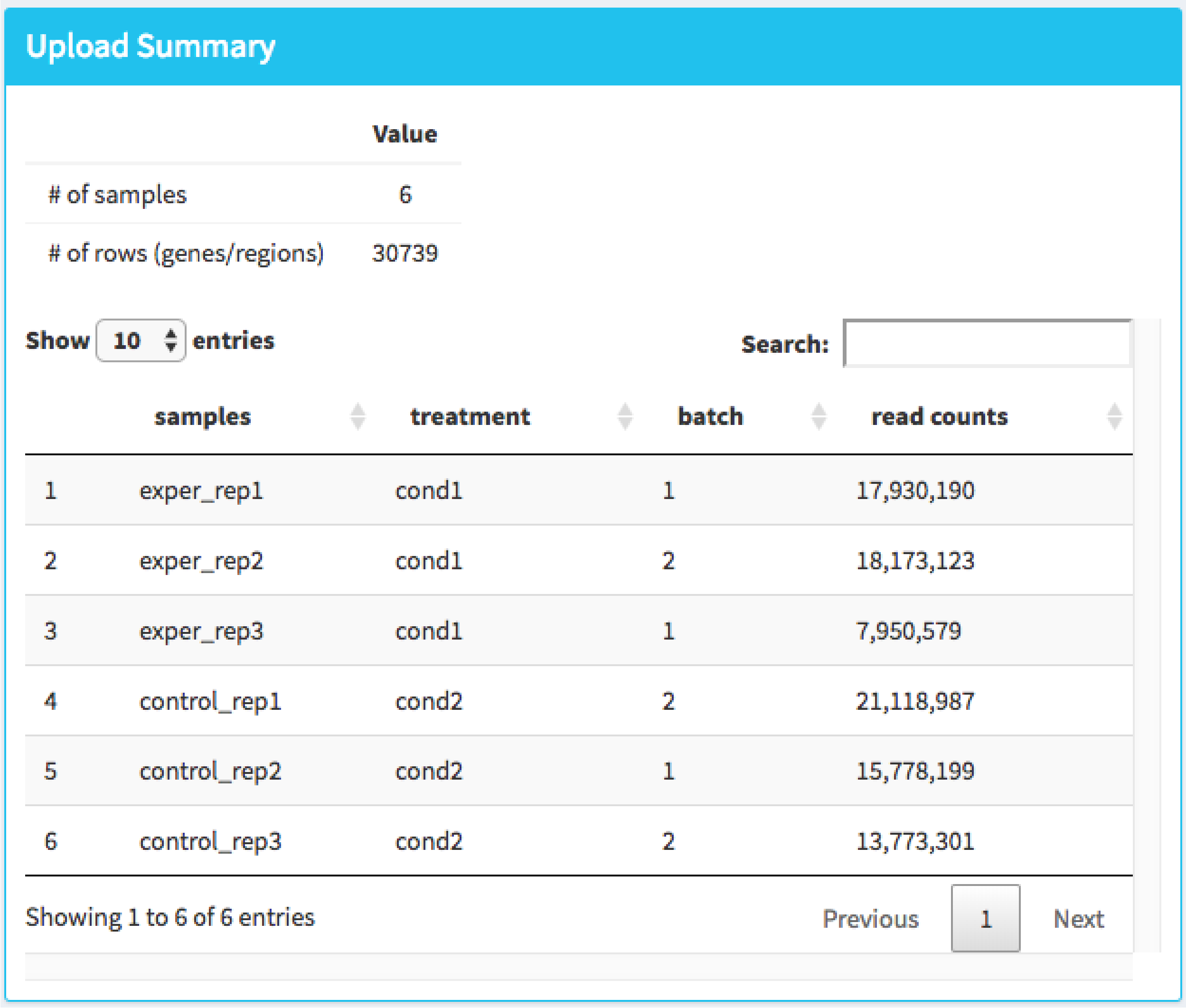
sample	batch	condition
exper_rep1	1	A
exper_rep2	2	A
exper_rep3	1	A
control_rep1	2	B
control_rep2	1	B
control_rep3	2	B

Metadata file can be formatted with comma, semicolon or tab separators similar to count data files. These files used to establish different batch effects for multiple conditions. You can have as many conditions as you may require, as long as all of the samples are present.

Note: The example above would result in the first set of conditions as `exper_rep1`, `exper_rep2`, `exper_rep3` from `A` and second set of conditions as `control_rep1`, `control_rep2`, `control_rep3` from `B` as they correspond to those conditions in the `condition` column.

In the same way, 'batch' would have the first set as `exper_rep1`, `exper_rep3`, `control_rep2` from `1` and second set as `exper_rep2`, `control_rep1`, `control_rep3` from `2` as they correspond to those conditions in the `batch` column.

Once the count data and metadata files have been loaded in DEBrowser, you can click upload button to visualize your data as shown at below:



Upload Summary

	Value
# of samples	6
# of rows (genes/regions)	30739

	samples	treatment	batch	read counts
1	exper_rep1	cond1	1	17,930,190
2	exper_rep2	cond1	2	18,173,123
3	exper_rep3	cond1	1	7,950,579
4	control_rep1	cond2	2	21,118,987
5	control_rep2	cond2	1	15,778,199
6	control_rep3	cond2	2	13,773,301

Showing 1 to 6 of 6 entries

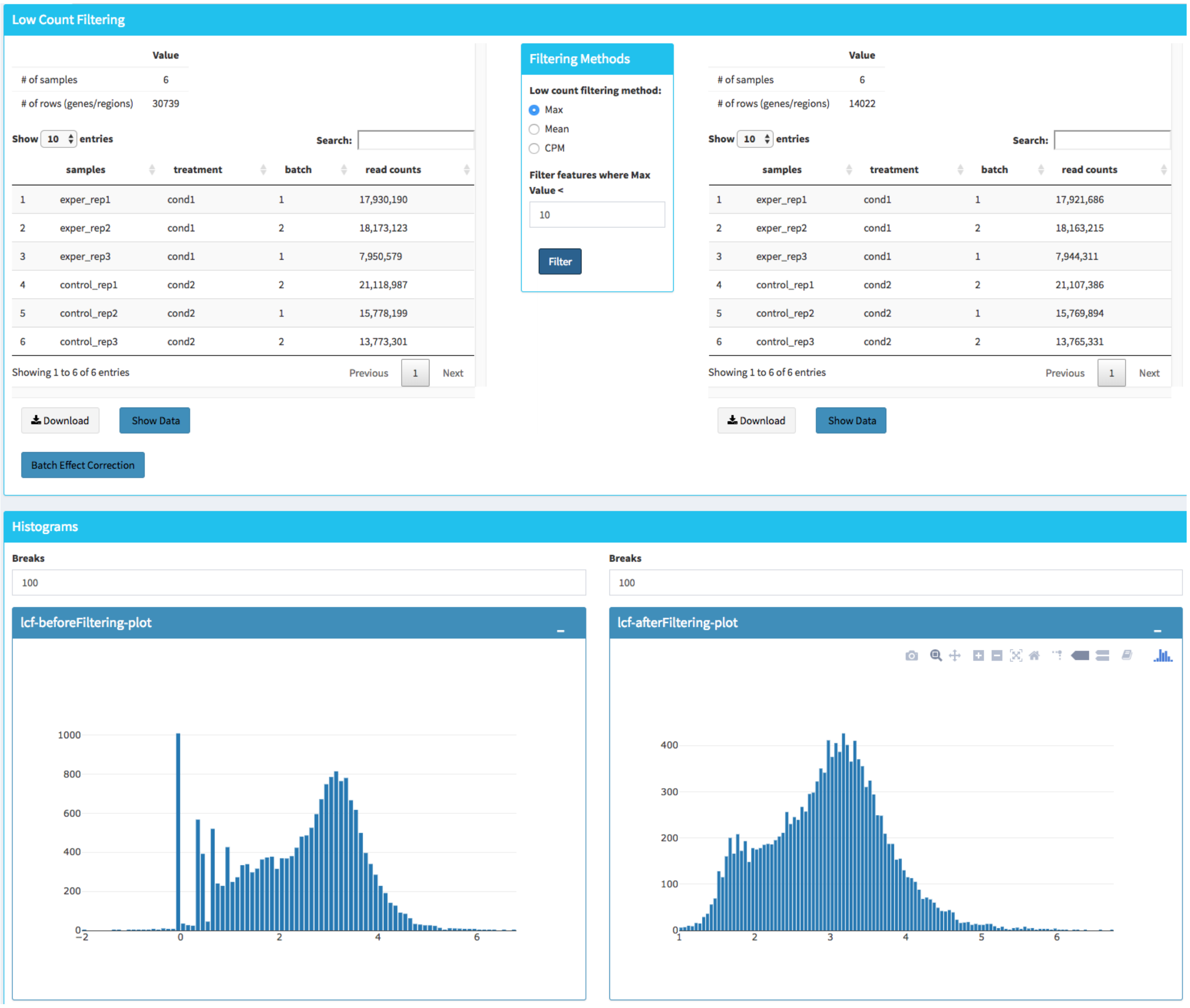
After loading the gene quantification file, and if specified the metadata file containing your batch correction fields, you then have the option to filter low counts and conduct batch effect correction prior to your analysis. Alternatively, you may skip these steps and directly continue with differential expression analysis or view quality control (QC) information of your dataset.

1.2 Low Count Filtering

In this section, you can simultaneously visualize the changes of your dataset while filtering out the low count genes. Choose your filtration criteria from **Filtering Methods** box which is located just center of the screen. Three methods are available to be used:

- **Max:** Filters out genes where maximum count for each gene across all samples are less than defined threshold.
- **Mean:** Filters out genes where mean count for each gene are less than defined threshold.
- **CPM:** First, counts per million (CPM) is calculated as the raw counts divided by the library sizes and multiplied by one million. Then it filters out genes where at least defined number of samples is less than defined CPM threshold.

After selection of filtering methods and entering threshold value, you can proceed by clicking **Filter** button which is located just bottom part of the **Filtering Methods** box. On the right part of the screen, your filtered dataset will be visualized for comparison as shown at figure below.


Low Count Filtering
Show 10 entries
Search:
Previous
1
Next
Download
Show Data
Batch Effect Correction
Filtering Methods
Filter
Histograms
lcf-beforeFiltering-plot
lcf-afterFiltering-plot
1000
800
600
400
200
0
−2
0
2
4
6
400
300
200
100
0
1
2
3
4
5
6

	Value
# of samples	6
# of rows (genes/regions)	30739

	samples	treatment	batch	read counts
1	exper_rep1	cond1	1	17,930,190
2	exper_rep2	cond1	2	18,173,123
3	exper_rep3	cond1	1	7,950,579
4	control_rep1	cond2	2	21,118,987
5	control_rep2	cond2	1	15,778,199
6	control_rep3	cond2	2	13,773,301

Showing 1 to 6 of 6 entries

Low count filtering method:
● Max
○ Mean
○ CPM

Filter features where Max Value <
10

	Value
# of samples	6
# of rows (genes/regions)	14022

	samples	treatment	batch	read counts
1	exper_rep1	cond1	1	17,921,686
2	exper_rep2	cond1	2	18,163,215
3	exper_rep3	cond1	1	7,944,311
4	control_rep1	cond2	2	21,107,386
5	control_rep2	cond2	1	15,769,894
6	control_rep3	cond2	2	13,765,331

Showing 1 to 6 of 6 entries

Breaks
100

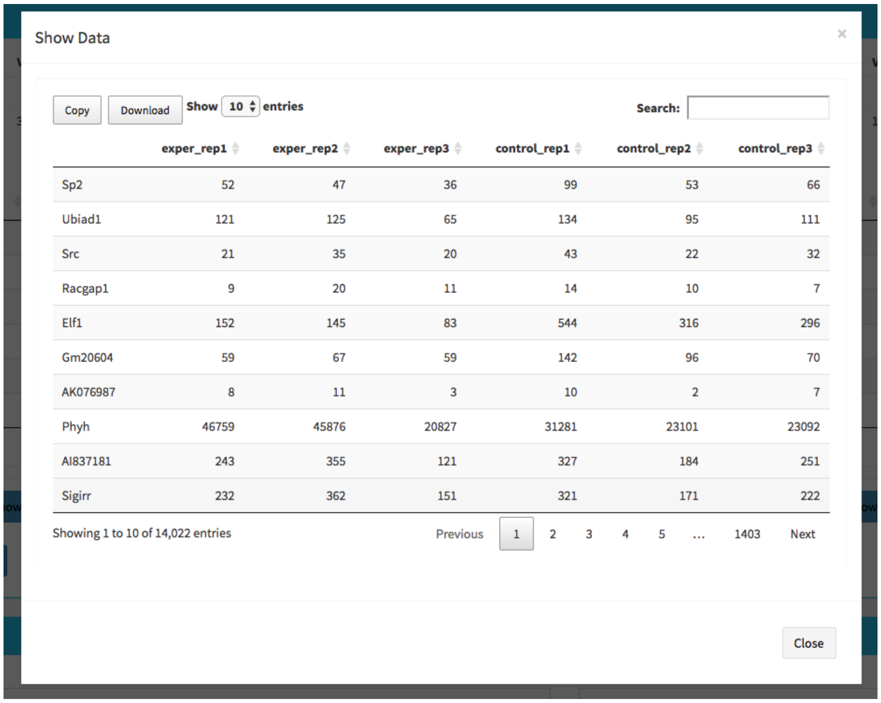
Breaks
100

You can easily compare following features, before and after filtering:

- Number of genes/regions.
- Read counts for each sample.

- Overall histogram of the dataset.
- gene/region vs samples data

Important: To investigate the gene/region vs samples data in detail as shown at below, you may click the **Show Data** button, located bottom part of the data tables. Alternatively, you may download all filtered data by clicking **Download** button which located next to **Show Data** button.



	exper_rep1	exper_rep2	exper_rep3	control_rep1	control_rep2	control_rep3
Sp2	52	47	36	99	53	66
Ubiad1	121	125	65	134	95	111
Src	21	35	20	43	22	32
Racgap1	9	20	11	14	10	7
Elf1	152	145	83	544	316	296
Gm20604	59	67	59	142	96	70
AK076987	8	11	3	10	2	7
Phyh	46759	45876	20827	31281	23101	23092
AI837181	243	355	121	327	184	251
Sigirr	232	362	151	321	171	222

Showing 1 to 10 of 14,022 entries

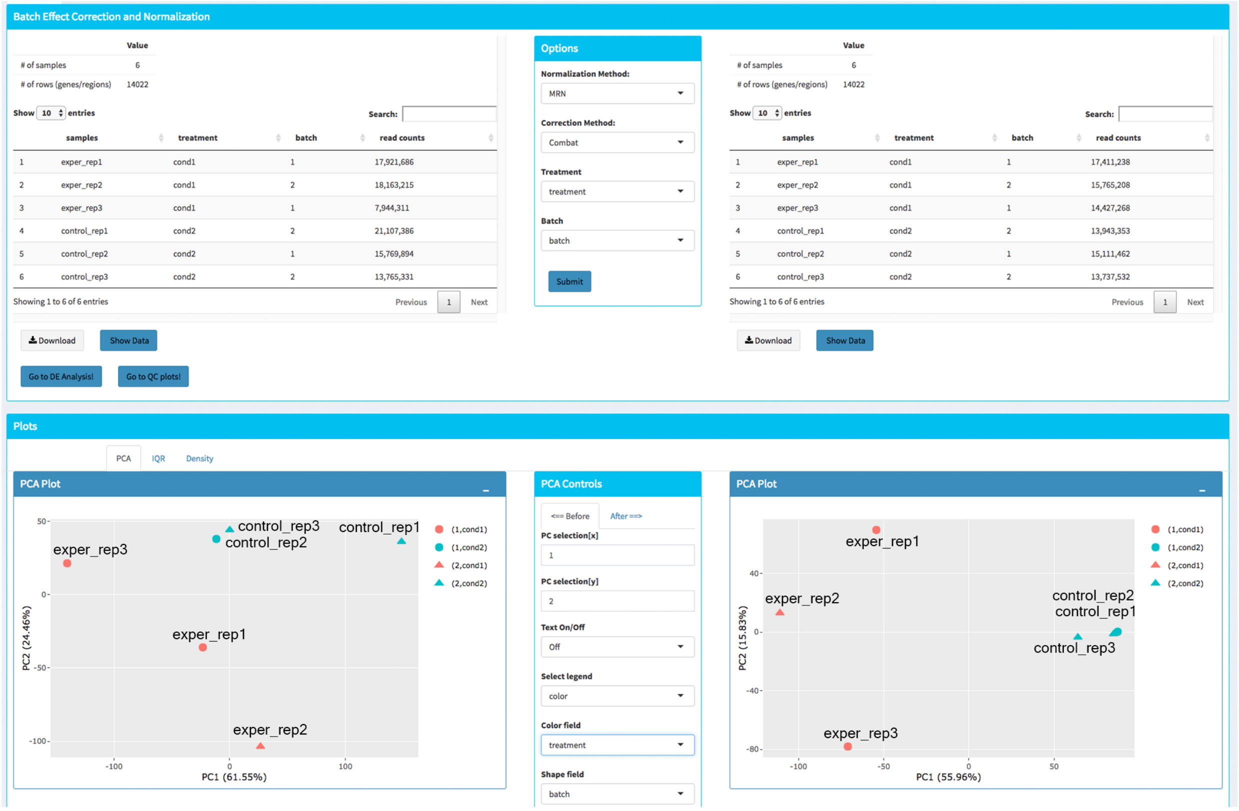
Afterwards, you may continue your analysis with **Batch Effect Correction** or directly jump to differential expression analysis or view quality control (QC) information of your dataset.

1.3 Batch Effect Correction and Normalization

If specified metadata file containing your batch correction fields, then you have the option to conduct batch effect correction prior to your analysis. By adjusting parameters of **Options** box, you can investigate your character of your dataset. These parameters of the options box are explained as following:

- **Normalization Method:** DEBrowser allows performing normalization prior the batch effect correction. You may choose your normalization method (among MRN (Median Ratio Normalization), TMM (Trimmed Mean of M-values), RLE (Relative Log Expression) and upperquartile), or skip this step by choosing **none** for this item. For our sample data, we are going to choose MRN normalization.
- **Correction Method:** DEBrowser uses ComBat (part of the SVA bioconductor package) or Harman to adjust for possible batch effect or conditional biases. For more information, you can visit following links for documentation: ComBat, Harman For our sample data, Combat correction was selected.
- **Treatment:** Please select the column that is specified in metadata file for comparison, such as cancer vs control. It is named *treatment* for our sample metadata.
- **Batch:** Please select the column name in metadata file which differentiate the batches. For example in our metadata, it is called *batch*.

Upon clicking submit button, comparison tables and plots will be created on the right part of the screen as shown below.



Batch Effect Correction and Normalization

	Value
# of samples	6
# of rows (genes/regions)	14022

Show 10 entries

	samples	treatment	batch	read counts
1	exper_rep1	cond1	1	17,921,686
2	exper_rep2	cond1	2	18,163,215
3	exper_rep3	cond1	1	7,944,311
4	control_rep1	cond2	2	21,107,386
5	control_rep2	cond2	1	15,769,894
6	control_rep3	cond2	2	13,765,331

Showing 1 to 6 of 6 entries

Options

Normalization Method: MRN

Correction Method: Combat

Treatment: treatment

Batch: batch

	Value
# of samples	6
# of rows (genes/regions)	14022

Show 10 entries

	samples	treatment	batch	read counts
1	exper_rep1	cond1	1	17,411,238
2	exper_rep2	cond1	2	15,765,208
3	exper_rep3	cond1	1	14,427,268
4	control_rep1	cond2	2	13,943,353
5	control_rep2	cond2	1	15,111,462
6	control_rep3	cond2	2	13,737,532

Showing 1 to 6 of 6 entries

PCA Controls

PC selection[x]: 1

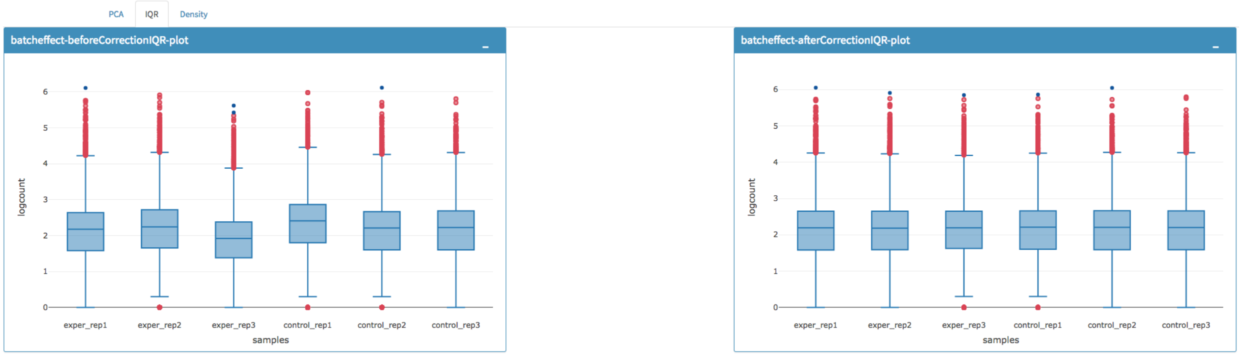
PC selection[y]: 2

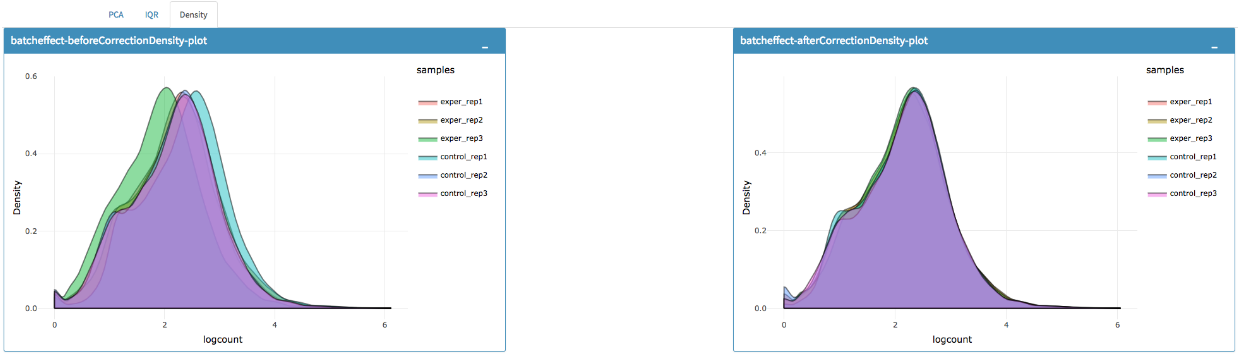
Text On/Off: Off

Select legend: color

Color field: treatment

Shape field: batch





You can investigate the changes on the data by comparing following features:

- Read counts for each sample.

- PCA, IQR and Density plot of the dataset.
- Gene/region vs samples data

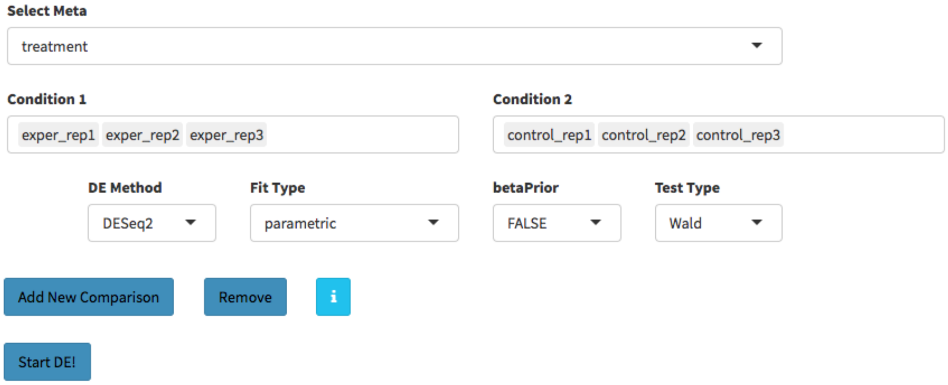
Tip: You can investigate the gene/region vs samples data in detail by clicking the **Show Data** button, or download all corrected data by clicking **Download** button.

Since we have completed **batch effect correction and normalization** step, we can continue with one of the following options: 'Go to DE Analysis' and, 'Go to QC plots!'. First option takes you to page where differential expression analyses are conducted with DESeq2, EdgeR or Limma. The second option, 'Go to QC plots!', takes you to a page where you can view quality control metrics of your data by PCA, All2All, Heatmap, Density, and IQR plots.

1.4 DE Analysis

The first option, 'Go to DE Analysis', takes you to the next step where differential expression analyses are conducted.

- **Sample Selection:** In order to run DE analysis, you first need to select the samples which will be compared. To do so, click on "Add New Comparison" button, and choose **Select Meta** box as **treatment** to simplify fill `Condition 1` and `Condition 2` based on the **treatment** column of the metadata as shown below.



Select Meta: treatment

Condition 1: exper_rep1 exper_rep2 exper_rep3

Condition 2: control_rep1 control_rep2 control_rep3

DE Method: DESeq2

Fit Type: parametric

betaPrior: FALSE

Test Type: Wald

If you need to remove samples from a condition, simply select the sample you wish to remove and hit the delete/backspace key. In case, you need to add a sample to a condition you can click on one of the condition text boxes to bring up a list of samples and then click on the sample you wish to add from the list and it will be added to the textbox for that comparison.

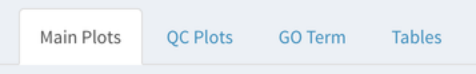
Tip: You can add multiple conditions to compare by clicking on "Add New Comparison" button, and view the results separately after DE analysis.

- **Method Selection:** Three DE methods are available for DEBrowser: DESeq2, EdgeR, and Limma. DESeq2 and EdgeR are designed to normalize count data from high-throughput sequencing assays such as RNA-Seq. On the other hand, Limma is a package to analyse of normalized or transformed data from microarray or RNA-Seq assays. We have selected DESeq2 for our test sample and showed the related results at below.

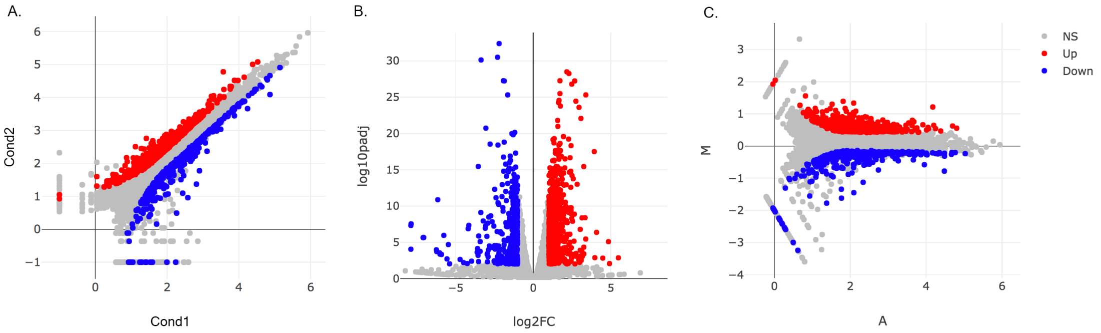
After clicking on the 'Submit!' button, DESeq2 will analyze your comparisons and store the results into separate data tables. It is important to note that the resulting data produced from DESeq is normalized. Upon finishing the DESeq analysis, a result table will appear which allows you to download the data by clicking "Download" button. To visualize the data with interactive plots please click on "Go to Main Plots!" button.

1.5 The Main Plots of DE Analysis

Upon finishing the DESeq analysis, please click on **Go to Main Plots!** button which will open **Main Plots** tab where you will be able to view the interactive plots.

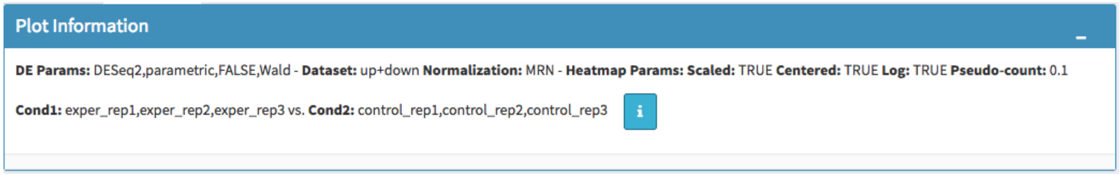


The page will load with **Scatter Plot**. You can switch to **Volcano Plot** and **MA Plot** by using **Plot Type** section at the left side of the menu. Since these plots are interactive, you can click to **zoom** button on the top of the graph and select the area you would like to zoom in by drawing a rectangle. Please see the plots at below:



A. Scatter plot, **B.** Volcano plot, **C.** MA plot

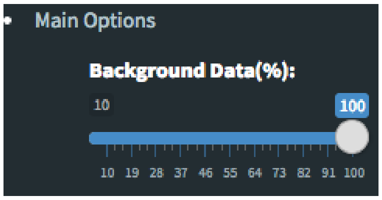
You can easily track the plotting parameters by checking **Plot Information** box as shown at below. Selected DE parameters, chosen dataset, compared conditions, and normalization method are listed. Additionally, heatmap parameters (scaled, centered, log, pseudo-count) could be simply followed by this info box.



DE Params: DESeq2,parametric,FALSE,Wald - **Dataset:** up+down **Normalization:** MRN - **Heatmap Params: Scaled:** TRUE **Centered:** TRUE **Log:** TRUE **Pseudo-count:** 0.1

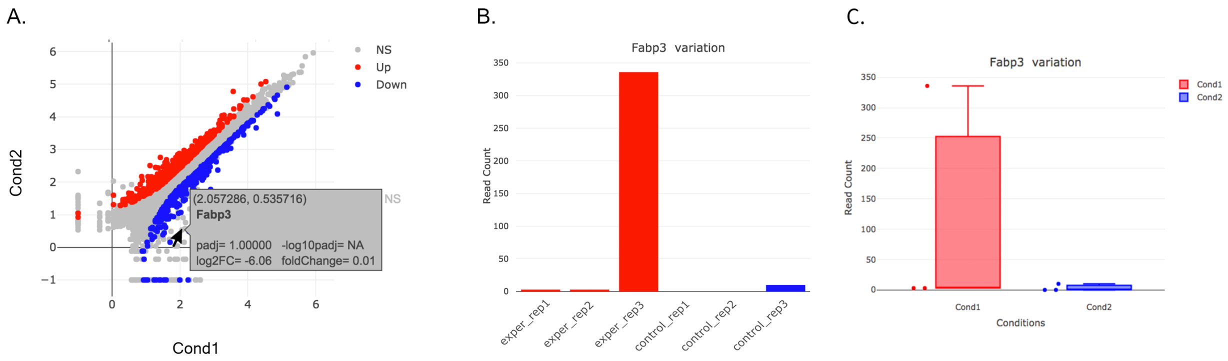
Cond1: exper_rep1,exper_rep2,exper_rep3 vs. **Cond2:** control_rep1,control_rep2,control_rep3

Tip: Please keep in mind that to increase the performance of the generating graph, by default 10% of non-significant(NS) genes are used to generate plots. You might show all NS genes by please click **Main Options** button and change Background Data(%) to 100% on the left sidebar.



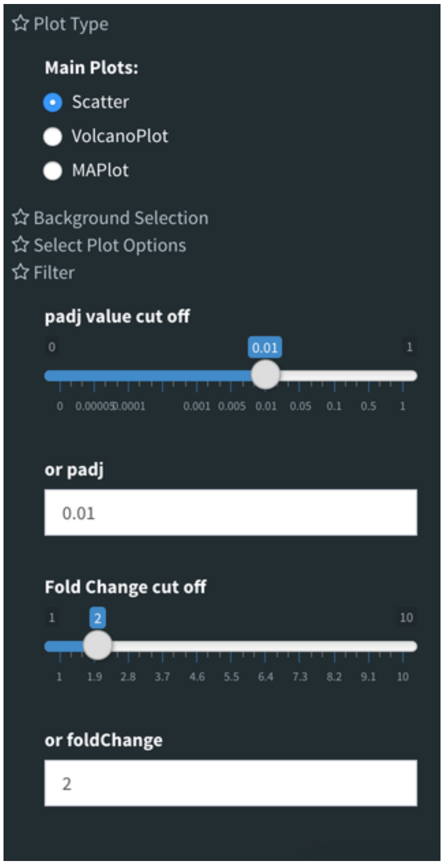
Background Data(%): 100

You can hover over the scatterplot points to display more information about the point selected. A few bargraphs will be generated for the user to view as soon as a scatterplot point is hovered over.



A. Hover on Fabp3 gene, **B.** Read Counts vs Samples, **C.** Read Counts vs Conditions

You also have a wide array of options when it comes to fold change cut-off levels, p-adjusted (padj) cut-off values, which comparison set to use, and dataset of genes to analyze.



☆ Plot Type

Main Plots:

● Scatter
○ VolcanoPlot
○ MAPlot

☆ Background Selection
☆ Select Plot Options
☆ Filter

padj value cut off

0 — 0.01 — 1

or padj

0.01

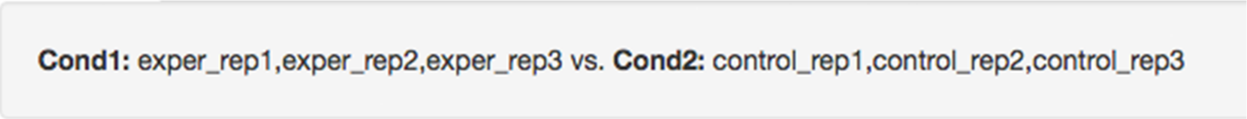
Fold Change cut off

1 — 2 — 10

or foldChange

2

Tip: It is important to note that when conducting multiple comparisons, the comparisons are labeled based on the order that they are input. If you don't remember which samples are in your current comparison you can always view the samples in each condition at the top of the main plots.

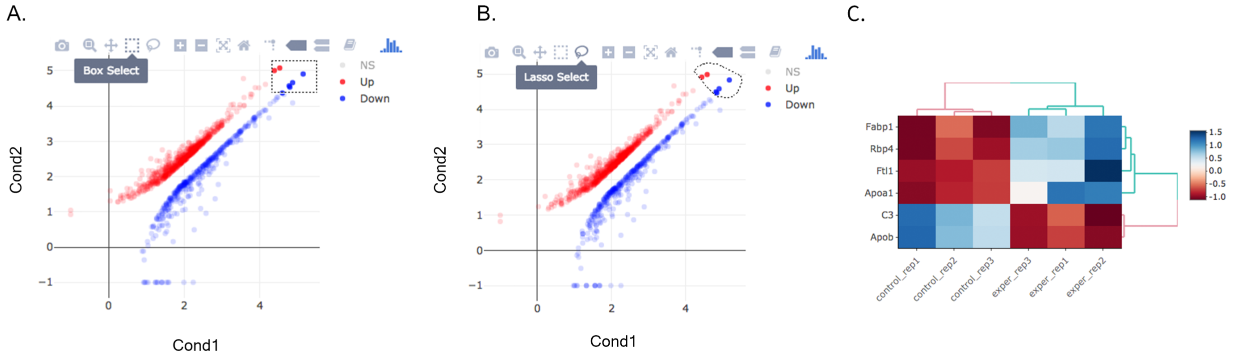


Cond1: exper_rep1,exper_rep2,exper_rep3 vs. **Cond2:** control_rep1,control_rep2,control_rep3

After DE analysis, you can always download the results in CSV format by clicking the **Download Data** button located under the **Data Options**. You can also download the plot or graphs by clicking on the **download** button at top of each plot or graph.

1.6 The Heatmap of DE Analysis

Once you've selected a specific region on Main Plots (Scatter, Volcano or MA plot), a new heatmap of the selected area will appear just next to your plot. If you want to hide some groups (such as Up, Down or NS based on DE analysis), just click on the group label on the top right part of the figure. In this way, you can select a specific part of the genes by **lasso select** or **box select** tools that includes only **Up** or **Down** Regulated genes. As soon as you completed your selection, heatmap will be created simultaneously. Please find details about heatmaps on Heatmaps section.



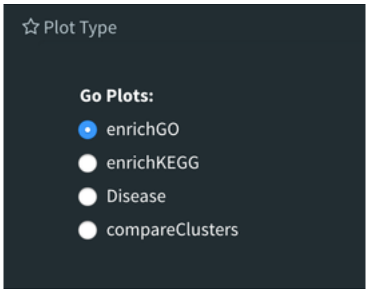
A. Box Selection, **B.** Lasso Selection, **C.** Created heatmap based on selection

---

Tip: We strongly recommend normalization before plotting heatmaps. To normalize, please change the parameters that are located under: **Data options -> Normalization Methods** and select the method from the dropdown box.

1.7 GO Term Plots

The next tab, 'GO Term', takes you to the ontology comparison portion of DEBrowser. From here you can select the standard dataset options such as p-adjust value, fold change cut off value, which comparison set to use, and which dataset to use on the left menu. In addition to these parameters, you also can choose from the 4 different ontology plot options: 'enrichGO', 'enrichKEGG', 'Disease', and 'compareCluster'. Selecting one of these plot options queries their specific databases with your current DESeq results.



Go Plots:
● enrichGO
○ enrichKEGG
○ Disease
○ compareClusters

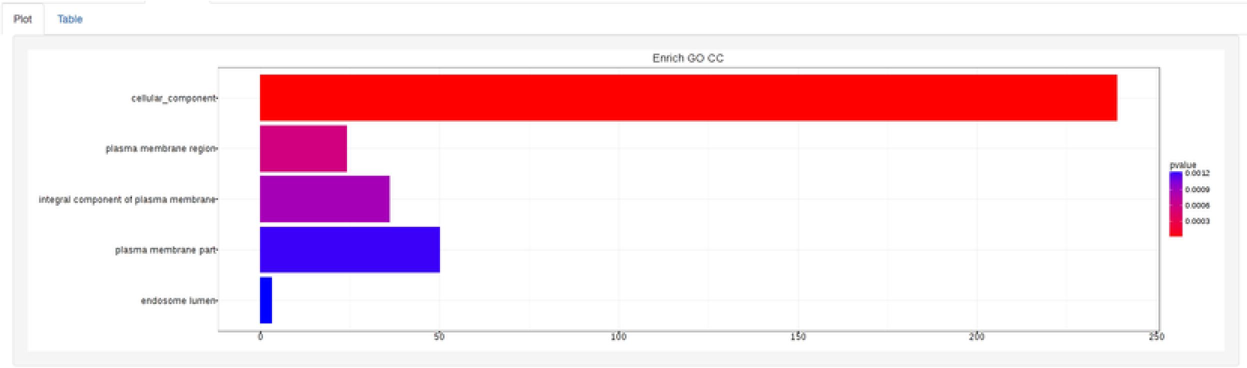
Your GO plots include:

- enrichGO - use enriched GO terms

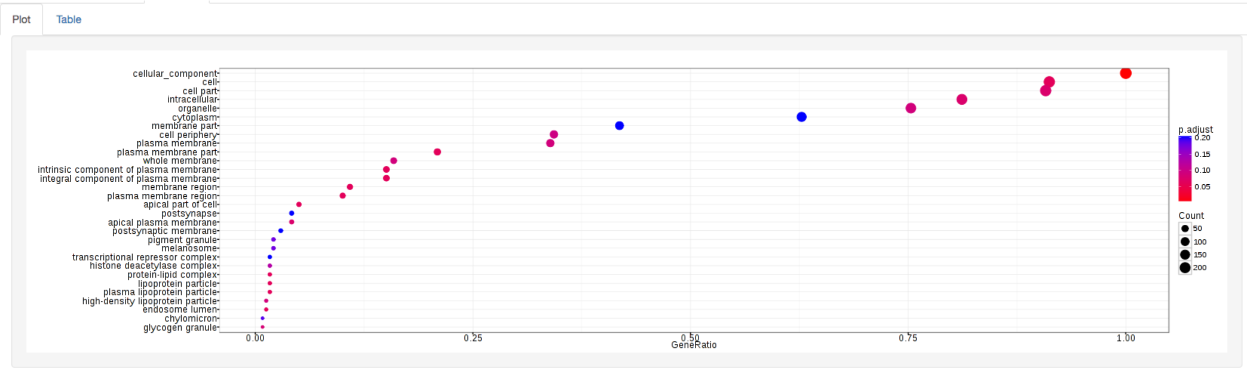
- enrichKEGG - use enriched KEGG terms
- Disease - enriched for diseases
- compareClusters - comparison of your clustered data

The types of plots you will be able to generate include:

Summary plot:


Plot
Table
Enrich GO CC
cellular_component
plasma membrane region
integral component of plasma membrane
plasma membrane part
endosome lumen
0
50
100
150
200
250
pvalue
0.0012
0.0009
0.0006
0.0003

GOdotplot:


Plot
Table
cellular_component
cell
cell part
intracellular
organelle
cytoplasm
membrane part
cell periphery
plasma membrane
plasma membrane part
whole membrane
intrinsic component of plasma membrane
integral component of plasma membrane
membrane region
plasma membrane region
apical part of cell
postsynapse
apical plasma membrane
postsynaptic membrane
pigment granule
melanosome
transcriptional repressor complex
histone deacetylase complex
protein-lipid complex
lipoprotein particle
plasma lipoprotein particle
high-density lipoprotein particle
endosome lumen
chylomicron
glycogen granule
0.00
0.25
0.50
0.75
1.00
GeneRatio
p.adjust
0.20
0.15
0.10
0.05
Count
50
100
150
200

Changing the type of ontology to use will also produce custom parameters for that specific ontology at the bottom of the left option panel.

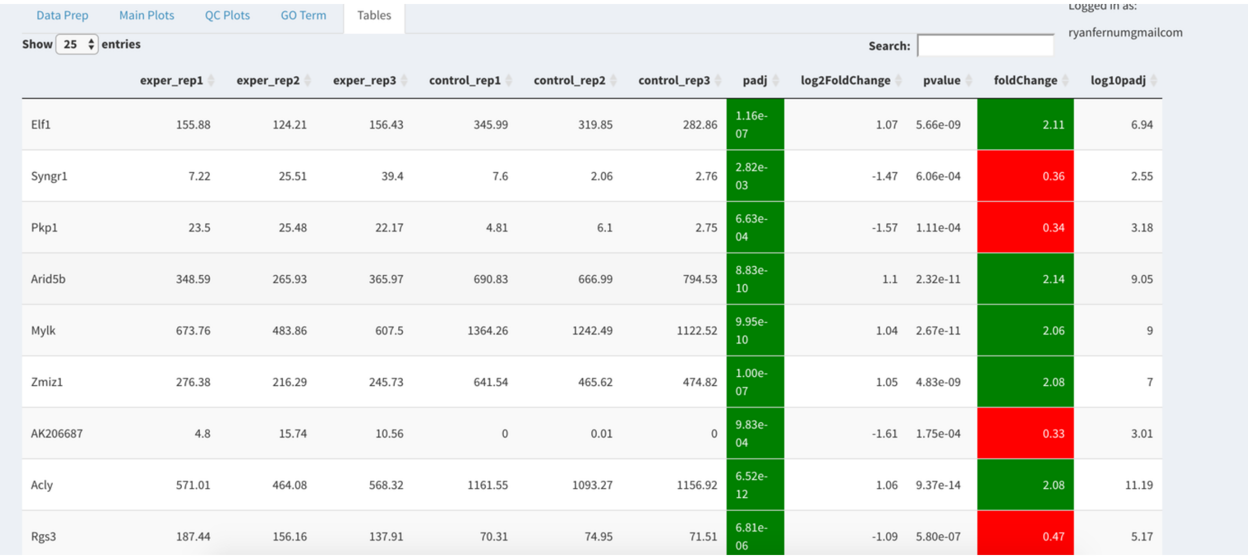
Once you have adjusted all of your parameters, you may hit the submit button in the top right and then wait for the results to show on screen!

1.8 Data Tables

The last tab at the top of the screen displays various different data tables. These datatables include:

- All Detected
- Up Regulated
- Down Regulated
- Up+down Regulated
- Selected scatterplot points
- Most varied genes

- Comparison differences



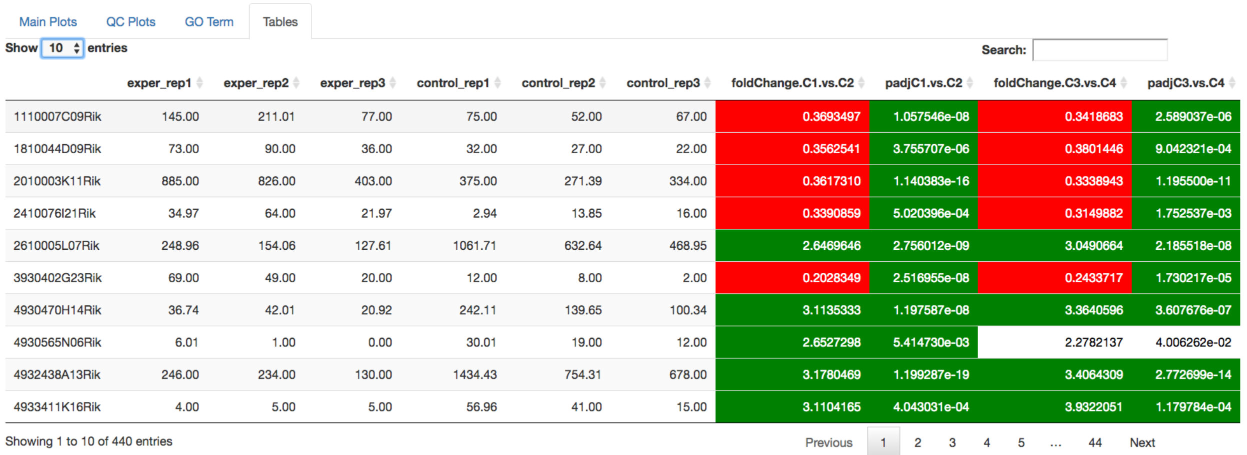
	exper_rep1	exper_rep2	exper_rep3	control_rep1	control_rep2	control_rep3	padj	log2FoldChange	pvalue	foldChange	log10padj
Elf1	155.88	124.21	156.43	345.99	319.85	282.86	1.16e-07	1.07	5.66e-09	2.11	6.94
Syngr1	7.22	25.51	39.4	7.6	2.06	2.76	2.82e-03	-1.47	6.06e-04	0.36	2.55
Pkp1	23.5	25.48	22.17	4.81	6.1	2.75	6.63e-04	-1.57	1.11e-04	0.34	3.18
Arid5b	348.59	265.93	365.97	690.83	666.99	794.53	8.83e-10	1.1	2.32e-11	2.14	9.05
Mylk	673.76	483.86	607.5	1364.26	1242.49	1122.52	9.95e-10	1.04	2.67e-11	2.06	9
Zmiz1	276.38	216.29	245.73	641.54	465.62	474.82	1.00e-07	1.05	4.83e-09	2.08	7
AK206687	4.8	15.74	10.56	0	0.01	0	9.83e-04	-1.61	1.75e-04	0.33	3.01
Acly	571.01	464.08	568.32	1161.55	1093.27	1156.92	6.52e-12	1.06	9.37e-14	2.08	11.19
Rgs3	187.44	156.16	137.91	70.31	74.95	71.51	6.81e-06	-1.09	5.80e-07	0.47	5.17

All of the tables tables, except the Comparisons table, contain the following information:

- ID - The specific gene ID
- Sample Names - The names of the samples given and they're corresponding tmm normalized counts
- Conditions - The log averaged values
- padj - padjusted value
- log2FoldChange - The Log2 fold change
- foldChange - The fold change
- log10padj - The log 10 padjusted value

The Comparisons table generates values based on the number of comparisons you have conducted. For each pairwise comparison, these values will be generated:

- Values for each sample used
- foldChange of comparison A vs B
- pvalue of comparison A vs B
- padj value of comparison A vs B



	exper_rep1	exper_rep2	exper_rep3	control_rep1	control_rep2	control_rep3	foldChange.C1.vs.C2	padjC1.vs.C2	foldChange.C3.vs.C4	padjC3.vs.C4
1110007C09Rik	145.00	211.01	77.00	75.00	52.00	67.00	0.3693497	1.057546e-08	0.3418683	2.589037e-06
1810044D09Rik	73.00	90.00	36.00	32.00	27.00	22.00	0.3562541	3.755707e-06	0.3801446	9.042321e-04
2010003K11Rik	885.00	826.00	403.00	375.00	271.39	334.00	0.3617310	1.140383e-16	0.3338943	1.195500e-11
2410076I21Rik	34.97	64.00	21.97	2.94	13.85	16.00	0.3390859	5.020396e-04	0.3149882	1.752537e-03
2610005L07Rik	248.96	154.06	127.61	1061.71	632.64	468.95	2.6469646	2.756012e-09	3.0490664	2.185518e-08
3930402G23Rik	69.00	49.00	20.00	12.00	8.00	2.00	0.2028349	2.516955e-08	0.2433717	1.730217e-05
4930470H14Rik	36.74	42.01	20.92	242.11	139.65	100.34	3.1135333	1.197587e-08	3.3640596	3.607676e-07
4930565N06Rik	6.01	1.00	0.00	30.01	19.00	12.00	2.6527298	5.414730e-03	2.2782137	4.006262e-02
4932438A13Rik	246.00	234.00	130.00	1434.43	754.31	678.00	3.1780469	1.199287e-19	3.4064309	2.772699e-14
4933411K16Rik	4.00	5.00	5.00	56.96	41.00	15.00	3.1104165	4.043031e-04	3.9322051	1.179784e-04

Showing 1 to 10 of 440 entries

You can further customize and filter each specific table a multitude of ways. For unique table or dataset options, select the type of table dataset you would like to customize on the left panel under 'Choose a dataset' to view it's additional options. All of the tables have a built in search function at the top right of the table and you can further sort the table by column by clicking on the column header you wish to sort by. The 'Search' box on the left panel allows for multiple searches via a comma-separated list. You can additionally use regex terms such as "^al" or "*lm" for even more advanced searching. This search will be applied to wherever you are within DEBrowser, including both the plots and the tables.

Tip: If you enter more than three lines of genes, search tool will automatically match the beginning and end of the search phrases. Otherwise it will find matched substrings in the gene list.

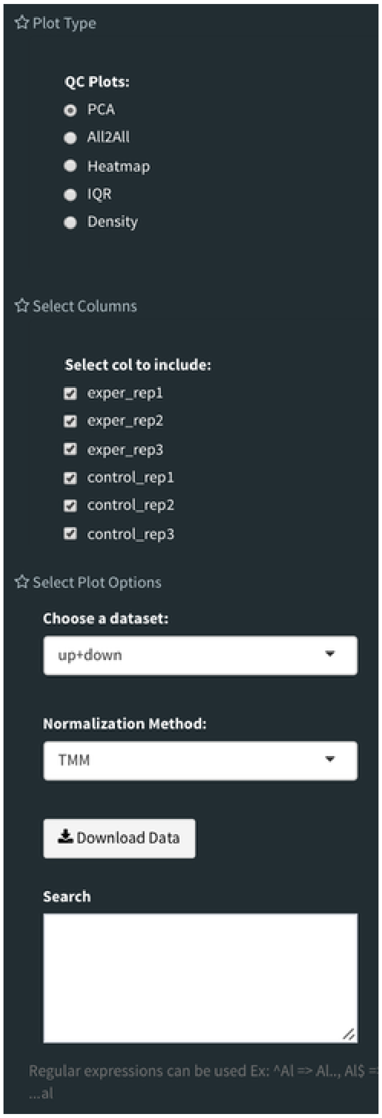
You can also view specific tables of your input data for each type of dataset available and search for a specific geneset by inputting a comma-separated list of genes or regex terms to search for in the search box within the left panel. To view these tables, you must select the tab labeled 'Tables' as well as the dataset from the dropdown menu on the left panel.

Tip: If you ever want to change your parameters, or even add a new set of comparisons, you can always return to the *Data Prep* tab to change and resubmit your data.

1.9 Quality Control Plots

Selecting the 'QC Plots' tab will take you to the quality control plots section. The page opens with a Principal Component Analysis (PCA) plot and users can also view a All2All, heatmap, IQR, and density by choosing **Plot Type** in the left panel. Here the dataset being used in the plots, depends on the parameters you selected in the left panel. Therefore, you are able to adjust the size of the plots under 'width' and 'height' as well as alter a variety of other parameters to adjust the specific plot you're viewing.

The All2All plot displays the correlation between each sample, Heatmap shows a heatmap representation of your data, IQR displays a barplot displaying the IQR between samples, and Density will display an overlapping density graph for each sample. You also have the ability to select the type of clustering and distance method for the heatmap produced to further customize your quality control measures. Users also have the option to select which type of normalization methods they would like to use for these specific plotting analysis within the left menu.


Plot Type
Select Columns
Select Plot Options
Download Data
Search

QC Plots:
- ● PCA
- ○ All2All
- ○ Heatmap
- ○ IQR
- ○ Density

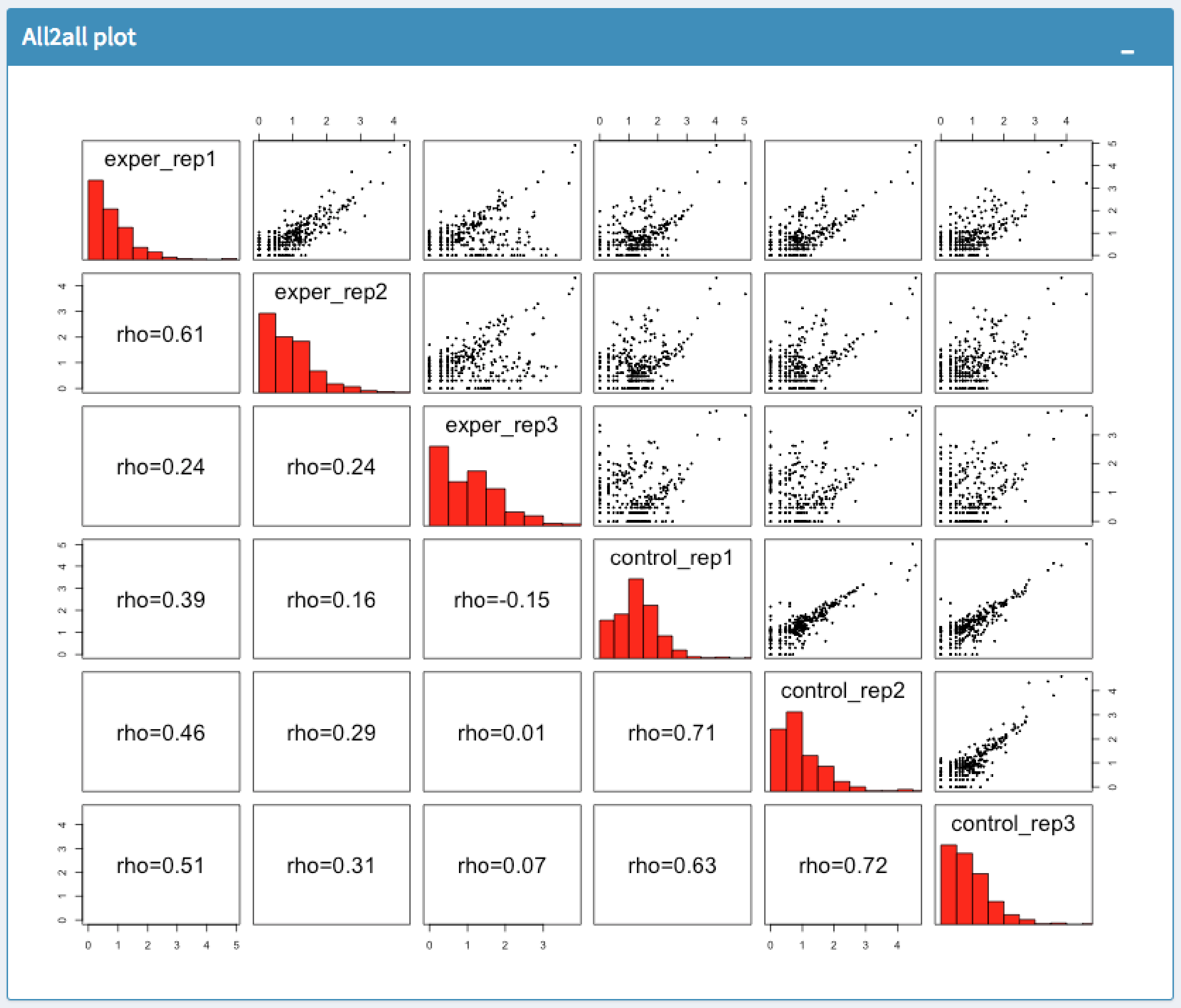
Select col to include:
- ☑ exper_rep1
- ☑ exper_rep2
- ☑ exper_rep3
- ☑ control_rep1
- ☑ control_rep2
- ☑ control_rep3

Choose a dataset: up+down

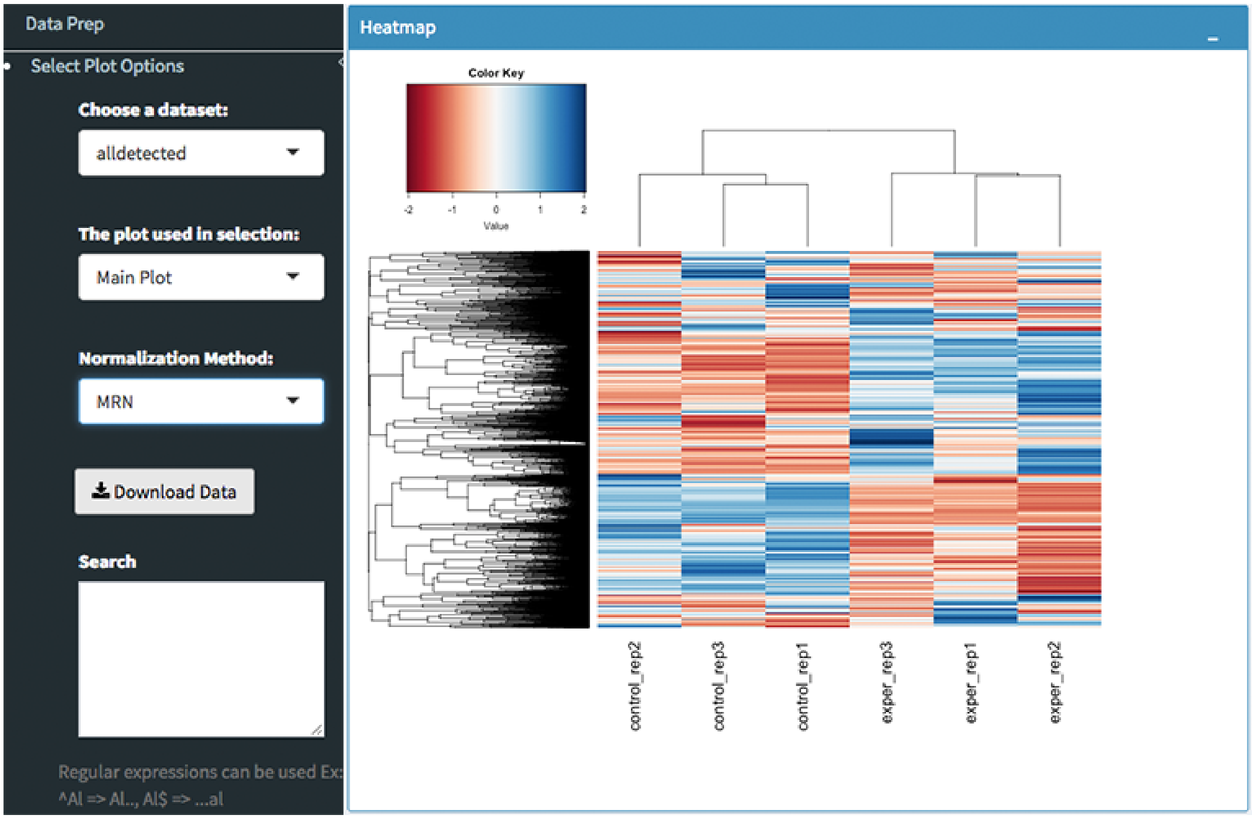
Normalization Method: TMM

Regular expressions can be used Ex: ^Al => Al.., Al$ =
...al

Ploting Options


All2all plot
exper_rep1
exper_rep2
exper_rep3
control_rep1
control_rep2
control_rep3
rho=0.61
rho=0.24
rho=0.24
rho=0.39
rho=0.16
rho=-0.15
rho=0.46
rho=0.29
rho=0.01
rho=0.71
rho=0.51
rho=0.31
rho=0.07
rho=0.63
rho=0.72

All2All Plot



Data Prep

Select Plot Options

Choose a dataset: alldetected

The plot used in selection: Main Plot

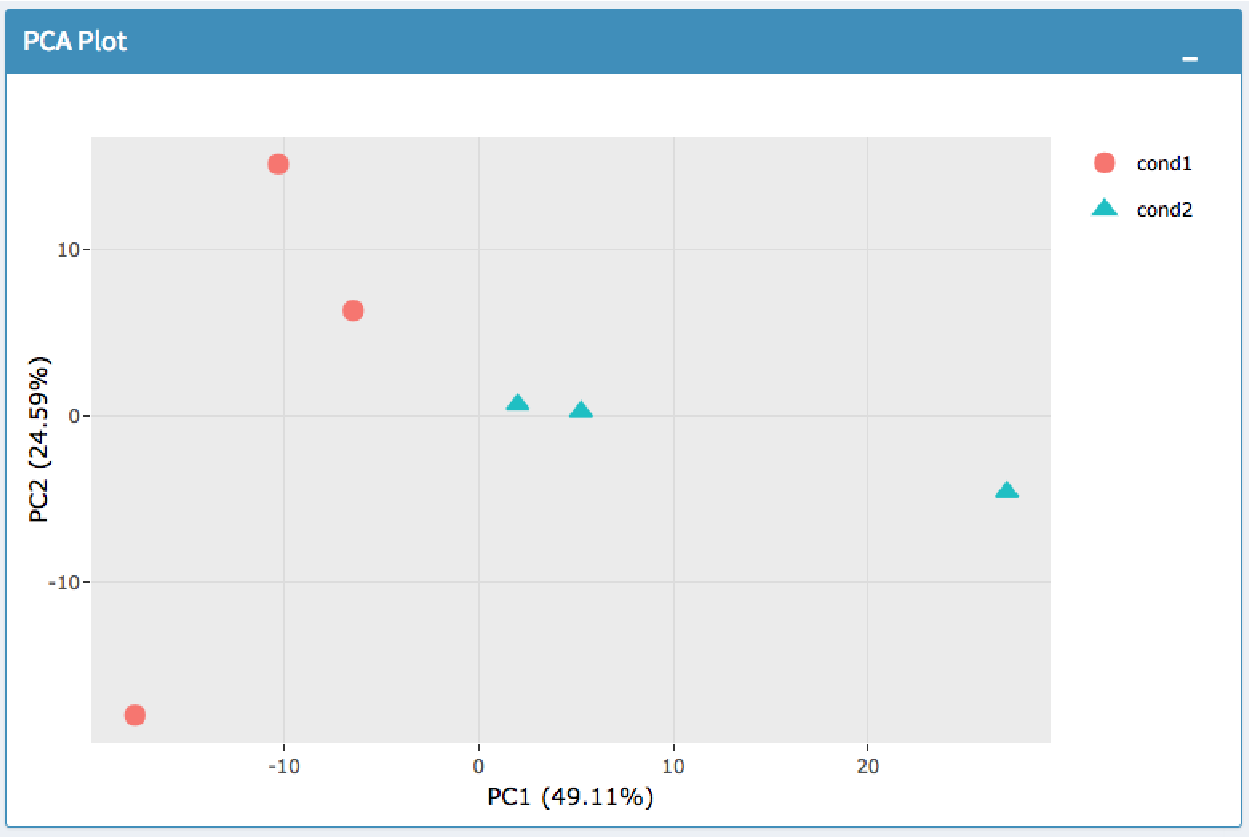
Normalization Method: MRN

Search

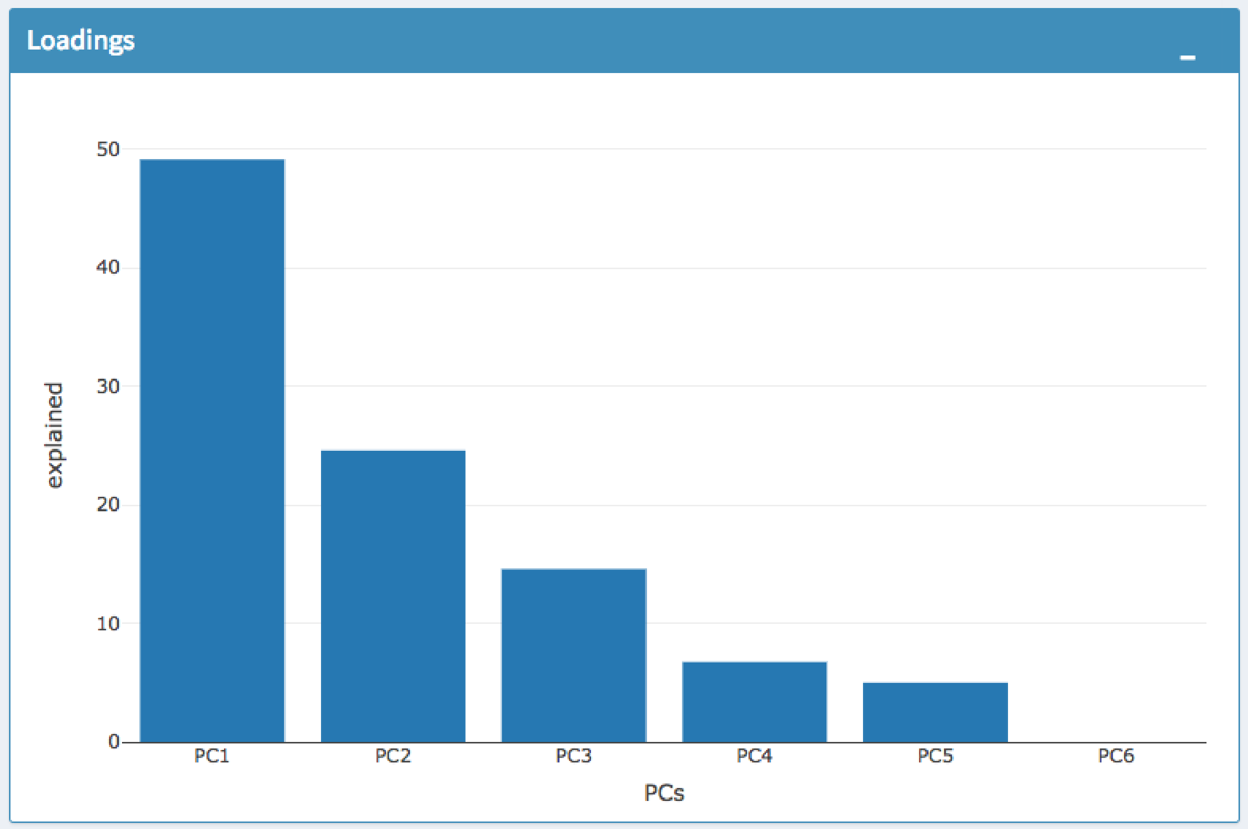
Regular expressions can be used Ex:
^Al => Al.., Al$ => ...al

Download Data
Heatmap
Color Key
-2
-1
0
1
2
Value
control_rep2
control_rep3
control_rep1
exper_rep3
exper_rep1
exper_rep2

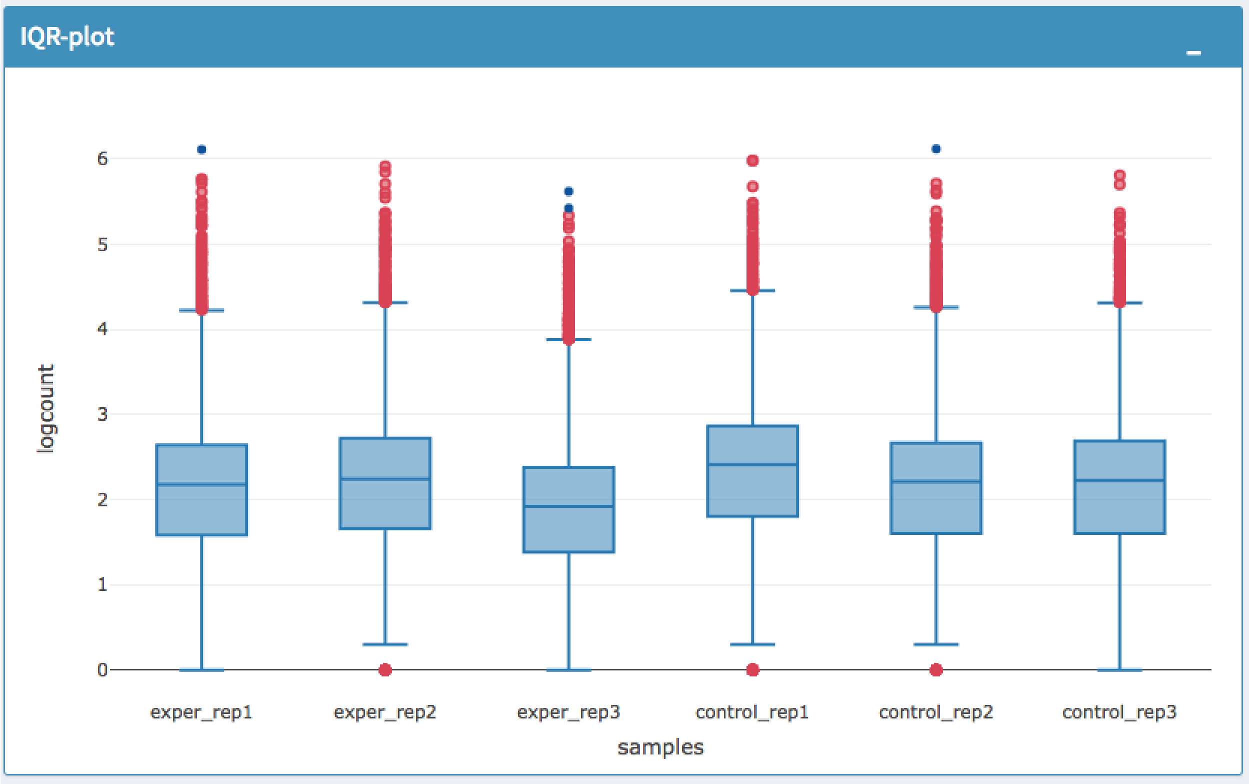
Heatmap Options to Normalize All Detected Data and Created Heatmap



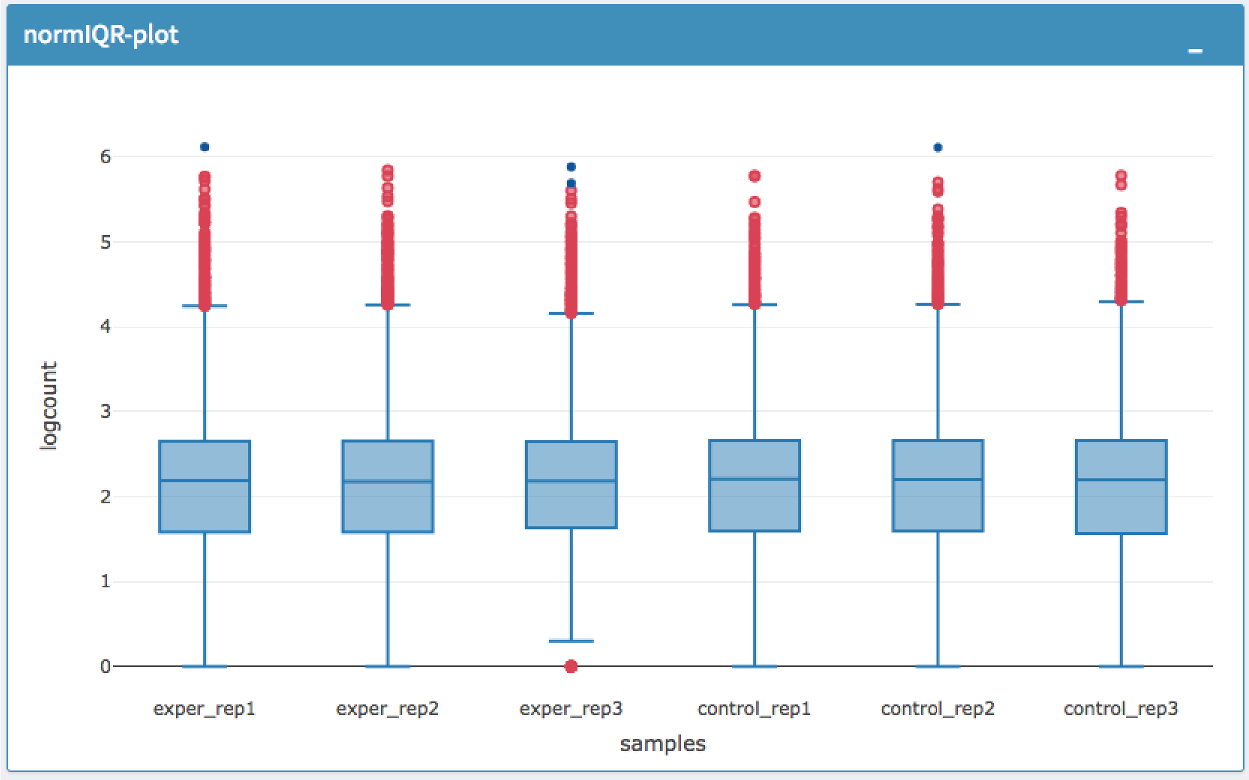
PCA Plot



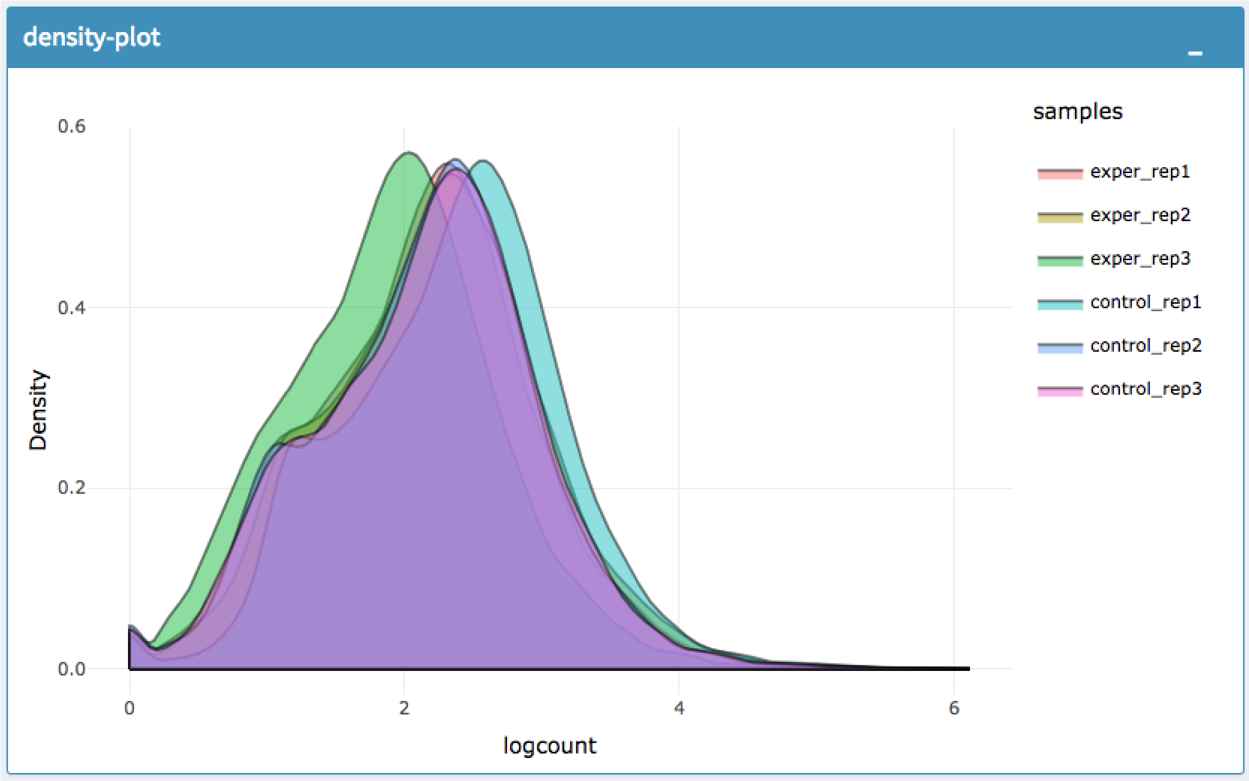
PCA Loadings



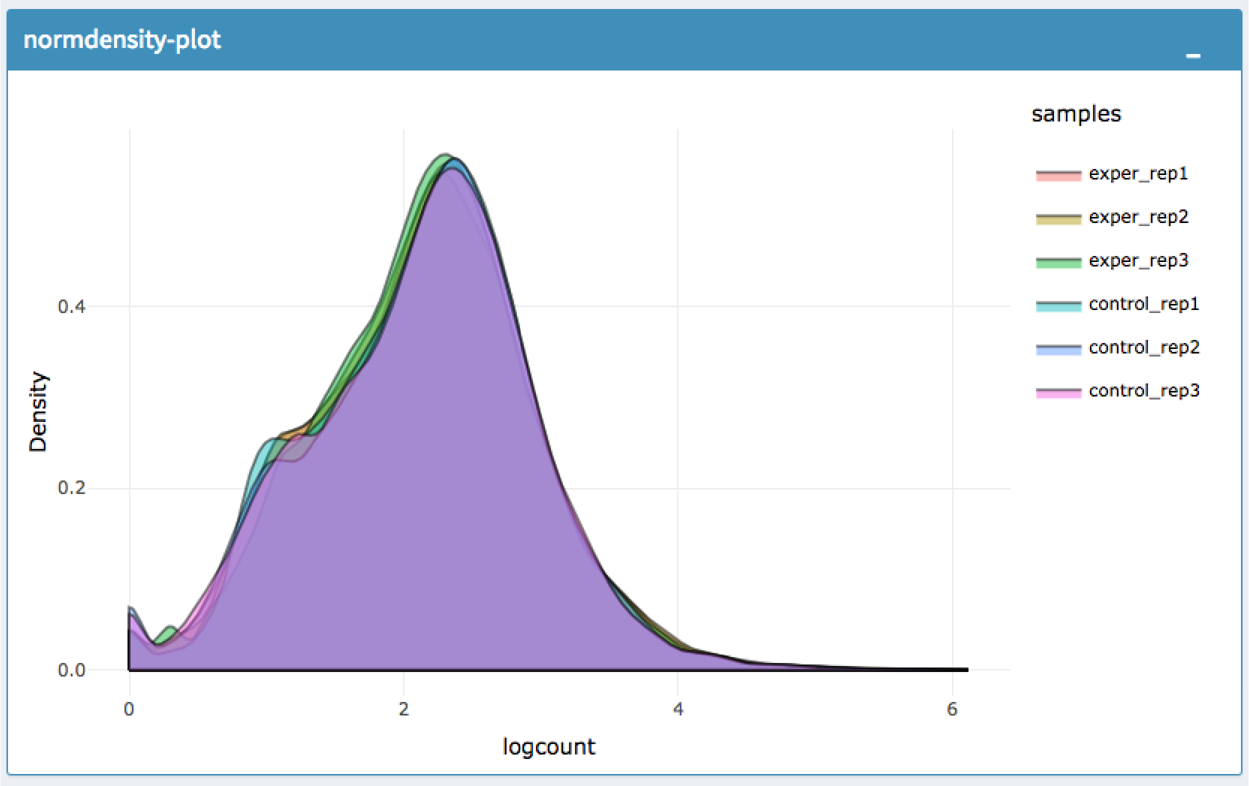
IQR Plot Before Normalization



IQR Plot After Normalization



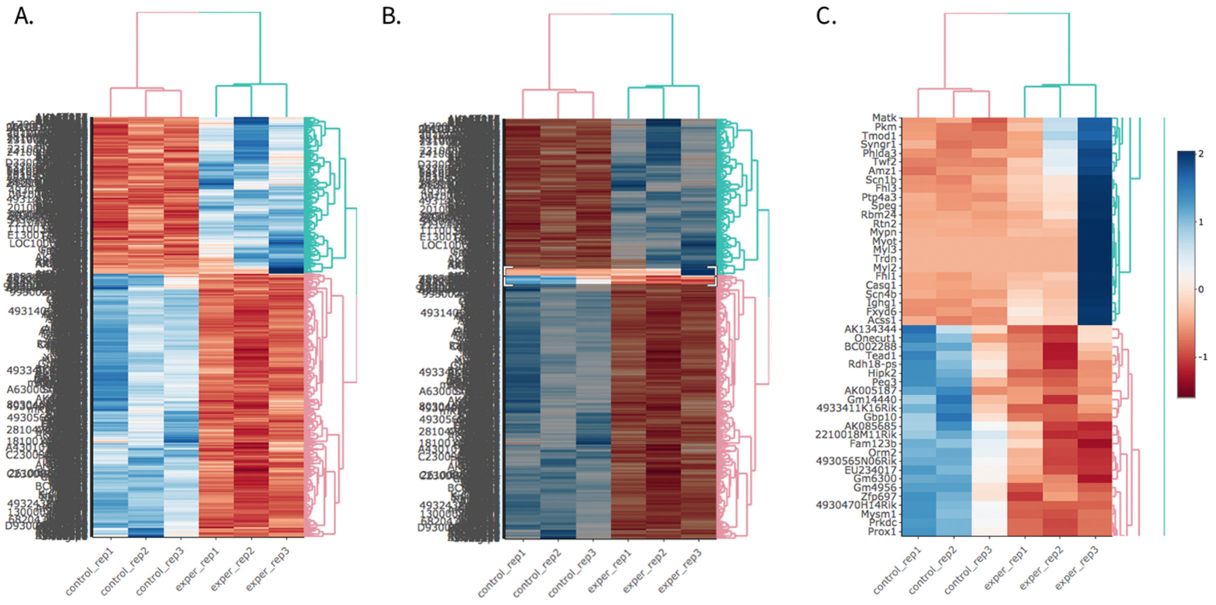
Density Plot Before Normalization



Density Plot After Normalization

Note: Each QC plot also has options to adjust the plot height and width, as well as a download button for a png output located above each plot.

For the Heatmap, you can also view an interactive session of the heatmap by selecting the 'Interactive' checkbox before submitting your heatmap request. Make sure that before selecting the interactive heatmap option that your dataset being used is 'Up+down'. Just like in the Main Plots, you can click and drag to create a selection. To select a specific portion of the heatmap, make sure to highlight the middle of the heatmap gene box in order to fully select a specific gene. This selection can be used later within the GO Term plots for specific queries on your selection. For find more details please click Heatmaps section.



A. Before Selection B. Selection of area with zoom tool C. Zoomed heatmap region which allows better viewing resolution.

1.10 Autoload Data via Hyperlink

DEBrowser also accepts TSV's via hyperlink by following conversion steps. First, using the API provided by Dolphin, we will convert TSV into an html represented TSV using this website:

```
https://dolphin.umassmed.edu/public/api/
```

The two parameters it accepts (and examples) are:

1. source=https://bioinfo.umassmed.edu/pub/debrowser/advanced_demo.tsv
2. format=JSON

Leaving you with a hyperlink for:

```
https://dolphin.umassmed.edu/public/api/?source=https://bioinfo.umassmed.edu/pub/
↪debrowser/advanced_demo.tsv&format=JSON
```

Next you will need to encode the url so you can pass it to the DEBrowser website. You can find multiple url encoders online, such as the one located at this link..

Encoding our URL will turn it into this:

```
http%3A%2F%2Fdolphin.umassmed.edu%2Fpublic%2Fapi%2F%3Fsource%3Dhttp%3A%2F%2Fbioinfo.
↪umassmed.edu%2Fpub%2Fdebrowser%2Fadvanced_demo.tsv%26format%3DJSON
```

Now this link can be used in DEBrowser as:

```
https://debrowser.umassmed.edu:443/debrowser/R/
```

It accepts two parameters:

```
1. jsonobject= http%3A%2F%2Fdolphin.umassmed.edu%2Fpublic%2Fapi%2F%3Fsource%3Dhttp%3A
↪%2F%2Fbioinfo.umassmed.edu%2Fpub%2Fdebrowser%2Fadvanced_demo.tsv%26format%3DJSON
2. title= no
```

The finished product of the link will look like this:

```
https://debrowser.umassmed.edu:443/debrowser/R/?jsonobject=https://dolphin.umassmed.
↪edu/public/api/?source=https://bioinfo.umassmed.edu/pub/debrowser/advanced_demo.tsv&
↪format=JSON&title=no
```

Inputting this URL into your browser will automatically load in that tsv to be analyzed by DEBrowser!

CHAPTER 2

Installation Guide

Before you start; you will have to install R and/or RStudio. You can install DEBrowser from bioconductor or from the source code. Install the required dependencies by running the following commands in R or RStudio. Please check *Operating System Dependencies* section, in case your operating system requires packages to be installed.

A.1 Bioconductor Installation (Recommended):

```
source("https://www.bioconductor.org/biocLite.R")
biocLite("debrowser")
```

A.2 Bioconductor Installation - Developer Version:

```
if (!requireNamespace("BiocManager", quietly=TRUE))
install.packages("BiocManager")
BiocManager::install("debrowser", version = "devel")
```

B. Installation instructions from source code:

```
install.packages("devtools") ## If you haven't installed devtools, you can easily␣
↪install it by using this command
library("devtools")
install_github("UMMS-Biocore/debrowser", build_vignettes = TRUE)
```

Alternatively, you can download the source code from here as a compressed format. Then you need to decompress and install with following command:

```
R CMD INSTALL debrowser-develop  ##where folder name is debrowser-develop
```

After debrowser installation, you can load and start DEBrowser by following commands:

```
library(debrowser)
startDEBrowser()
```

Once you run `startDEBrowser()` shiny will launch a web browser which is ready to use!

For more information about DEBrowser, please visit our **Quick-start Guide** section within documentation.

2.1 Operating System Dependencies

On Fedora/Red Hat/CentOS, these packages have to be installed:

```
openssl-devel, libxml2-devel, libcurl-devel, libpng-devel
```

On Ubuntu 18.04 LTS, you can install required packages by following command:

```
sudo apt-get install libcurl4-openssl-dev libssl-dev libv8-3.14-dev udunits-bin␣
↪libudunits2-* libxml2-dev
```

CHAPTER 3

DE Analysis

This guide contains a breif discription of DESeq2 used within the DEBrowser

3.1 Introduction

Differential gene expression analysis has become an increasingly popular tool in determining and viewing up and/or down experssed genes between two sets of samples. The goal of Differential gene expression analysis is to find genes or transcripts whose difference in expression, when accounting for the variance within condition, is higher than expected by chance. DESeq2 is an R package available via Bioconductor and is designed to normalize count data from high-throughput sequencing assays such as RNA-Seq and test for differential expression (Love et al. 2014). For more information on the DESeq2 algorithm, you can visit this website With multiple parameters such as padjust values, log fold changes, and plot styles, altering plots created with your DE data can be a hassle as well as time consuming. The Differential Expression Browser uses DESeq2, EdgeR, and Limma coupled with shiny to produce real-time changes within your plot queries and allows for interactive browsing of your DESeq results. In addition to DESeq analysis, DEBrowser also offers a variety of other plots and analysis tools to help visualize your data even further.

3.2 DESeq2

For the details please check the user guide. DESeq2 userguide

DESeq2 performs multiple steps in order to analyze the data you've provided for it. The first step is to indicate the condition that each column (experiment) in the table represent. You can group multiple samples into one condition column. DESeq2 will compute the probability that a gene is differentially expressed (DE) for ALL genes in the table. It outputs both a nominal and a multiple hypothesis corrected p-value (padj) using a negative binomial distribution.

3.3 Un-normalized counts

DESeq2 rquires count data as input obtained from RNA-Seq or another high-thorughput sequencing experiment in the form of matrix values. Here we convert un-integer values to integer to be able to run DESeq2. The matrix values should be un-normalized, since DESeq2 model internally corrects for library size. So, transformed or normalized values such as counts scaled by library size should not be used as input. Please use edgeR or limma for normalized counts.

3.4 Used parameters for DESeq2

- **fitType:** either "parametric", "local", or "mean" for the type of fitting of dispersions to the mean intensity. See estimateDispersions for description.
- **betaPrior:** whether or not to put a zero-mean normal prior on the non-intercept coefficients See nbinomWaldTest for description of the calculation of the beta prior. By default, the beta prior is used only for the Wald test, but can also be specified for the likelihood ratio test.
- **testType:** either "Wald" or "LRT", which will then use either Wald significance tests (defined by nbinomWaldTest), or the likelihood ratio test on the difference in deviance between a full and reduced model formula (defined by nbinomLRT)
- **rowsum.filter:** regions/genes/isoforms with total count (across all samples) below this value will be filtered out

3.5 EdgeR

For the details please check the user guide. EdgeR userguide.

3.6 Used parameters for EdgeR

- **Normalization:** Calculate normalization factors to scale the raw library sizes. Values can be "TMM","RLE","upperquartile","none".
- **Dispersion:** either a numeric vector of dispersions or a character string indicating that dispersions should be taken from the data object.
- **testType:** exactTest or glmLRT. exactTest: Computes p-values for differential abundance for each gene between two samples, conditioning on the total count for each gene. The counts in each group are assumed to follow a binomial distribution. glmLRT: Fits a negative binomial generalized log-linear model to the read counts for each gene and conducts genewise statistical tests.
- **rowsum.filter:** regions/genes/isoforms with total count (across all samples) below this value will be filtered out

3.7 Limma

For the details please check the user guide. Limma userguide.

Limma is a package to analyse of microarray or RNA-Seq data. If data is normalized with spike-in or any other scaling, tranforamtion or normalization method, Limma can be ideal. In that case, prefer limma rather than DESeq2 or EdgeR.

3.8 Used parameters for Limma

- **Normalization:** Calculate normalization factors to scale the raw library sizes. Values can be "TMM","RLE","upperquartile","none".
- **Fit Type:** fitting method; "ls" for least squares or "robust" for robust regression
- **Norm. Bet. Arrays:** Normalization Between Arrays; Normalizes expression intensities so that the intensities or log-ratios have similar distributions across a set of arrays.
- **rowsum.filter:** regions/genes/isoforms with total count (across all samples) below this value will be filtered out

3.9 ComBat

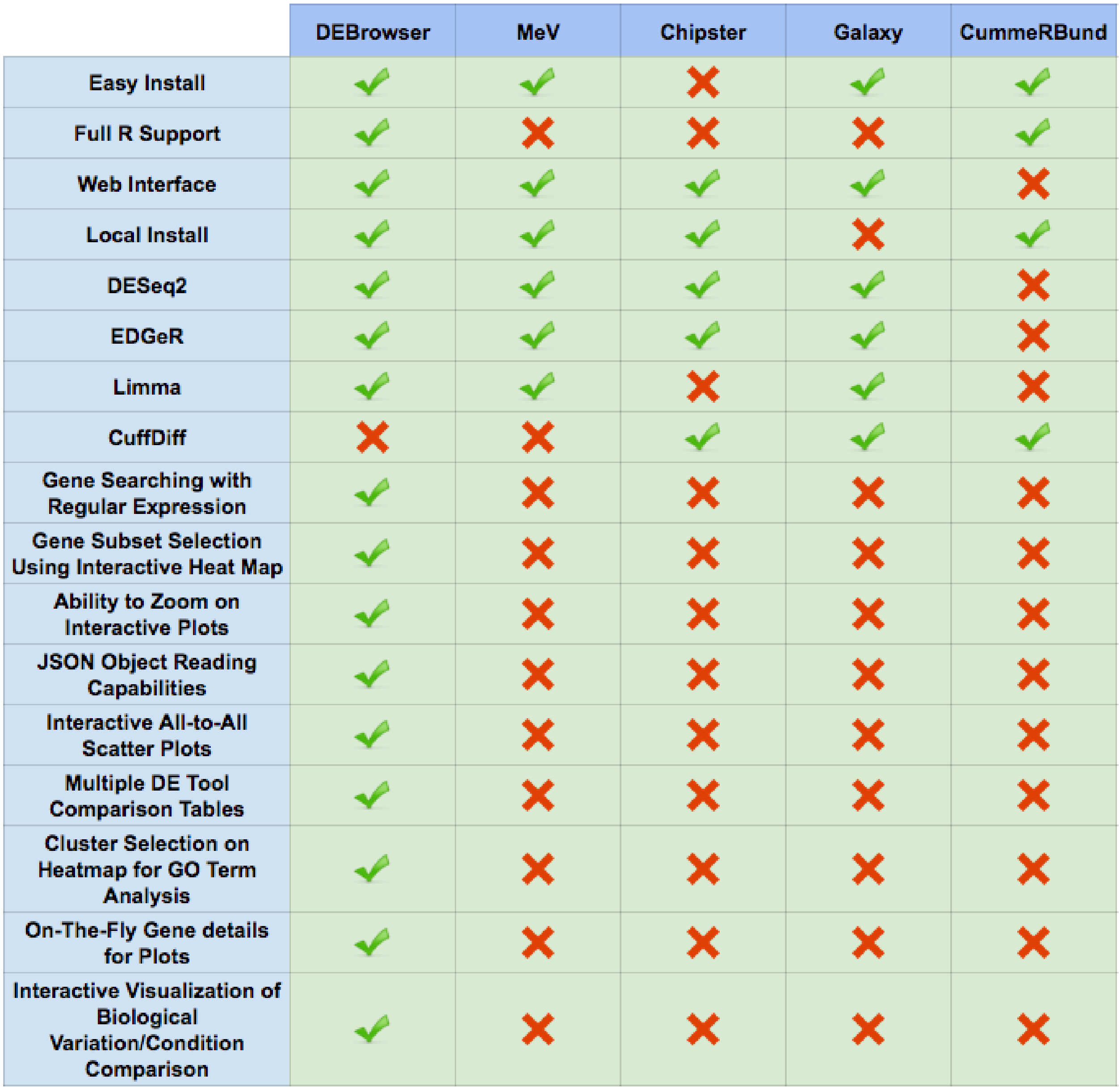
For more details on ComBat, please check the user guide. ComBat userguide.

ComBat is part of the SVA R Bioconductor package which specializes in corecting for known batch effects. No additional parameters are selected or altered when running SVA's ComBat.

3.10 DEBrowser

DEBrowser utilizes Shiny, a R based application development tool that creates a wonderful interactive user interface (UI) combinded with all of the computing prowess of R. After the user has selected the data to analyze and has used the shiny UI to run DESeq2, the results are then input to DEBrowser. DEBrowser manipulates your results in a way that allows for interactive plotting by which changing padj or fold change limits also changes the displayed graph(s). For more details about these plots and tables, please visit our quickstart guide for some helpful tutorials.

For comparisons against other popular data visualization tools, see the table below.



	DEBrowser	MeV	Chipster	Galaxy	CummeRBund
Easy Install	✓	✓	✗	✓	✓
Full R Support	✓	✗	✗	✗	✓
Web Interface	✓	✓	✓	✓	✗
Local Install	✓	✓	✓	✗	✓
DESeq2	✓	✓	✓	✓	✗
EDGeR	✓	✓	✓	✓	✗
Limma	✓	✓	✗	✓	✗
CuffDiff	✗	✗	✓	✓	✓
Gene Searching with Regular Expression	✓	✗	✗	✗	✗
Gene Subset Selection Using Interactive Heat Map	✓	✗	✗	✗	✗
Ability to Zoom on Interactive Plots	✓	✗	✗	✗	✗
JSON Object Reading Capabilities	✓	✗	✗	✗	✗
Interactive All-to-All Scatter Plots	✓	✗	✗	✗	✗
Multiple DE Tool Comparison Tables	✓	✗	✗	✗	✗
Cluster Selection on Heatmap for GO Term Analysis	✓	✗	✗	✗	✗
On-The-Fly Gene details for Plots	✓	✗	✗	✗	✗
Interactive Visualization of Biological Variation/Condition Comparison	✓	✗	✗	✗	✗

For more information on the programs compared against DEBrowser, please visit these pages:

- MeV
- Chipster
- Galaxy
- CummeRbund

3.11 References

1. Anders,S. et al. (2014) HTSeq - A Python framework to work with high-throughput sequencing data.
2. Chang,W. et al. (2016) shiny: Web Application Framework for R.
3. Chang,W. and Wickham,H. (2015) ggvis: Interactive Grammar of Graphics.

4. Giardine,B. et al. (2005) Galaxy: a platform for interactive large-scale genome analysis. Genome Res., 15, 1451–1455.
5. Howe,E.A. et al. (2011) RNA-Seq analysis in MeV. Bioinformatics, 27, 3209–3210.
6. Kallio,M.A. et al. (2011) Chipster: user-friendly analysis software for microarray and other high-throughput data. BMC Genomics, 12, 507.
7. Li,B. and Dewey,C.N. (2011) RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. BMC Bioinformatics, 12, 323.
8. Love,M.I. et al. (2014) Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biol., 15, 550.
9. Reese,S.E. et al. (2013) A new statistic for identifying batch effects in high-throughput genomic data that uses guided principal component analysis. Bioinformatics, 29, 2877–2883.
10. Reich,M. et al. (2006) GenePattern 2.0. Nat. Genet., 38, 500–501.
11. Risso,D. et al. (2014) Normalization of RNA-seq data using factor analysis of control genes or samples. Nat. Biotechnol., 32, 896–902.
12. Ritchie,M.E. et al. (2015) limma powers differential expression analyses for RNA-sequencing and microarray studies. Nucleic Acids Res., 43, e47–e47.
13. Trapnell,C. et al. (2012) Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. Nat. Protoc., 7, 562–578.
14. Vernia,S. et al. (2014) The PPAR$alpha$-FGF21 hormone axis contributes to metabolic regulation by the hepatic JNK signaling pathway. Cell Metab., 20, 512–525.
15. Murtagh, Fionn and Legendre, Pierre (2014). Ward's hierarchical agglomerative clustering method: which algorithms implement Ward's criterion? Journal of Classification 31 (forthcoming).
16. Johnson et al. (2007) Adjusting batch effects in microarray expression data using empirical Bayes methods. Biostatistics, 8, 118-127.

CHAPTER 4

Heatmaps

The heatmap is a great way to analyze replicate results of genes all in one simple plot. Users have the option to change the clustering method used as well as the distance method used to display their heatmap. In addition, you can also change the size of the heatmap produced and adjust the p-adjust and fold change cut off for this plot as well.

4.1 Used clustering and linkage methods in heatmap

- **complete:** Complete-linkage clustering is one of the linkage method used in hierarchical clustering. In each step of clustering, closest cluster pairs are always merged up to a specified distance threshold. Distance between clusters for complete link clustering is the maximum of the distances between the members of the clusters.
- **ward D2:** Ward method aims to find compact and spherical clusters. The distance between two clusters is calculated by the sum of squared deviations from points to centroids. "ward.D2" method uses criterion (Murtagh and Legendre 2014) to minimize ward clustering method. The only difference ward.D2 and ward is the dissimilarities from ward method squared before cluster updating. This method tends to be sensitive to the outliers.
- **single:** Distance between clusters for single linkage is the minimum of the distances between the members of the clusters.
- **average:** Distance between clusters for average linkage is the average of the distances between the members of the clusters.
- **mcquitty:** mcquitty linkage is when two clusters are joined, the distance of the new cluster to any other cluster is calculated by the average of the distances of the soon to be joined clusters to that other cluster.
- **median:** This is a different averaging method that uses the median instead of the mean. It is used to reduce the effect of outliers.
- **centroid:** The distance between cluster pairs is defined as the Euclidean distance between their centroids or means.

4.2 Used distance methods in heatmap

- **cor:** 1 - cor(x) are used to define the dissimilarity between samples. It is less sensitive to the outliers and scaling.
- **euclidean:** It is the most common use of distance. It is sensitive to the outliers and scaling. It is defined as the square root of the sum of the square differences between gene counts.
- **maximum:** The maximum distance between two samples is the sum of the maximum expression value of the corresponding genes.
- **manhattan:** The Manhattan distance between two samples is the sum of the differences of their corresponding genes.
- **canberra:** Canberra distance is similar to the Manhattan distance and it is a special form of the Minkowski distance. The difference is that the absolute difference between the gene counts of the two genes is divided by the sum of the absolute counts prior to summing.
- **minkowsky:** It is generalized form of euclidean distance.

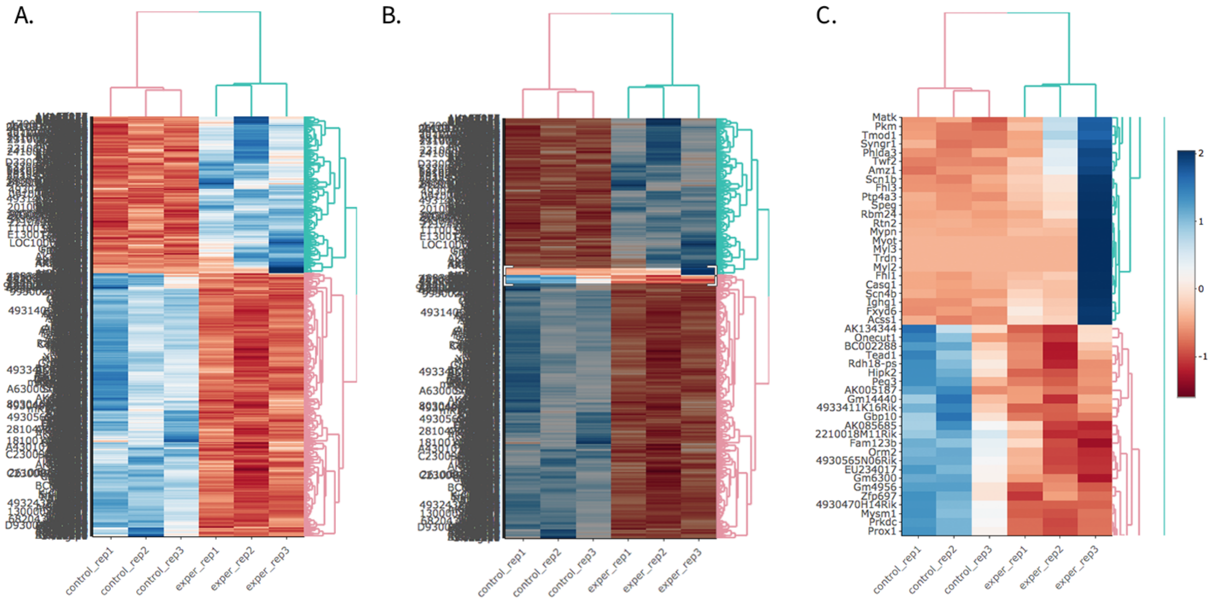
Note: For distances other than 'cor', the distance function defined will be (1 - (the correlation between samples)).

For additional information about the clustering methods you can consult this website and for distance methods here.

4.3 Interactive Heatmap

You can also select to view an interactive version of the heatmap by clicking on the 'Interactive' checkbox on the left panel under the height and width options. Selecting this feature changes the heatmap into an interactive version with two colors, allowing you to select specific genes to be compared within the GO term plots.

Just like in the Main Plots, you can click and drag to create a selection. To select a specific portion of the heatmap, make sure to highlight the middle of the heatmap gene box in order to fully select a specific gene. This selection can be used later within the GO Term plots for specific queries on your selection.

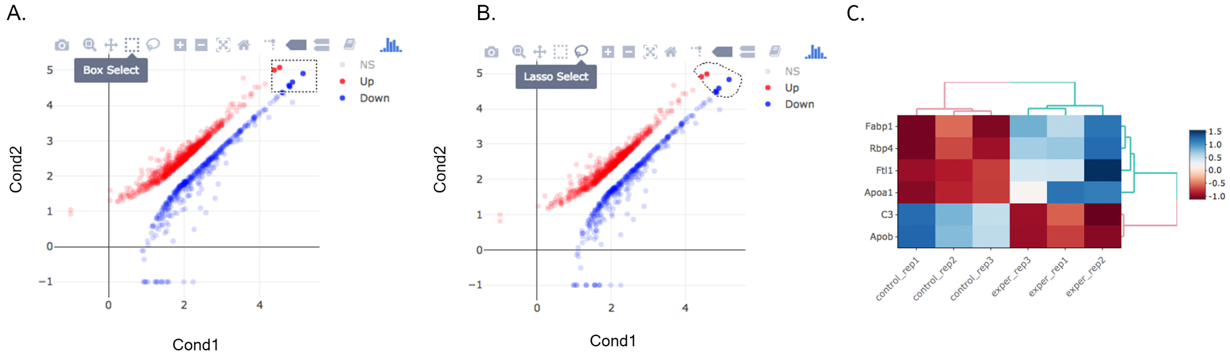


A. Before Selection B. Selection of area with zoom tool C. Zoomed heatmap region which allows better viewing resolution.

Tip: Interactive Feature: In order to increase the performance of the generating heatmaps, **interactive** option is disabled by default. After deciding plotting/clustering parameters of the heatmap, you might activate this feature to investigate each block in detail.

4.4 The Heatmap of DE Analysis

Once you've selected a specific region on Main Plots (Scatter, Volcano or MA plot), a new heatmap of the selected area will appear just next to your plot. If you want to hide some groups (such as Up, Down or NS based on DE analysis), just click on the group label on the top right part of the figure. In this way, you can select a specific part of the genes by **lasso select** or **box select** tools that includes only **Up** or **Down** Regulated genes. As soon as you completed your selection, heatmap will be created simultaneously.



A. Box Selection, **B.** Lasso Selection, **C.** Created heatmap based on selection

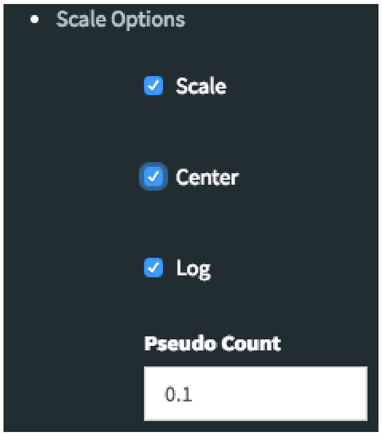
---

Tip: We strongly recommend normalization before plotting heatmaps. To normalize, please change the parameters that are located under: **Data options -> Normalization Methods** and select the method from the dropdown box.

4.5 The Scale Option of Heatmap

By using **Scale Option** field on the left sidebar menu, it is possible to adjust scaling parameters of DEBrowser. There are four main options:

1. **Center**: If it is checked then centering is done by subtracting the column means of data from their corresponding columns. Otherwise no centering is done.(Default value:Checked)
2. **Scale**: The value of scale determines how column scaling is performed (after centering). If scale is checked then scaling is done by dividing the (centered) columns of the data by their standard deviations if center is checked, and the root mean square if center is unchecked. If scale is unchecked, no scaling is done.(Default value:Checked)
3. **Log**: The value of log determines the log2 operation of data matrix (Default value:Checked)
4. **Pseudo-Count**: This value added to each element to prevent getting undefined (logarithm of zero) before calculation of log2(Default value:0.1)



- Scale Options

☑ Scale

☑ Center

☑ Log

Pseudo Count

0.1

CHAPTER 5

DEBrowser Modules

Debrowser is created with moduler structure which allows user to run each module separately. In this guide, you can find explanation about structure of modules and how to run each module.

5.1 Demo Data

To start with, you need to create following variables: `demodata` and `metadatatable` by using your data and save as `demodata.Rda`. The structure of these variables showed at below:

```
> head (demodata)
               exper_rep1 exper_rep2 exper_rep3 control_rep1 control_rep2 control_rep3
AK212155         0.00       0.00          0          0.0            0            0
Sp2             52.00      47.00         36         99.0           53           66
AK051368         4.39       1.11          0          1.1            0            0
Ubiad1         121.00     125.00         65        134.0           95          111
Src             21.00      35.00         20         43.0           22           32
Racgap1          9.00      20.00         11         14.0           10            7

> head (metadatatable)
  samples          treatment batch
1   exper_rep1     cond1      1
2   exper_rep2     cond1      2
3   exper_rep3     cond1      1
4 control_rep1     cond2      2
5 control_rep2     cond2      1
6 control_rep3     cond2      2
```

One way to import tsv files is showed at below:

```
demodata  <- read.table("~/Downloads/shKRAS.tsv", header=T, row.names=1, sep="\t")
```

Now we can run each modules separately through R studio by clicking **Run App** button.

5.2 Example Modules

You can reach demo data and latest versions of modules through our github page.

Barmain plot:

```
library(debrowser)
options(warn =-1)
header <- dashboardHeader( title = "DEBrowser Bar Plots" )

sidebar <- dashboardSidebar(
    sidebarMenu(
        id="DEAnlysis",
        menuItem("BarMain", tabName = "BarMain"),
        textInput("genename", "Gene/Region Name", value = "Foxa3"),
        plotSizeMarginsUI("barmain", h=400)
    )
)

body <- dashboardBody(
    tabItems(
        tabItem(
            tabName="BarMain",
            fluidRow(column(12,getBarMainPlotUI("barmain")))
        )
    )
)

ui <- dashboardPage(header, sidebar, body, skin = "blue")

server <- function(input, output, session) {
    load(system.file("extdata", "demo", "demodata.Rda", package = "debrowser"))
        observe({
            if (!is.null(input$genename))
                callModule(debrowserbarmainplot, "barmain", demodata, metadatatable
↪$sample, metadatatable$treatment, input$genename)
            })
}

shinyApp(ui, server)
```

The example module is created with UI and Server with `shinyApp(ui, server)` command. Similarly UI structure is defined with the `ui <- dashboardPage(header, sidebar, body, skin = "blue")` command. You can simply follow the structure of UI which is created by four variables: `header`, `sidebar`, `body` and `skin`. In the server function, `demodata`, `metadatatable` variables are loaded from `demodata.Rda`, and debrowserbarmainplot module is called with `callModule` function.

Main Plots:

```
library(plotly)
library(debrowser)
header <- dashboardHeader(title = "DEBrowser Main Plots")
sidebar <- dashboardSidebar(  sidebarMenu(id="DEAnalysis",
    menuItem("Main", tabName = "Main"),
    mainPlotControlsUI("main"),
    plotSizeMarginsUI("main")))

```

(continues on next page)

(continued from previous page)

```
body <- dashboardBody(
    tabItems(
        tabItem(tabName="Main", getMainPlotUI("main"),
            column(4,
                   verbatimTextOutput("main_hover"),
                   verbatimTextOutput("main_selected")
            )
        )
    )
)

ui <- dashboardPage(header, sidebar, body, skin = "blue")

server <- function(input, output, session) {
    #Example usage with demodata
    load(system.file("extdata", "demo", "demodata.Rda",
                 package = "debrowser"))
    dat <-c()
    dat$columns <- c("exper_rep1", "exper_rep2", "exper_rep3",
                    "control_rep1", "control_rep2", "control_rep3")
    dat$conds <- factor( c("Control", "Control", "Control",
                    "Treat", "Treat", "Treat") )
    dat$data <- data.frame(demodata[, dat$columns])

    # You can also use your dataset by reading your data from a file like below;
    # The data in this commented out example is not supplied but these lines
    # can give you an idea about how to read the data from a file;
    #
    # data  <- read.table("~/Downloads/shKRAS.tsv", header=T, row.names=1, sep="\t")
    # dat$columns <- c("CNT.2", "CNT.3", "CNT.4",
    #                 "shKRAS_T1", "shKRAS_T2", "shKRAS_T3")
    # dat$conds <- factor( c("Control", "Control", "Control",
    #                         "shKRAS", "shKRAS", "shKRAS") )
    # dat$data <- data.frame(data[, dat$columns])
    #
    xdata <- generateTestData(dat)
    selected <- callModule(debrowsermainplot, "main", xdata)

    output$main_hover <- renderPrint({
        selected$shgClicked()
    })
    output$main_selected <- renderPrint({
        selected$selGenes()
    })
}
shinyApp(ui, server)
```

PCA plot:

```
library(debrowser)
header <- dashboardHeader(title = "DEBrowser PCA Plots")

sidebar <- dashboardSidebar(
    sidebarMenu(id="DataAssessment",
    menuItem("PCA", tabName = "PCA"),
    menuItem("PCA Options",
    pcaPlotControlsUI("pca")),
```

(continues on next page)

(continued from previous page)

```
    plotSizeMarginsUI("pca", w=600, h=400, t=50, b=50, l=60, r=0)
))

body <- dashboardBody(
            tabItems(
                tabItem(tabName="PCA", getPCAPlotUI("pca"),
                    column(4,
                        verbatimTextOutput("pca_hover"),
                        verbatimTextOutput("pca_selected")
                    )
                )
            )
        )

ui <- shinydashboard::dashboardPage(header, sidebar, body, skin = "blue")

server <- function(input, output, session) {
    load(system.file("extdata", "demo", "demodata.Rda", package = "debrowser"))
    selected <- callModule(debrowserpcaplot, "pca", demodata)
}

shinyApp(ui, server)
```

All2All Plot:

```
library(debrowser)
options(warn =-1)
header <- dashboardHeader( title = "DEBrowser All2All Plots")
sidebar <- dashboardSidebar(  sidebarMenu(id="DEAnlysis",
    menuItem("All2All", tabName = "All2All"),
    plotSizeMarginsUI("all2all", h=800, w=800),
    all2allControlsUI("all2all")
    )
)

body <- dashboardBody(
        tabItems(
            tabItem(tabName="All2All",
                fluidRow(column(12,getAll2AllPlotUI("all2all")))
            )
        )
    )

ui <- dashboardPage(header, sidebar, body, skin = "blue")

server <- function(input, output, session) {
    load(system.file("extdata", "demo", "demodata.Rda",package = "debrowser"))
    observe({
        callModule(debrowserall2all, "all2all", demodata, input$cex)
    })
    }

shinyApp(ui, server)
```

Batch Effect Module:

```
library(debrowser)
options(warn =-1)
header <- dashboardHeader(title = "DEBrowser Batch Effect")
sidebar <- dashboardSidebar(sidebarMenu(id="DataPrep",
    menuItem("BatchEffect", tabName = "BatchEffect")
    )
)
body <- dashboardBody(
    tabItems(
        tabItem(tabName="BatchEffect", batchEffectUI("batcheffect"),
            column(4,
                   verbatimTextOutput("batcheffecttable")
            )
        )
    )
)

ui <- dashboardPage(header, sidebar, body, skin = "blue")

server <- function(input, output, session) {
    load(system.file("extdata", "demo", "demodata.Rda", package = "debrowser"))
    ldata <- reactiveValues(count=NULL, meta=NULL)
    ldata$count <- demodata
    ldata$meta <- metadatatable
    data <- callModule(debrowserbatcheffect, "batcheffect", ldata)
    observe({
        output$batcheffecttable <- renderPrint({
            head( data$BatchEffect()$count )
        })
    })
}

shinyApp(ui, server)
```

Main Box plot:

```
library(debrowser)
library(plotly)
options(warn =-1)

header <- dashboardHeader(title = "DEBrowser Box Plots")
sidebar <- dashboardSidebar(  sidebarMenu(id="DEAnlysis",
    menuItem("BoxMain", tabName = "BoxMain"),
    textInput("genename", "Gene/Region Name", value = "Foxa3" ),
    plotSizeMarginsUI("boxmain", h=400, t = 30)
    )
)

body <- dashboardBody(
    tabItems(
        tabItem(tabName="BoxMain",
            fluidRow(column(12,getBoxMainPlotUI("boxmain")))
        )
    )
)
ui <- dashboardPage(header, sidebar, body, skin = "blue")

```

(continues on next page)

(continued from previous page)

```
server <- function(input, output, session) {
    load(system.file("extdata", "demo", "demodata.Rda", package = "debrowser"))
observe({
    if (!is.null(input$genename))
        callModule(debrowserboxmainplot, "boxmain", demodata, metadatatable$sample,␣
↪metadatatable$treatment, input$genename)
    })
}

shinyApp(ui, server)
```

Density Plot:

```
library(debrowser)
options(warn =-1)
header <- dashboardHeader(title = "DEBrowser Density Plots" )
sidebar <- dashboardSidebar(  sidebarMenu(id="DataAssessment",
    menuItem("Density", tabName = "Density"),
    textInput("maxCutoff", "Max Cutoff", value = "10" ),
    plotSizeMarginsUI("density", h=400),
    plotSizeMarginsUI("afterFiltering", h=400)
    )
)

body <- dashboardBody(
    tabItems(
        tabItem(tabName="Density",
            fluidRow(column(12,getDensityPlotUI("density"))),
            fluidRow(column(12,getDensityPlotUI("afterFiltering")))
        )
    )
)

ui <- dashboardPage(header, sidebar, body, skin = "blue")

server <- function(input, output, session) {
    load(system.file("extdata", "demo", "demodata.Rda", package = "debrowser"))
    filtd <- reactive({
    # Filter out the rows that has maximum 100 reads in a sample
    subset(demodata, apply(demodata, 1, max, na.rm = TRUE)  >=  as.numeric(input
↪$maxCutoff))
    })
    observe({
        if(!is.null(filtd())){
            callModule(debrowserdensityplot, "density", demodata)
            callModule(debrowserdensityplot, "afterFiltering", filtd())
        }
    })
}
shinyApp(ui, server)
```

Heatmap Module:

```
library(debrowser)
library(DESeq2)
library(heatmaply)
library(RColorBrewer)
```

(continues on next page)

(continued from previous page)

```
library(gplots)
options(warn=-1)
header <- dashboardHeader(title = "DEBrowser Heatmap")
sidebar <- dashboardSidebar( getJSLine(), sidebarMenu(id="DataAssessment",
    menuItem("Heatmap", tabName = "Heatmap"),
    plotSizeMarginsUI("heatmap"),
    heatmapControlsUI("heatmap"))
)

body <- dashboardBody(
    tabItems(
        tabItem(tabName="Heatmap",  getHeatmapUI("heatmap"),
            column(4,
                   verbatimTextOutput("heatmap_hover"),
                   verbatimTextOutput("heatmap_selected")
            )
        )
    )
)

ui <- dashboardPage(header, sidebar, body, skin = "blue")

server <- function(input, output, session) {
    load(system.file("extdata", "demo", "demodata.Rda", package = "debrowser"))
    insulinSignalingGenes <- reactive({
        genes <-  c("Prkar2a", "Tsc1", "Mapk8", "Sos1", "Pik3r1", "Srebf1",
                    "Insr", "Fasn", "Ppp1r3b", "Pik3r3", "Ptprf", "Pklr",
                    "Irs2", "Socs4", "Eif4ebp1", "Ppp1r3c", "Pygl", "Socs2",
                    "Cbl","Acaca", "Crkl")
        normDat <- getNormalizedMatrix(demodata, method = "MRN")
        normDat[genes, ]
    })
    selected <- reactiveVal()
    observe({
        withProgress(message = 'Creating plot', style = "notification", value = 0.1,
        { selected(callModule(debrowserheatmap, "heatmap", insulinSignalingGenes())) }
        )
    })
    output$heatmap_hover <- renderPrint({
        if (!is.null(selected()) && !is.null(selected()$shgClicked()) &&
            selected()$shgClicked() != "")
            return(paste0("Clicked: ",selected()$shgClicked()))
        else
            return(paste0("Hovered:", selected()$shg()))
    })
    output$heatmap_selected <- renderPrint({
        if (!is.null(selected()))
            selected()$selGenes()
    })
}
shinyApp(ui, server)
```

CHAPTER 6

DEBrowser Additions

In the future, we are going to add:

- Venn Diagrams to compare overlapping differentially expressed genes in different condition comparison results.
- Increase in the number of used clustering methods.
- GO term analysis gene lists will be added for found GO categories.

This page will be updated as new capabilities are added to DEBrowser and/or if we begin new advancements.

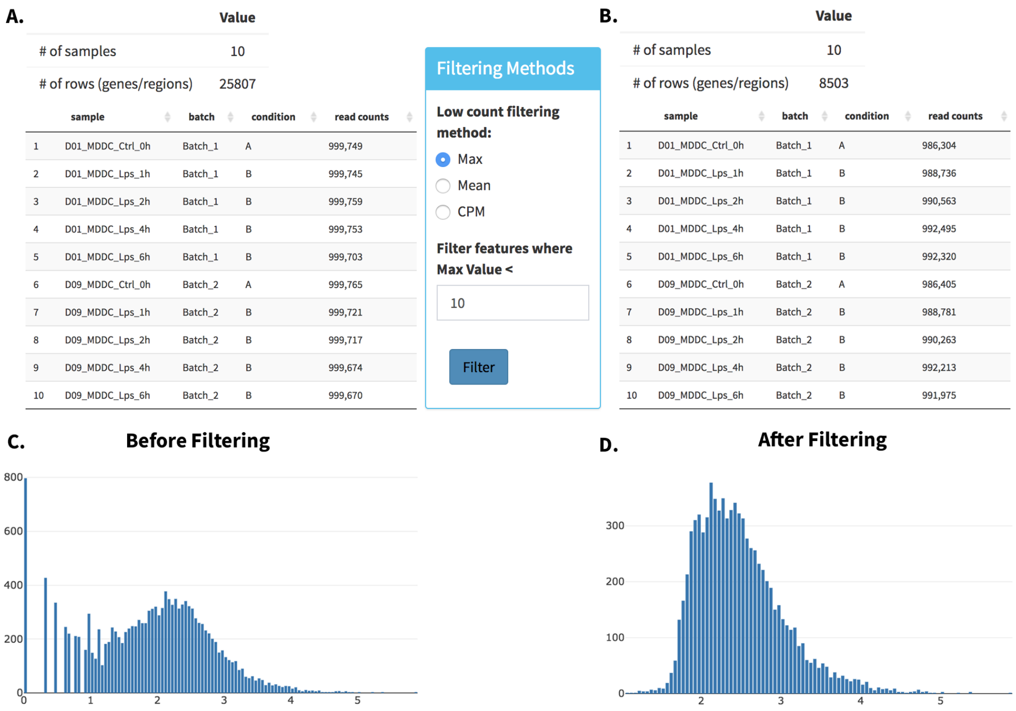
CHAPTER 7

Examples

This guide is walkthrough for the preparation of figures which is used in DEBrowser paper. PCA, Heatmap, All2All will be plotted as an example for QC plots. Next, differential expression analysis will be conducted and their results will be visualized with Main plots such as **Scatter**, **Volcano** and **MA**. More detailed analysis will be covered by using simultaneously created Heatmap and KEGG pathway on the selected portion of the data.

7.1 QC plots without Batch Effect Correction

1) **Upload Data:** To begin the analysis, you need to load Demo Data by clicking **Load Demo (Donnard et al)!** button. Then click on **Filter** button to start **Low Count Filtering**.
2) **Low Count Filtering:** Filtering method is selected as **Max** with cutoff 10 (which filter genes where maximum count for each gene across all samples are less than 10) and activated by clicking **Filter** button which is located at the center of the page. After filtration you can see the distribution of the data as shown at below. Now, you can proceed by clicking **Batch Effect Correction** button.


A.

	Value
# of samples	10
# of rows (genes/regions)	25807

	sample	batch	condition	read counts
1	D01_MDDC_Ctrl_0h	Batch_1	A	999,749
2	D01_MDDC_Lps_1h	Batch_1	B	999,745
3	D01_MDDC_Lps_2h	Batch_1	B	999,759
4	D01_MDDC_Lps_4h	Batch_1	B	999,753
5	D01_MDDC_Lps_6h	Batch_1	B	999,703
6	D09_MDDC_Ctrl_0h	Batch_2	A	999,765
7	D09_MDDC_Lps_1h	Batch_2	B	999,721
8	D09_MDDC_Lps_2h	Batch_2	B	999,717
9	D09_MDDC_Lps_4h	Batch_2	B	999,674
10	D09_MDDC_Lps_6h	Batch_2	B	999,670

Filtering Methods

Low count filtering method:

● Max

○ Mean

○ CPM

Filter features where Max Value <

10

B.

	Value
# of samples	10
# of rows (genes/regions)	8503

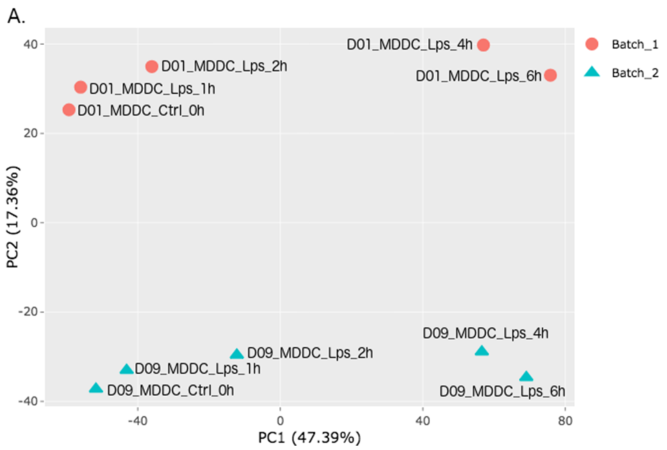
	sample	batch	condition	read counts
1	D01_MDDC_Ctrl_0h	Batch_1	A	986,304
2	D01_MDDC_Lps_1h	Batch_1	B	988,736
3	D01_MDDC_Lps_2h	Batch_1	B	990,563
4	D01_MDDC_Lps_4h	Batch_1	B	992,495
5	D01_MDDC_Lps_6h	Batch_1	B	992,320
6	D09_MDDC_Ctrl_0h	Batch_2	A	986,405
7	D09_MDDC_Lps_1h	Batch_2	B	988,781
8	D09_MDDC_Lps_2h	Batch_2	B	990,263
9	D09_MDDC_Lps_4h	Batch_2	B	992,213
10	D09_MDDC_Lps_6h	Batch_2	B	991,975

3) **Batch Effect Correction and Normalization:** Following options were selected to normalize the data:

    - **Normalization method:** MRN
    - **Correction Method:** None

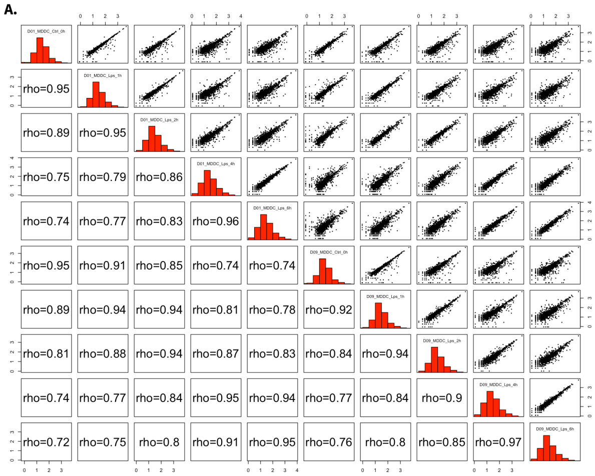
In order to adjust the appearance, use PCA controls which is located between two PCA plots.

- **Text On/Off:** On
- **Select legend:** color
- **Color field:** batch
- **Shape field:** batch

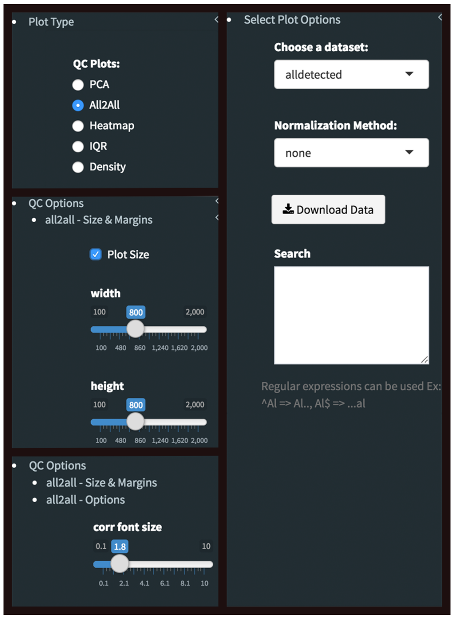


4) **All2All:** After batch effect correction, you can click 'Go to QC plots!' to view quality control

metrics on your data. The page opens with a Principal Component Analysis (PCA) plot. You can select **All2All** option from **Plot type** on the left sidebar menu. In order to get the figure as shown at below, you need to adjust other parameters of **plot options** on the left sidebar menu.



- **All2All - Plot Options:** Following options are selected and their screenshots are shown at below.
  - **Plot Type:** All2All
  - **Data Options:** Choose a dataset: all-detected
  - **QC options - all2all - Size & Margins:** Check the box of the **Plot Size** and adjust width and height as 800 and 800, respectively.
  - **QC options - all2all - Options:** corr font size: 1.8 (adjust the font size of the text inside the box)


Plot Type
QC Options
all2all - Size & Margins
QC Options
all2all - Size & Margins
all2all - Options
Select Plot Options
Download Data
100 480 860 1,240 1,620 2,000
0.1 2.1 4.1 6.1 8.1 10

QC Plots:
○ PCA
● All2All
○ Heatmap
○ IQR
○ Density

☑ Plot Size

width: 800 (100–2,000)

height: 800 (100–2,000)

corr font size: 1.8 (0.1–10)

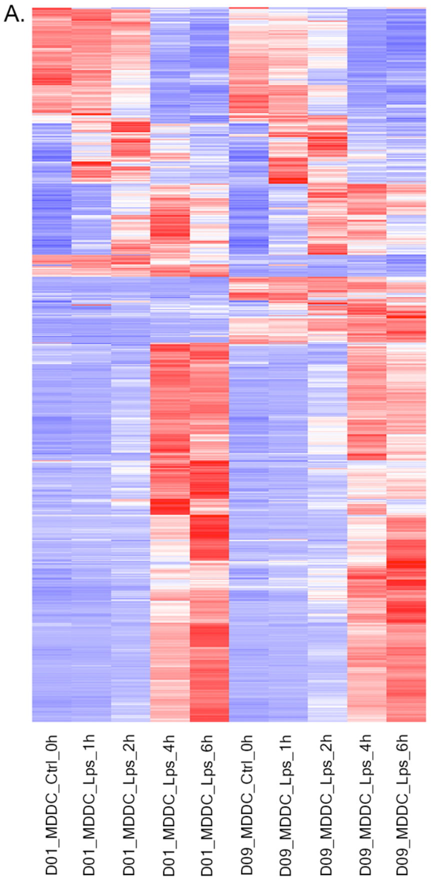
Choose a dataset: alldetected

Normalization Method: none

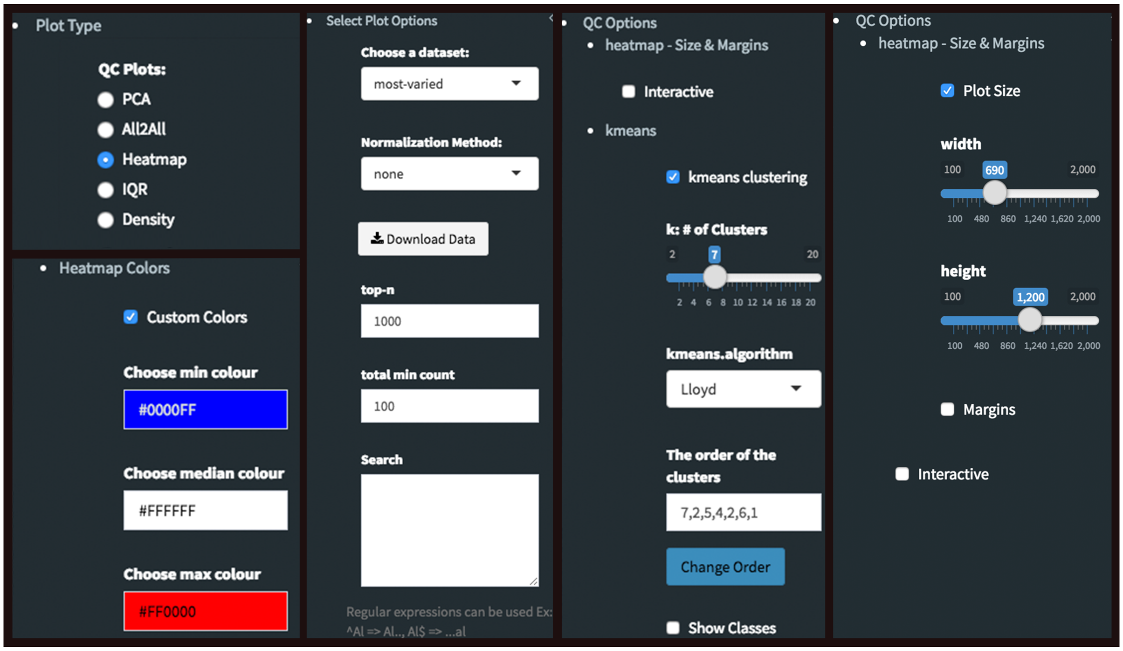
Search

Regular expressions can be used Ex: ^Al => Al.., Al$ => ...al

5) **Heatmap:** To visualize heatmap as shown at below, please select **Heatmap** option from **Plot type** on the left sidebar menu and adjust plot options.



- **Heatmap - Plot Options:** Similar to All2All plot, we need to adjust plotting options on the left sidebar menu.
  - **Plot Type:** Heatmap
  - **Heatmap Colors:** Check the box of custom colors.
  - **Data Options:** Choose a dataset: most varied, top-n:1000, total min count:100 (to show the top 1000 most varied genes (based on coefficient of variance) whose total counts are higher than 100)
  - **QC options - kmeans:** Check the box of kmeans clustering. Select 7 as # of clusters. You might need to change the order of the clusters and click **change order** button to get gradual changes on heatmap as in the figure.
  - **QC options - heatmap - Size & Margins:** Check the box of the **Plot Size** and adjust width and height to 690 and 1200, respectively.



- Plot Type

QC Plots:

○ PCA
○ All2All
● Heatmap
○ IQR
○ Density

- Heatmap Colors

☑ Custom Colors

Choose min colour #0000FF

Choose median colour #FFFFFF

Choose max colour #FF0000

- Select Plot Options

Choose a dataset: most-varied

Normalization Method: none

top-n 1000

total min count 100

Search

Regular expressions can be used Ex: ^Al => Al.., Al$ => ...al

- QC Options
  - heatmap - Size & Margins

☐ Interactive

  - kmeans

☑ kmeans clustering

k: # of Clusters 7

kmeans.algorithm Lloyd

The order of the clusters 7,2,5,4,2,6,1

☐ Show Classes

- QC Options
  - heatmap - Size & Margins

☑ Plot Size

width 690

height 1,200

☐ Margins

☐ Interactive

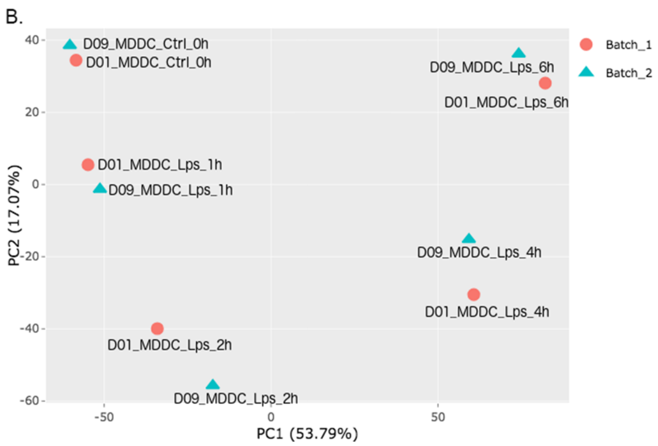
7.2 QC plots after Batch Effect Correction

Since we finalized out plots without applying batch effect correction, we can return back to batch effect correction step and change the **Correction Method** as **Combat** and continue to create new graphs with the same parameters as we used before. To make it more user friendly, we are going to start explain these steps from the beginning. If you choose to continue from batch effect correction, please skip first two steps and continue reading from 3rd step: **Batch Effect Correction and Normalization**.

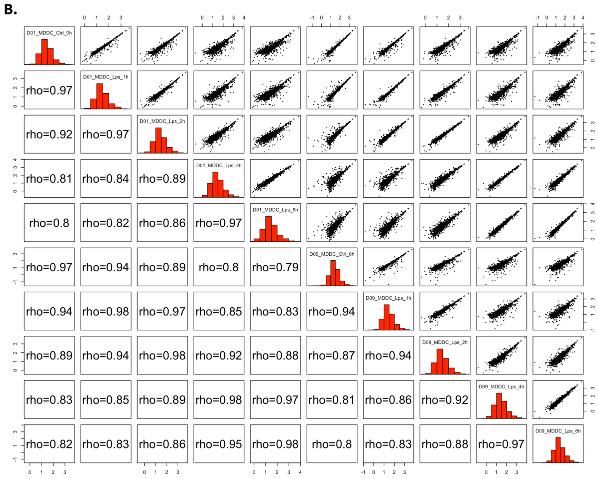
1) **Upload Data:** To begin the analysis, load Demo Data by clicking **Load Demo (Donnard et al)!** button. Then click on **Filter** button to start **Low Count Filtering**.
2) **Low Count Filtering:** Select **Max** method with cutoff 10 (which filter genes where maximum count for each gene across all samples are less than 10), then click **Filter** button which is located at the center of the page. After filtration, proceed to next step by clicking **Batch Effect Correction** button.
3) **Batch Effect Correction and Normalization:** Following options were selected to apply both normalization and batch effect correction:
   - **Normalization method:** MRN
   - **Correction Method:** Combat
   - **Treatment:** treatment
   - **Batch:** batch

Please adjust PCA controls (which is located between two PCA plots) as listed below.

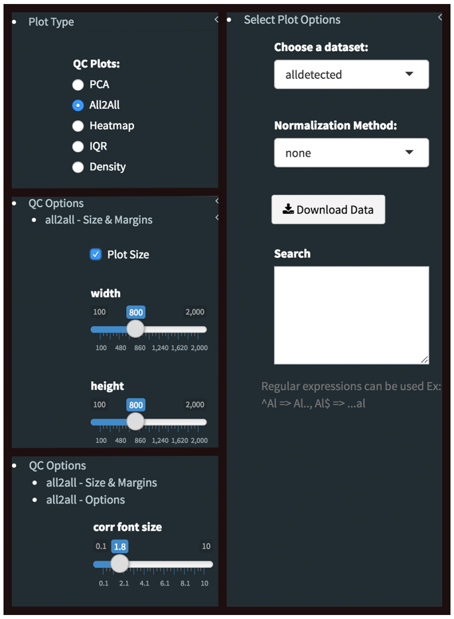
- **Text On/Off:** On
- **Select legend:** color
- **Color field:** batch
- **Shape field:** batch



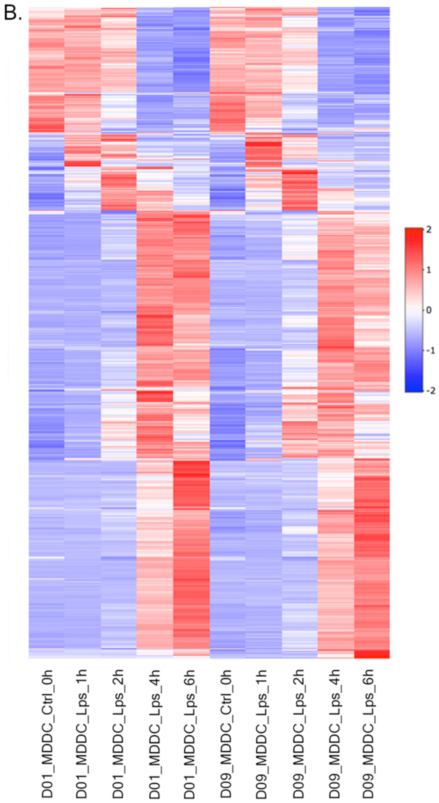
4) **All2All:** After batch effect correction, click 'Go to QC plots!' and select **All2All** option from **Plot type** on the left sidebar menu. Please adjust All2All - Plot Options as listed below.



- **All2All - Plot Options:**
  - **Plot Type:** All2All
  - **Data Options:** Choose a dataset: all-detected
  - **QC options - all2all - Size & Margins:** Check the box of the **Plot Size** and adjust width and height to 800 and 800, respectively.
  - **QC options - all2all - Options:** corr font size: 1.8

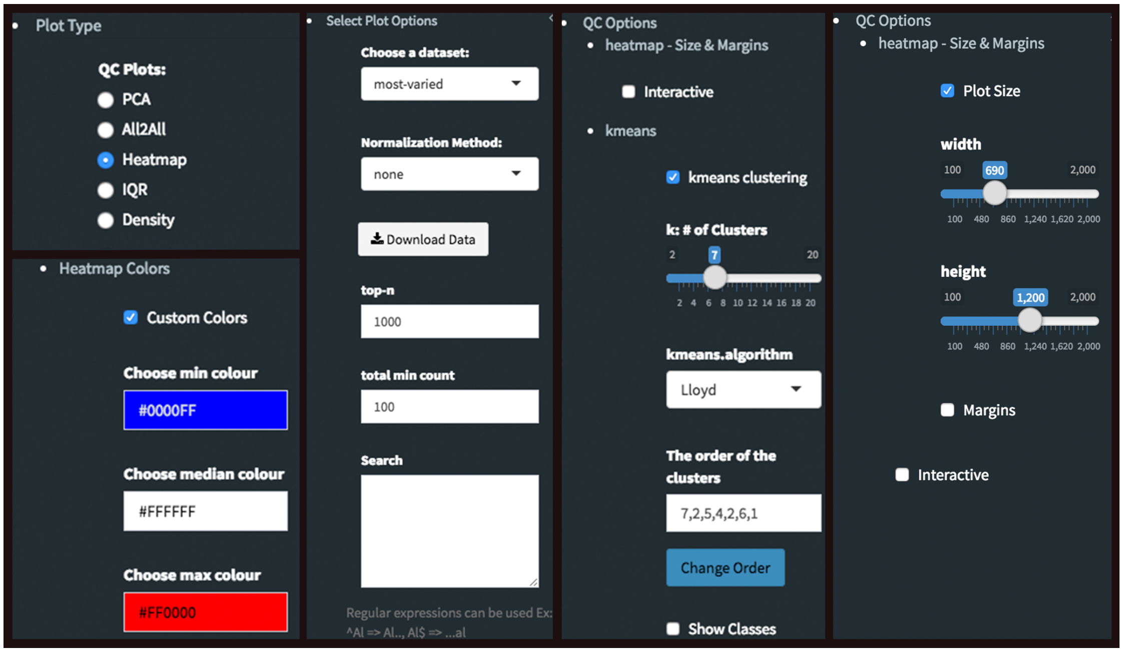

Plot Type
QC Plots:
○ PCA
● All2All
○ Heatmap
○ IQR
○ Density
QC Options
all2all - Size & Margins
☑ Plot Size
width
100 800 2,000
100 480 860 1,240 1,620 2,000
height
100 800 2,000
100 480 860 1,240 1,620 2,000
QC Options
all2all - Size & Margins
all2all - Options
corr font size
0.1 1.8 10
0.1 2.1 4.1 6.1 8.1 10
Select Plot Options
Choose a dataset:
alldetected
Normalization Method:
none
Download Data
Search
Regular expressions can be used Ex:
^Al => Al.., Al$ => ...al

5) **Heatmap:** Please select **Heatmap** option from **Plot type** on the left sidebar menu and adjust plot options according to the list below.



- **Heatmap - Plot Options:**
  - **Plot Type:** Heatmap
  - **Heatmap Colors:** Check the box of custom colors.

   - **Data Options:** Choose a dataset: most varied, top-n:1000, total min count:100 (to show the top 1000 most varied genes (based on coefficient of variance) whose total counts are higher than 100)
   - **QC options - kmeans:** Check the box of kmeans clustering. Select 7 as # of clusters. You might need to change the order of the clusters and click **change order** button to get gradual changes on heatmap as in the figure.
   - **QC options - heatmap - Size & Margins:** Check the box of the **Plot Size** and adjust width and height to 690 and 1200, respectively.



- Plot Type

QC Plots:
○ PCA
○ All2All
● Heatmap
○ IQR
○ Density

- Heatmap Colors

☑ Custom Colors

Choose min colour: #0000FF

Choose median colour: #FFFFFF

Choose max colour: #FF0000

- Select Plot Options

Choose a dataset: most-varied

Normalization Method: none

Download Data

top-n: 1000

total min count: 100

Search

Regular expressions can be used Ex: ^Al => Al.., Al$ => ...al

- QC Options
  - heatmap - Size & Margins

☐ Interactive

  - kmeans

☑ kmeans clustering

k: # of Clusters: 7

kmeans.algorithm: Lloyd

The order of the clusters: 7,2,5,4,2,6,1

Change Order

☐ Show Classes

- QC Options
  - heatmap - Size & Margins

☑ Plot Size

width: 690

height: 1,200

☐ Margins

☐ Interactive

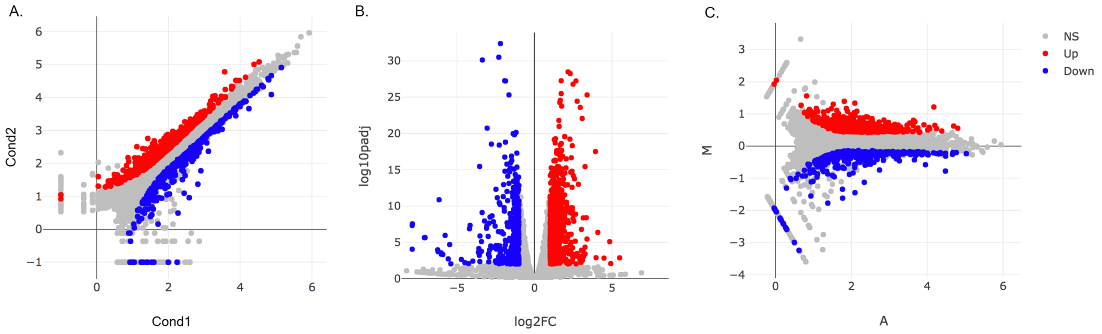
7.3 The Differential Expression Plots

1) **Upload Data:** To begin the analysis, load Count Data by clicking **Load Demo (Vernia et. al)!** button. Then click on **Filter** button to start **Low Count Filtering**.
2) **Low Count Filtering:** Select **Max** method with cutoff 10 (which filter genes where maximum count for each gene across all samples are less than 10), then click **Filter** button which is located at the center of the page. Proceed to next step by clicking **Batch Effect Correction** button.
3) **Batch Effect Correction and Normalization:** We are going to skip both normalization and batch effect correction by selecting following options:
   - **Normalization method:** None
   - **Correction Method:** None
4) **DE Analysis:** After batch effect correction, click 'Go to DE Analysis'. In this page, we will add groups for comparison. Click on **Add New Comparison** button and select **Select Meta** as **treatment**. It will automatically separate experiment and control data into two groups. You can leave other parameters as default as listed below and click "Submit" button.
   - **DE method:** DESeq2
   - **Fit Type:** parametric

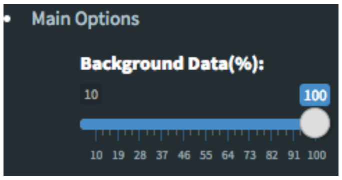
- **betaPrior:** FALSE
- **Test Type:** Wald

examples/../debrowser_pics2/example_DE_form.png

5) **Main Plots Analysis:** Upon finishing the DESeq analysis, you will see DE Results in table format. Please click on **Go to Main Plots!** button which will open **Scatter Plot**. You can switch to **Volcano Plot** and **MA Plot** by using **Plot Type** section at the left side of the menu. Since these plots are interactive, you can click to **zoom** button on the top of the graph and select the area you would like to zoom in by drawing a rectangle. Please see the plots at below:



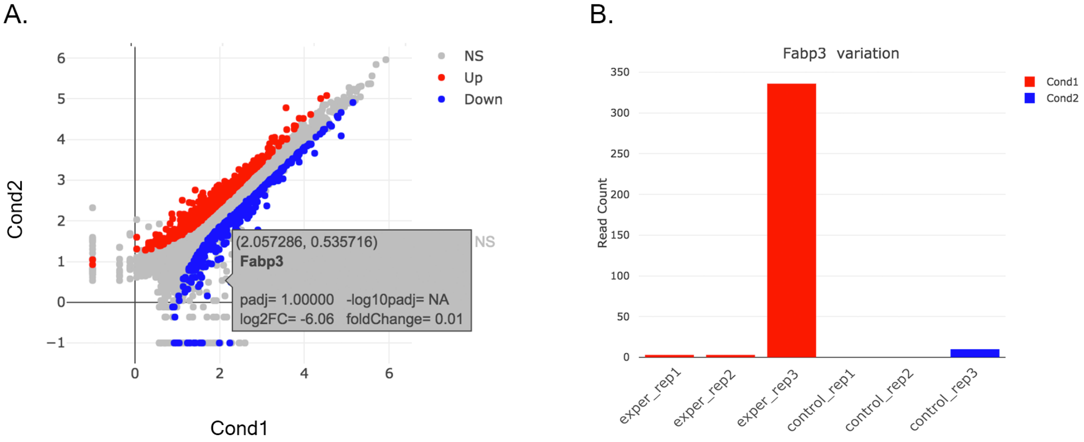
Please keep in mind that to increace the performance of the generating graph, by default 10% of non-significant(NS) genes are used to generate plots. We used all of the NS genes in our plots that showed above, therefore please click **Main Options** button and change Background Data(%) to 100% on the left sidebar.



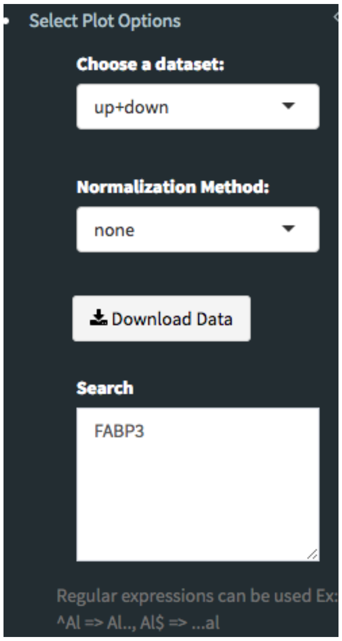
Main Options

Background Data(%): 100

6) **Read count plots:** Lets return back to **Scatter Plot** by using **Plot Type** section. You can hover on each point on the graph to see their read counts as a bar graph as shown at below. In this example FABP3 is selected to show the high variance of this gene across samples.



If you want to mark FABP3 gene on the plot, click on Data Options and enter **FABP3** in to the **search field** as showed below. You will see green mark on the plot that shows FABP3.



Select Plot Options

Choose a dataset: up+down

Normalization Method: none

Search

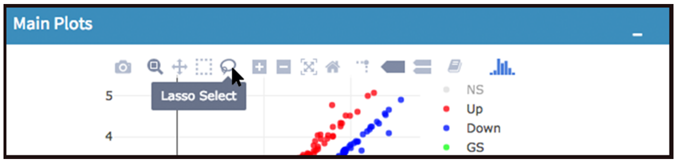
FABP3

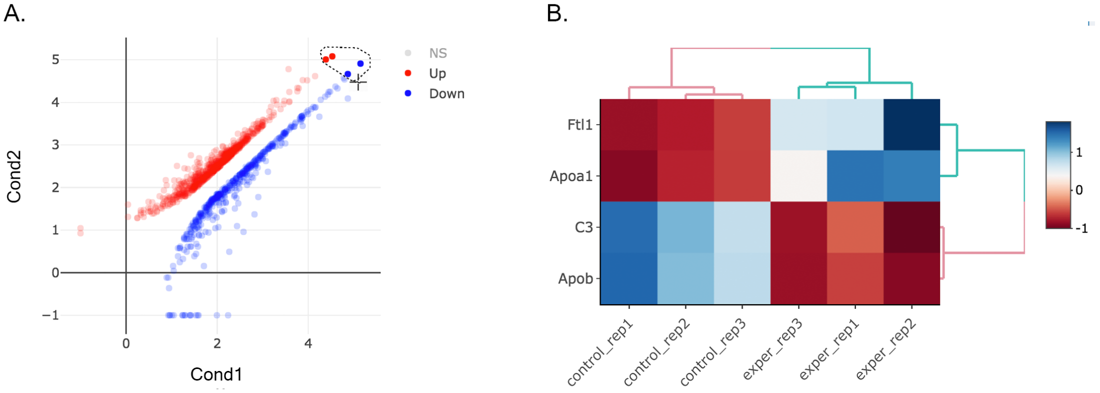
Regular expressions can be used Ex:
^Al => Al.., Al$ => ...al

7) **Lasso selection:** DEBrowser can draw heatmaps of any selected region of any main plot. Selection can be made in a rectangular form or as a free-form using plotly's lasso select. To do so, first click **NS** label at the upper right side of the figure, and hide non-significant genes. Then click on lasso select button at the top of the plot and select the genes you're interested as shown at below. Heatmap

will appear just next to scatter plot. Additionally, you can activate interactive mapping option for heatmap by clicking **Interactive** button under **Heatmap Options** on the left sidebar menu. Now, you can hover on the each block of heatmap to see gene name and its value.

Tip: Interactive Feature: In order to increase the performance of the generating heatmaps, **interactive** option is disabled by default. After deciding plotting/clustering parameters of the heatmap, you might activate this feature to investigate each block in detail.



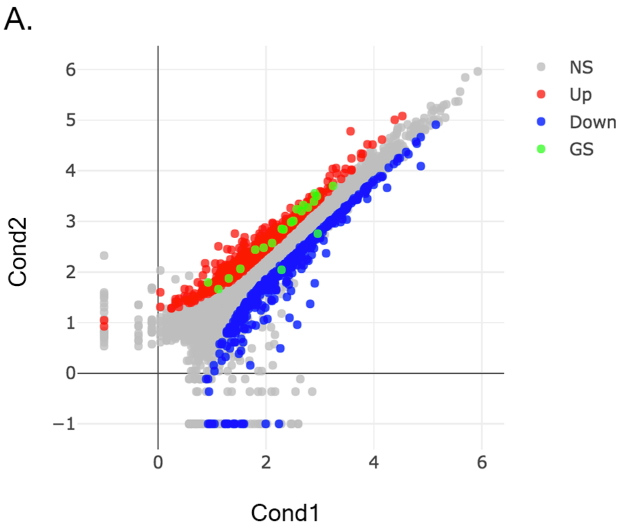


8) **Scatter plot of the genes enriched in insulin signalling pathway:** In this example, we will highlight genes enriched in insulin signalling pathway. If you already hid NS genes, you can show them by clicking on the **NS** label at the upper right side of the figure. Click on the **Data Options** and enter following genes in to the **search field**:

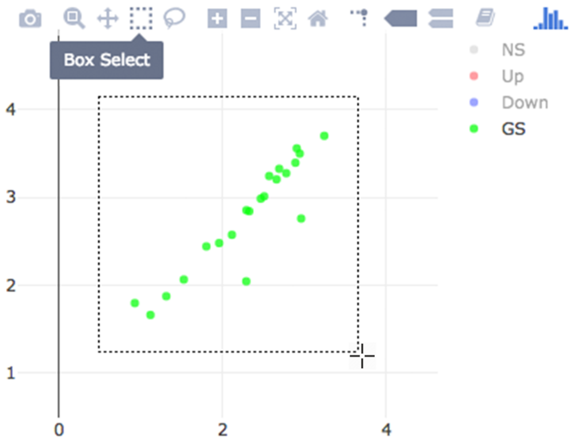
```
Cbl
Sos1
Irs2
Insr
Ptprf
Tsc1
Crkl
Prkar2a
Acaca
Fasn
Mapk8
Ppp1r3b
Ppp1r3c
Srebf1
Pklr
Pik3r1
Pygl
Pik3r3
Socs4
Socs2
Eif4ebp1
```

Tip: If you enter more than three lines of genes, search tool will automatically match the beginning and end of the search phrases. Otherwise it will find matched substrings in the gene list.

Now, you will see green marks on the **searched genes** as shown below:

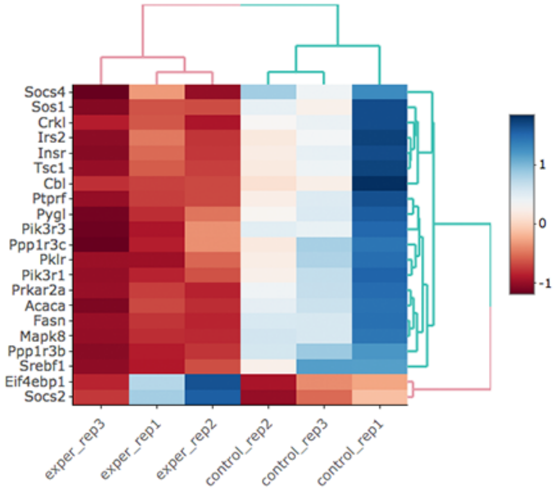


Lets, hide all the genes other then **searched genes** by clicking **NS**, **Up** and **Down** labels at the upper right side of the figure. Since only the selected genes are left on the graph, we can select these genes by clicking on **Select Box** icon and drawing a rectangle which covers all of these genes.



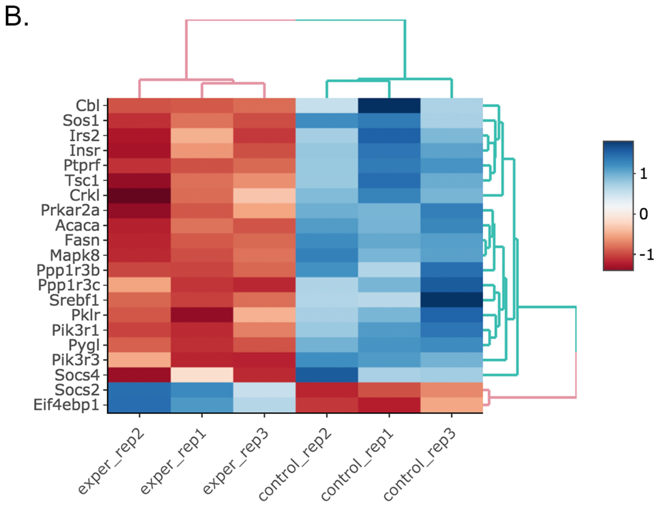
Here as shown below, heatmap will be simultaneously created just next to scatter plot. You might need to change plot margins as following:

- **Heatmap options -> heatmap - Size & Margins:** Please check the box of the **Plot Size** and adjust width and height to 580 and 500, respectively.



Since the data is not normalized, data of exper_rep3 looks like it belongs to control group. We strongly recommend normalization before plotting subset of genes. To normalize, please change the parameters as described below and see the updated figure at below:

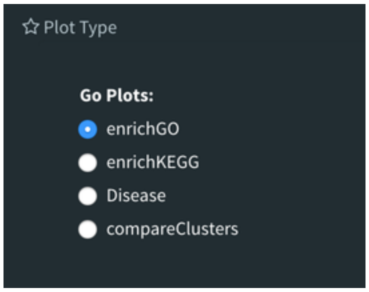
- **Data options -> Normalization Methods:** Please select **MRN** from the dropdown box.



Activating **Interactive** feature changes the heatmap into an interactive version with two colors, allowing you to select specific genes to be compared within the GO term plots.

7.4 GO Term Plots

The next tab, 'GO Term', takes you to the ontology comparison portion of DEBrowser. From here you can select the standard dataset options such as p-adjust value, fold change cut off value, which comparison set to use, and which dataset to use on the left menu. In addition to these parameters, you also can choose from the 4 different ontology plot options: 'enrichGO','enrichKEGG', 'Disease', and 'compareCluster'. Selecting one of these plot options queries their specific databases with your current DESeq results.



Go Plots:

● enrichGO

○ enrichKEGG

○ Disease

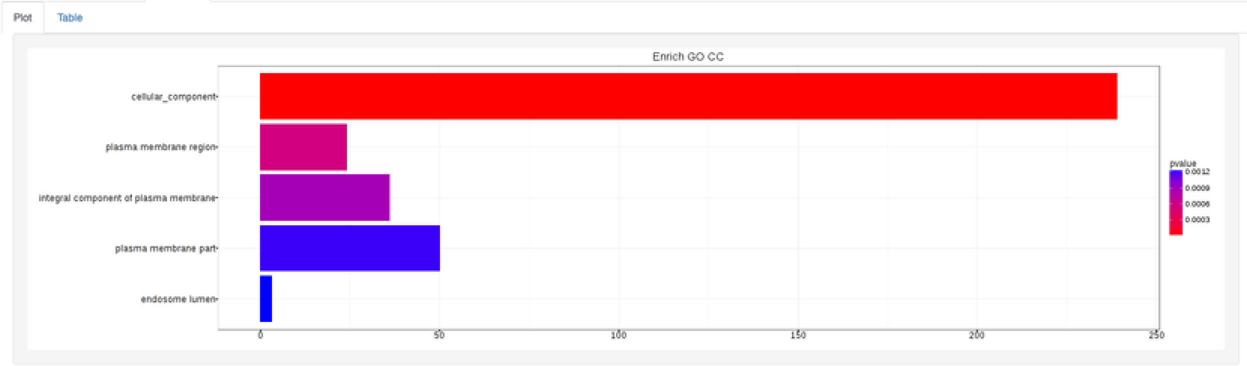
○ compareClusters

Your GO plots include:

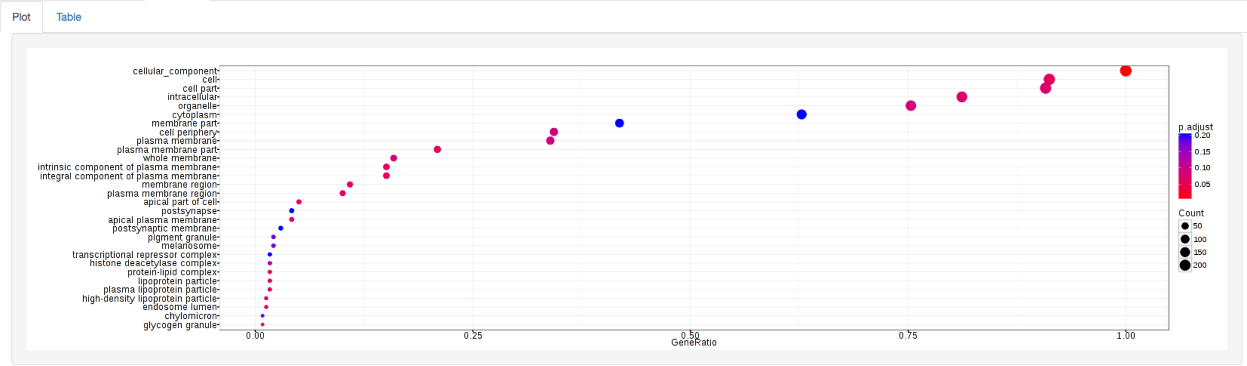
- enrichGO - use enriched GO terms
- enrichKEGG - use enriched KEGG terms
- Disease - enriched for diseases
- compareClusters - comparison of your clustered data

The types of plots you will be able to generate include:

Summary plot:



GOdotplot:



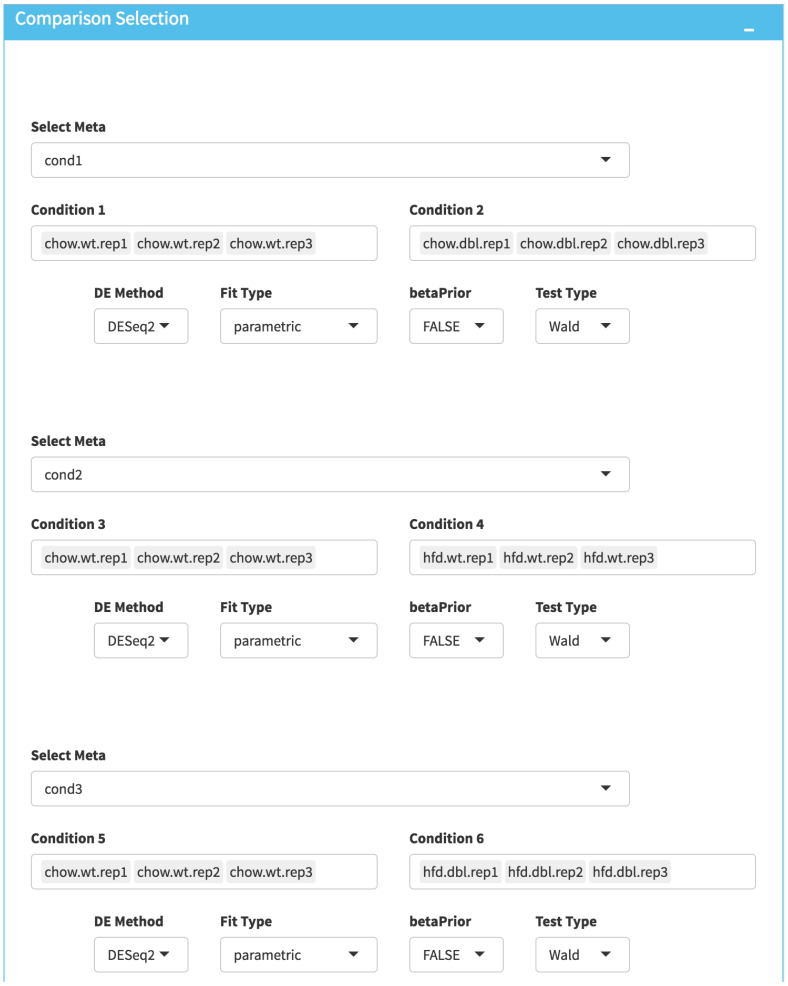
Changing the type of ontology to use will also produce custom parameters for that specific ontology at the bottom of the left option panel.

Once you have adjusted all of your parameters, you may hit the submit button in the top right and then wait for the results to show on screen!

7.5 Log2 fold change comparison for PPARα pathway

1) **Upload Data:** To begin the analysis, download full dataset (Vernia et. al) and full metadata on your computer. Then click **browse** button, and select downloaded files from your computer. Please keep **Separator** as **Tab** while this processes. Finally click **upload** button to see **Upload Summary**. Now you can click on **Filter** button to start **Low Count Filtering**.
2) **Low Count Filtering:** Select **Max** method with cutoff 10 (which filter genes where maximum count for each gene across all samples are less than 10), then click **Filter** button which is located at the center of the page. We are going to skip normalization and batch effect correction step by clicking 'Go to DE Analysis' button.
3) **DE Analysis:** In this page, we will add multiple groups for comparison. Click on **Add New Comparison** button and select **Select Meta** as **Cond1**. Repeat this step for **Cond2** and **Cond3** and add two more comparisons. It will automatically separate experiment and control data into two groups. You can leave other parameters as default as listed below and click "Submit" button.

   - **DE method:** DESeq2
   - **Fit Type:** parametric
   - **betaPrior:** FALSE
   - **Test Type:** Wald



Comparison Selection

Select Meta: cond1

Condition 1: chow.wt.rep1 chow.wt.rep2 chow.wt.rep3

Condition 2: chow.dbl.rep1 chow.dbl.rep2 chow.dbl.rep3

DE Method: DESeq2 | Fit Type: parametric | betaPrior: FALSE | Test Type: Wald

Select Meta: cond2

Condition 3: chow.wt.rep1 chow.wt.rep2 chow.wt.rep3

Condition 4: hfd.wt.rep1 hfd.wt.rep2 hfd.wt.rep3

DE Method: DESeq2 | Fit Type: parametric | betaPrior: FALSE | Test Type: Wald

Select Meta: cond3

Condition 5: chow.wt.rep1 chow.wt.rep2 chow.wt.rep3

Condition 6: hfd.dbl.rep1 hfd.dbl.rep2 hfd.dbl.rep3

DE Method: DESeq2 | Fit Type: parametric | betaPrior: FALSE | Test Type: Wald

4) **Downloading fold2Change data of selected genes**: Upon finishing the DE analysis, you will see DE Results in table format. Please click on **Go to Main Plots!** button which will open **Scatter Plot**. On the left sidebar menu, click **Data options* tab and enter following genes regarding to PPARα pathway:

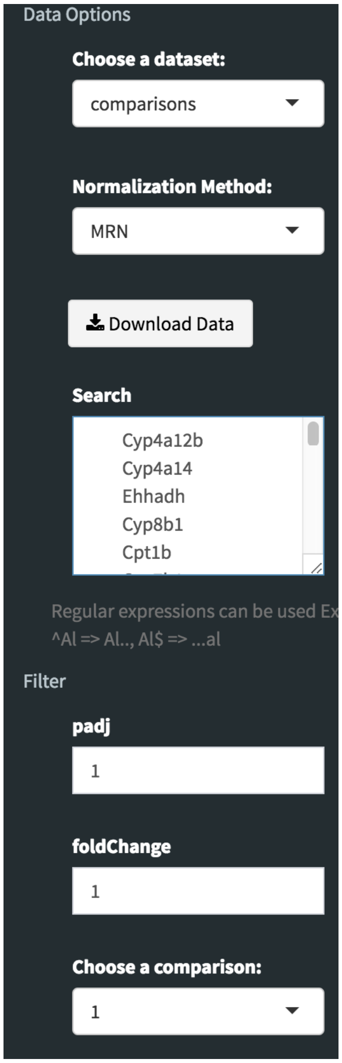
```
Cyp4a12b
Cyp4a14
Ehhadh
Cyp8b1
Cpt1b
Cyp7b1
Slc27a1
Apoa5
Pdpk1
Apoa1
Acadl
Fads2
Fabp4
Acadm
Apoa2
Apoc3
Fgf21
Fabp5
Fabp3
Lpl
Dbi
Nr1h3
Fabp7
Ppara
Ucp1
Sdc1
Sdc3
Sdc2
Fabp2
```

Afterwards, select **comparison** option for the **Choose a dataset** field. This option will add fold change columns to to our data.

Now, we need to disable filtration to get all searched genes in our dataset. To do so, enter following parameters into **Filter** field on the left sidebar menu.

- **padj:** 1
- **foldChange:** 1

To confirm you can check all adjusted parameters at image below.



Data Options

Choose a dataset: comparisons

Normalization Method: MRN

Search

Cyp4a12b
Cyp4a14
Ehhadh
Cyp8b1
Cpt1b

Regular expressions can be used Ex
^Al => Al.., Al$ => ...al

Filter

padj: 1

foldChange: 1

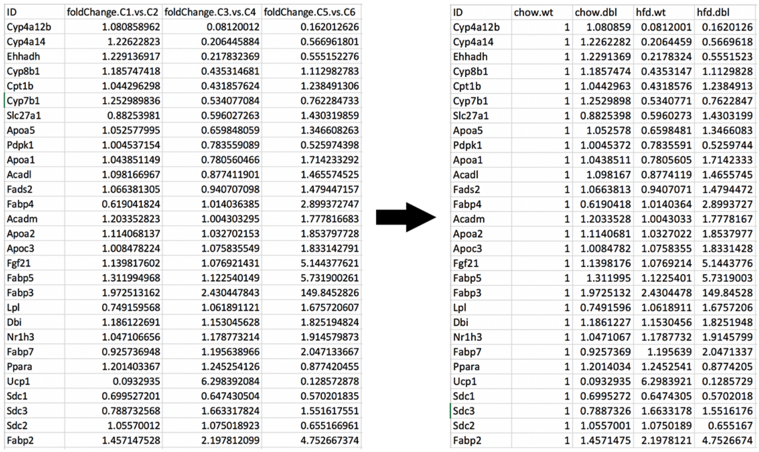
Choose a comparison: 1

It is time to download our dataset by clicking **Download Data** button on the **Data Options** field. You can open downloaded tsv file in Excel or similar programs. Once you open the file, you will see columns of count data, padj and fold2Change for all comparisons. Since we are only interested in fold2Change columns, you can delete the rest. Final data file should look like image on the left at below.

We will rename column names as follows and add new column called **chow.wt** which compares chow.wildtype with itself therefore it is filled with 1.

- **foldChange.C1.vs.C2** to chow.dbl
- **foldChange.C3.vs.C4** to hfd.wt
- **foldChange.C5.vs.C6** to hfd.dbl

To confirm you can also download the final version of the fold2data from this link.



ID	foldChange.C1.vs.C2	foldChange.C3.vs.C4	foldChange.C5.vs.C6
Cyp4a12b	1.080858962	0.08120012	0.162012626
Cyp4a14	1.22622823	0.206445884	0.566961801
Ehhadh	1.229136917	0.217832369	0.555152276
Cyp8b1	1.185747418	0.435314681	1.112982783
Cpt1b	1.044296298	0.431857624	1.238491306
Cyp7b1	1.252989836	0.534077084	0.762284733
Slc27a1	0.88253981	0.596027263	1.430319859
Apoa5	1.052577995	0.659848059	1.346608263
Pdpk1	1.004537154	0.783559089	0.525974398
Apoa1	1.043851149	0.780560466	1.714233292
Acadl	1.098166967	0.877411901	1.465574525
Fads2	1.066381305	0.940707098	1.479447157
Fabp4	0.619041824	1.014036385	2.899372747
Acadm	1.203352823	1.004303295	1.777816683
Apoa2	1.114068137	1.032702153	1.853797728
Apoc3	1.008478224	1.075835549	1.833142791
Fgf21	1.139817602	1.076921431	5.144377621
Fabp5	1.311994968	1.122540149	5.731900261
Fabp3	1.972513162	2.430447843	149.8452826
Lpl	0.749159568	1.061891121	1.675720607
Dbi	1.186122691	1.153045628	1.825194824
Nr1h3	1.047106656	1.178773214	1.914579873
Fabp7	0.925736948	1.195638966	2.047133667
Ppara	1.201403367	1.245254126	0.877420455
Ucp1	0.0932935	6.298392084	0.128572878
Sdc1	0.699527201	0.647430504	0.570201835
Sdc3	0.788732568	1.663317824	1.551617551
Sdc2	1.05570012	1.075018923	0.655166961
Fabp2	1.457147528	2.197812099	4.752667374

ID	chow.wt	chow.dbl	hfd.wt	hfd.dbl
Cyp4a12b	1	1.080859	0.0812001	0.1620126
Cyp4a14	1	1.2262282	0.2064459	0.5669618
Ehhadh	1	1.2291369	0.2178324	0.5551523
Cyp8b1	1	1.1857474	0.4353147	1.1129828
Cpt1b	1	1.0442963	0.4318576	1.2384913
Cyp7b1	1	1.2529898	0.5340771	0.7622847
Slc27a1	1	0.8825398	0.5960273	1.4303199
Apoa5	1	1.052578	0.6598481	1.3466083
Pdpk1	1	1.0045372	0.7835591	0.5259744
Apoa1	1	1.0438511	0.7805605	1.7142333
Acadl	1	1.098167	0.8774119	1.4655745
Fads2	1	1.0663813	0.9407071	1.4794472
Fabp4	1	0.6190418	1.0140364	2.8993727
Acadm	1	1.2033528	1.0043033	1.7778167
Apoa2	1	1.1140681	1.0327022	1.8537977
Apoc3	1	1.0084782	1.0758355	1.8331428
Fgf21	1	1.1398176	1.0769214	5.1443776
Fabp5	1	1.311995	1.1225401	5.7319003
Fabp3	1	1.9725132	2.4304478	149.84528
Lpl	1	0.7491596	1.0618911	1.6757206
Dbi	1	1.1861227	1.1530456	1.8251948
Nr1h3	1	1.0471067	1.1787732	1.9145799
Fabp7	1	0.9257369	1.195639	2.0471337
Ppara	1	1.2014034	1.2452541	0.8774205
Ucp1	1	0.0932935	6.2983921	0.1285729
Sdc1	1	0.6995272	0.6474305	0.5702018
Sdc3	1	0.7887326	1.6633178	1.5516176
Sdc2	1	1.0557001	1.0750189	0.655167
Fabp2	1	1.4571475	2.1978121	4.7526674

5) **Creating Heatmap for fold2change data**: To create heatmap for fold change data, you have two options: A. Using startHeatmap() function or B. Use DEBrowser Heatmap module.

   - A. Open new R session and run following command in R or R Studio to run Heatmap module in web browser:

     ```
     startHeatmap()
     ```

   Similar to DEBrowser, you can click **browse** button, and select prepared log2change file from your computer. Please keep **Separator** as **Tab**. Finally click **upload** button to see **Upload Summary**.

   - B. Open new R session and run following command in R or R Studio to load dataset as data frame (comparisons):

     ```
     comparisons <- read.delim("~/Downloads/comparisons.tsv", row.
     ↪names=1)
     ```

     You may need to change the path of the file according to your folder structure. Now, in order to open heatmap module, you need to run following script:

     ```
     library(debrowser)
     library(DESeq2)
     library(heatmaply)
     library(RColorBrewer)
     library(gplots)

     options(warn=-1)
     header <- dashboardHeader(title = "DEBrowser Heatmap")
     sidebar <- dashboardSidebar(  getJSLine(), sidebarMenu(id=
     ↪"DataAssessment",
         menuItem("Heatmap", tabName = "Heatmap"),
         plotSizeMarginsUI("heatmap"),
         heatmapControlsUI("heatmap")))
     body <- dashboardBody(
         tabItems(
         tabItem(tabName="Heatmap",  getHeatmapUI("heatmap"),
             column(4, verbatimTextOutput("heatmap_hover"),␣
     ↪verbatimTextOutput("heatmap_selected")
             )
         )
     ```

(continues on next page)

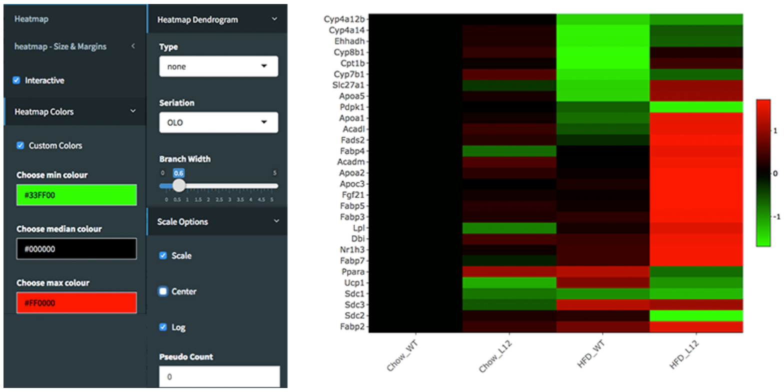
(continued from previous page)

```
))

ui <- dashboardPage(header, sidebar, body, skin = "blue")

server <- function(input, output, session) {
selected <- reactiveVal()
observe({
    withProgress(message = 'Creating plot', style =
↪"notification", value = 0.1, {
    selected(callModule(debrowserheatmap, "heatmap",␣
↪comparisons))
    })
})
output$heatmap_hover <- renderPrint({
    if (!is.null(selected()) && !is.null(selected()
↪$shgClicked()) && selected()$shgClicked() != "")
        return(paste0("Clicked: ",selected()$shgClicked()))
    else
        return(paste0("Hovered:", selected()$shg()))
})
output$heatmap_selected <- renderPrint({
    if (!is.null(selected()))
        selected()$selGenes()
})
}

shinyApp(ui, server)
```

Shiny will launch a web browser which is ready to use as a heatmap module. You need to specify following parameters to create log2fold change graph:

- **Interactive:** Checked
- **Custom Colors:** Checked
- **Custom Colors -> Choose min colour:** #33FF00
- **Custom Colors -> Choose median colour:** #000000
- **Custom Colors -> Choose max colour:** #FF0000
- **Heatmap Dendrogram -> Type:** none
- **Scale Options -> Scale:** Checked
- **Scale Options -> Center:** Unchecked
- **Scale Options -> Log:** Checked
- **Scale Options -> Pseudo Count:** 0

Once you specify these parameters, your heatmap will be seen as image at below.



Heatmap

heatmap - Size & Margins

☑ Interactive

Heatmap Colors

☑ Custom Colors

Choose min colour: #33FF00

Choose median colour: #000000

Choose max colour: #FF0000

Heatmap Dendrogram

Type: none

Seriation: OLO

Branch Width: 0.6

Scale Options

☑ Scale

☐ Center

☑ Log

Pseudo Count: 0

0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
Cyp4a12b
Cyp4a14
Ehhadh
Cyp8b1
Cpt1b
Cyp7b1
Slc27a1
Apoa5
Pdpk1
Apoa1
Acadl
Fads2
Fabp4
Acadm
Apoa2
Apoc3
Fgf21
Fabp5
Fabp3
Lpl
Dbi
Nr1h3
Fabp7
Ppara
Ucp1
Sdc1
Sdc3
Sdc2
Fabp2
Chow_WT
Chow_L12
HFD_WT
HFD_L12
1
0
-1

CHAPTER 8

Frequently asked questions (FAQ)

8.1 Why un-normalized counts?

DESeq2 requires count data as input obtained from RNA-Seq or another high-thorughput sequencing experiment in the form of matrix values. Here we convert un-integer values to integer to be able to run DESeq2. The matrix values should be un-normalized, since DESeq2 model internally corrects for library size. So, transformed or normalized values such as counts scaled by library size should not be used as input. Please use edgeR or limma for normalized counts.

8.2 Why am I getting error while uploading files?

- DEBrowser supports tab, comma or semi-colon separated files. However spaces or characters in numeric regions not supported and causes an error while uploading files. It is crutial to remove these kind of instances from the files before uploading files.
- Another reason of getting an error is using same gene name multiple times. This may occurs after opening files in programs such as Excel, which tends to automatically convert some gene names to dates (eg. SEP9 to SEP.09.2018). This leads numerous problems therefore you need to disable these kind of automatic conversion before opening files in these kind of programs.
- Some files contain both tab and space as an delimiter which lead to error. It is required to be cleaned from these kind of files before loading.

8.3 Why some columns not showed up after upload?

If a character in numeric area or space is exist in one of your column, either column will be eliminated or you will get an error. Therefore it is crutial to remove for these kind of instances from your files before uploading.

8.4 Why am I getting error while uploading CSV/TSV files exported from Excel?

- You might getting an error, because of using same gene name multiple times. This may occurs after opening files in programs such as Excel, which tends to automatically convert some gene names to dates (eg. SEP9 to SEP.09.2018). Therefore you need to disable these kind of automatic conversion before opening files in these kind of programs.

8.5 Why can't I see all the background data in Main Plots?

In order to increase the performance, by default 10% of non-significant(NS) genes are used to generate plots. We strongly suggest you to use all of the NS genes in your plots while publishing your results. You can easily change this parameter by clicking **Main Options** button and change Background Data(%) to 100% on the left sidebar.

8.6 Why am I getting error when I click on DE Genes in Go Term Analysis?

To start **Go Term** analysis, it is important to select correct organism from **Choose an organism** field. After selecting other desired parameters, you can click **Submit** button to run Go Term analysis. After this stage, you will able to see **categories** regarding to your selected gene list in the **Table** Tab. Once you select this category, you can click DE Genes button to see gene list regarding to selected category.

8.7 How to download selected data from Main plots/QC Plots/Heatmaps?

First, you need to choose **Choose dataset** field as **selected** under **Data Options** in the left sidebar. When you select this option, new field: **The plot used in selection** will appear under **Choose dataset** field. You need to specify the plot you are interested from following options: Main plot, Main Heatmap, QC Heatmap. Finally you can click **Download Data** button to download data, or if you wish to see the selected data, you can click **Tables** tab.

CHAPTER 9

Indices and tables

- genindex
- modindex
- search