DESeq2 (multifactorial pairwise comparisons)

For an introduction to using the DE, see Using the Discovery Environment.
Please work through the tutorial and add your comments on the bottom of this page, or email comments to support@cyverse.org. Thank you.

Rationale and background

Currently, DeSeq apps (both DeSeq and DeSeq2) in DE, do not allow multifactorial pairwise comparison of RNA-Seq data for differential gene expression analysis. The app - "DEseq2 (multifactorial pairwise comparisons)" is based on SARTools (R package dedicated to the differential analysis of RNA-seq data) which allows multifactorial pairwise comparison of RNA-Seq data for differential gene expression analysis. It provides tools to generate descriptive and diagnostic graphs, to run the differential analysis with the DEseq2 package, and to export the results into easily readable tab-delimited files. It also facilitates the generation of an HTML report which displays all the figures produced, explains the statistical methods, and gives the results of the differential analysis.

The SARTools R package has been developed at PF2 - Institut Pasteur by M.-A. Dillies and H. Varet (hugo.varet@pasteur.fr). Please cite H. Varet, L. Brillet-Guéguen, J.-Y. Coppee and M.-A. Dillies, SARTools: A DESeq2- and EdgeR-Based R Pipeline for Comprehensive Differential Analysis of RNA-Seq Data, PLoS One, 2016, doi: http://dx.doi.org/10.1371/journal.pone.0157022 when using this tool for any analysis published."

Note
SARTools does not intend to replace edgeR: it simply provides an environment to go with them. For more details about the methodology behind edgeR, the user should read their documentation and papers. In addition, the current app is not intended to perform edgeR's GLM. That version is currently under progress.

Introduction and Overview

DESeq2 estimates differentially expressed gene lists based on a negative binomial distribution model. Previous methods for identifying differentially expressed gene lists assumed a Poisson distribution; however, Poisson does not account for variation (or overdispersion) found in expression data. DESeq2 uses a negative binomial distribution (similar to edgeR), assuming variance in the case of few replicates.
The input is a tab-delimited file containing genes and their expression values. The results include files detailing the results of differential expression testing (one that includes all of the results, and one that only includes the results that exceed a minimum false-discovery rate). Also included for visualization purposes are plots of the estimated dispersions, the log fold changes against the mean normalized counts and a histogram of p-values. The plots are purely for visualization purposes and may not be necessary for all users.

Prerequisites

1. A CyVerse account (Register for a CyVerse account at https://user.cyverse.org/).
2. An up-to-date Java-enabled web browser. (Firefox recommended. If you wish to work with your own large datasets and upload them using iCommands, Chrome is not suitable due to its issues in utilizing 64-bit Java.)
3. Input files:

    a. **Target file**: The user has to supply a tab-delimited file which describes the experiment, i.e. which contains the name of the biological condition associated with each sample. This file is called "target". This file has one row per sample and is composed of at least three columns with headers:
        - first column: unique names of the samples (short but informative as they will be displayed on all the figures); (Ex: "label")
        - second column: name of the count files; (Ex: "file")
        - third column: biological conditions; (Ex: "group)
        - optional columns: further information about the samples (day of library preparation for example). (Ex: "cond")

    The table below shows an example of a target file:

label	file	group	cond
5_OP_1	count3.txt	OP	5
5_OP_2	count3.txt	OP	5
5_OP_3	count3.txt	OP	5
33_OP_1	count3.txt	OP	33
33_OP_2	count3.txt	OP	33
33_OP_3	count3.txt	OP	33
5_M_1	count3.txt	M	5

5_M_2	count3.txt	M	5
5_M_3	count3.txt	M	5
33_M_1	count3.txt	M	33
33_M_2	count3.txt	M	33
33_M_3	count3.txt	M	33
5_LL_1	count3.txt	LL	5
5_LL_2	count3.txt	LL	5
5_LL_3	count3.txt	LL	5
33_LL_1	count3.txt	LL	33
33_LL_2	count3.txt	LL	33
33_LL_3	count3.txt	LL	33

b. **Raw counts file or Raw counts folder**: The DESeq2 statistical analysis assumes that reads have already been mapped and that counts per feature (gene or transcript) are available. There are two different ways to provide the option to the app.
   - A raw counts file that contains all the samples, each column corresponds to a sample with gene/transcript the same and a column first column which consists of the unique IDs of the features. (See an example of this type in Table 2). You can use Htseq-Count-Merge-0.6.1 to generate that kind of file.
   - A directory consisting of one count file per sample with two tabs delimited columns without the header. The first column is the unique IDs of the features and the second column has raw counts associated with these features (null or positive integers) (See an example of this type in Table 3). You can use HTSeq-count-0.6.1 to generate this type of directory

Table 2: Example of a Raw counts file:

Contig	5_OP_1	5_OP_2	5_OP_3	33_OP_1	33_OP_2	33_OP_3	5_M_1	5_M_2	5_M_3	33_M_1	33_M_2	33_M_3	5_LL_1	5_
oystercontig_1	8	54	10	17	3	1	19	47	42	44	6	2	229	47
oystercontig_2	16	4	16	56	2	1	2	3	0	28	0	0	2	19
oystercontig_3	2	8	3	13	2	2	1	24	20	41	4	8	23	12
oystercontig_4	7	2	24	139	2	2	3	1	2	10	0	0	1	1
oystercontig_5	0	2	1	1	0	0	0	0	1	0	0	0	1	0
oystercontig_6	0	0	0	3	0	0	7	0	0	2	0	0	1	0
oystercontig_7	127	30	9	46	13	7	153	111	60	60	2	13	245	20
oystercontig_8	154	386	57	561	91	123	566	693	503	851	47	129	634	92
oystercontig_9	1	1	0	0	0	0	20	3	4	1	0	0	33	11

Table 3: Example of a count file per sample with two tab delimited columns without the header.

oystercontig_1	301
oystercontig_2	8
oystercontig_3	70
oystercontig_4	0
oystercontig_5	0
oystercontig_6	2
oystercontig_7	123
oystercontig_8	375
oystercontig_9	0

Note
The user should provide the same number of read files inside a directory corresponding to the number of rows in the target file. If the counts and the target files are not supplied in the required formats, the app will not work and you will not be able to run the analysis.

4. Parameters

- `Project name`: name of the project (must be supplied by the user);
- `Author Name`: author of the analysis (must be supplied by the user);
- `Reference biological condition`: reference biological condition used to compute fold-changes (no default, must be one of the levels target file);
- `batch`: adjustment variable to use as a batch effect, must be a column of the target file (`"day"` for example, or`NULL` if no batch effect needs to be taken into account);
- `Variable of Interest`: variable of interest, i.e. biological condition, in the target file (Mandatory. `"group"` by default);
- `FeaturesToRemove`: character vector containing the IDs of the features to remove before running the analysis (default is,`"alignment_not_unique"). Other available features are"ambiguous""no_feature"`, `"not_aligned"`, `"too_low_aQual"` to remove HTSeq-count specific rows);
- `locfunc`: function used for the estimation of the size factors (default is,`"median"` or `"shorth"`from the gene filter` package);
- `Transfomation method for PCA/clustering`: method of transformation of the counts for the clustering and the PCA (default is`"VST"` for Variance Stabilizing Transformation, or `"rlog"` for Regularized Log Transformation);
- `Mean-variance relationship`: type of model for the mean-dispersion relationship (`"parametric"` by default, or`"local"`);
- `Independent Filtering`: TRUE (default) or FALSE to execute or not the independent filtering;
- `cooksCutoff`: TRUE (default) or FALSE to execute or not the detection of the outliers;
- `Significance threshold`: significance threshold applied to the adjusted p-values to select the differentially expressed features (default is `0.05`);
- `p-value adjustment method`: p-value adjustment method for multiple testing (`"BH"` by default, `"BY"` or any value of `p.adjust.methods`);
- `colors`: colors used for the figures (one per biological condition)

All these parameters will be saved and written at the end of the HTML report in order to keep track of what has been done.

Test/sample data

This tutorial uses the test data that is stored in the Data Store at Community Data > iplantcollaborative > example_data > DESeq2_multi.

Starting a DESeq2 (multifactorial pairwise comparisons) job in the DE

Open the DE Apps window and search for edgeR (multifactorial pairwise comparisons).

In the Analysis Name:

1. Change the name for your analysis (optional).
2. Enter any comments (optional).
3. In the **Select output folder** field, click **Browse** and navigate to the folder of your choice. You can leave the default name iplant/home/username/analyses.
4. To retain copies of the input files in your analysis results output folder, click the **Retain Inputs** checkbox.

Click the Input files panel:

1. If you want to test the file_type test data:
    a. For the Target file, browse to select **target3.txt** inside file_type.
    b. For the Row counts file, browse to select **counts3.txt**.
2. If you want to test the folder_type test data:
    a. For the Target file, browse to select **target3.txt** inside file_type
    b. For the Raw counts folder, browse to select **raw1 inside** folder_type
3. Please note: Only one of the above two options need to be selected

Click on the Parameters panel:

1. `Project name`: test_deseq2_file_type
2. `Author Name`: Upendra
3. `Reference biological condition`: OP
4. `batch`:
5. `Variable of Interest`: group
6. `FeaturesToRemove`: alignment_not_unique,ambiguous,no_feature,not_aligned,too_low_aQual
7. locfunc: median
8. `Transformation method for PCA/clustering: VST`
9. `Mean-variance relationship: parametric`
10. `Independent Filtering: TRUE`
11. `Cooks Cutoff: TRUE`
12. `Significance threshold`: `0.05`
13. `p-value adjustment method`: BH

14. `colors`: dodgerblue,orange,green

Click **Launch Analysis**.

Output from DEseq2 (multifactorial pairwise comparisons) app:

The following files and figures are generated

- `barplotTC.png`: the total number of reads per sample;
- `barplotNull.png`: percentage of null counts per sample;
- `densplot.png`: estimation of the density of the counts for each sample;
- `majSeq.png`: percentage of reads caught by the feature having the highest count in each sample;
- `pairwiseScatter.png`: pairwise scatter plot between each pair of samples and SERE values (not produced if more than 30 samples);
- `diagSizeFactorsHist.png`: diagnostic of the estimation of the size factors;
- `diagSizeFactorsTC.png`: plot of the size factors vs the total number of reads;
- `countsBoxplot.png`: boxplots on raw and normalized counts;
- `cluster.png`: hierarchical clustering of the samples (based on VST or rlog data for DESeq2);
- `PCA.png`: first and second factorial planes of the PCA on the samples based on VST or rlog data;
- `dispersionsPlot.png`: graph of the estimations of the dispersions and diagnostic of log-linearity of the dispersions;
- `rawpHist.png`: histogram of the raw p-values for each comparison;
- `MAplot.png`: MA-plot for each comparison (log ratio of the means vs intensity);
- `volcanoPlot.png`: volcano plot for each comparison ($-\log_{10}\text{(adjusted P value)}$ vs log ratio of the means).

Some tab-delimited files are exported in the directory `tables`. They store information on the features as $\log_2\text{(FC)}$ or p-values and can be read easily in a spreadsheet:

- `TestVsRef.complete.txt`: contains all the features studied;
- `TestVsRef.down.txt`: contains only significant down-regulated features, i.e. less expressed in Test than in Ref;
- `TestVsRef.up.txt`: contains only significant up-regulated features i.e. more expressed in Test than in Ref.

For more information of how to interpret these figures, files, troubleshooting and FAQ please refer here

All these parameters will be saved and written at the end of the HTML report in order to keep track of what has been done.