EBSeq: An R package for differential expression analysis using RNA-seq data

Galaxy Manual

Jeea Choi and Ning Leng

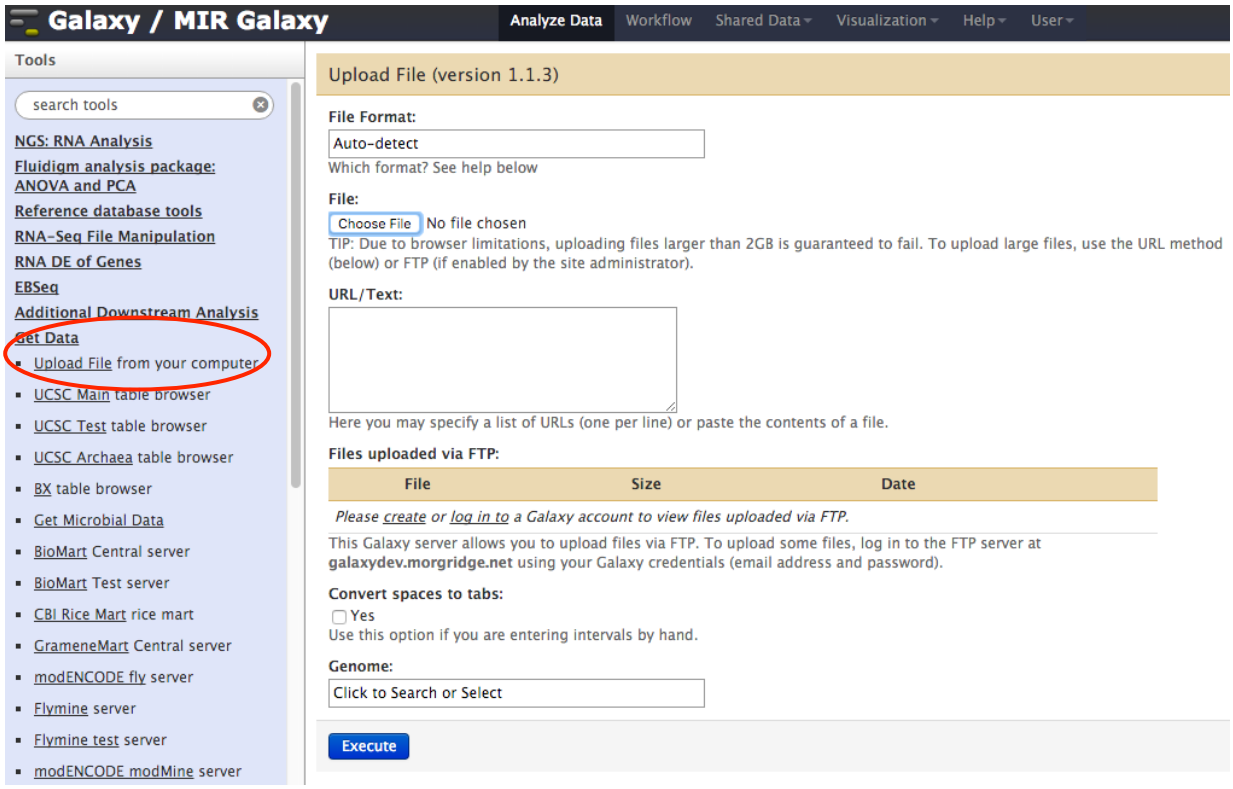
Table of Contents

1. Galaxy

The empirical Bayes model in Leng et al., 2013 is implemented in Galaxy (http://galaxy.morgridge.net/). This manual is a guideline for using the galaxy EBSeq interface, which will allow a user to run EBSeq without directly using R. Files can be uploaded in tab delimited format. Guideline for two condition gene DE analysis can be found in section 3, guidelines for multiple condition gene DE analysis can be found in section 5-7, and guidelines for two/multiple condition isoform DE analysis can be found in section 9/10.

2. Preparation for the analysis

As shown below, upload the file



3. Gene level DE test across two conditions

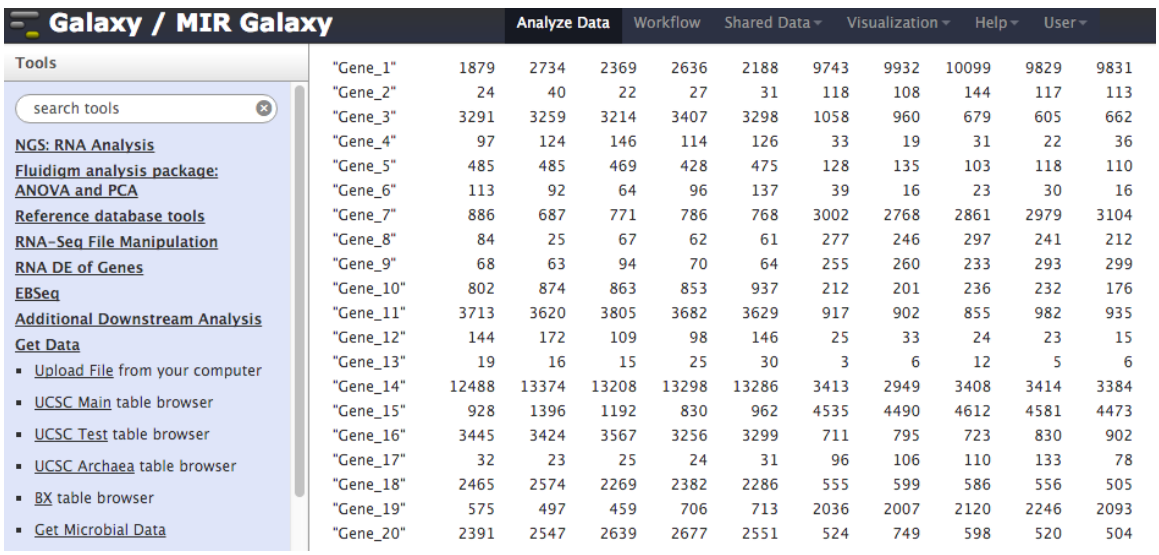
Input requirement:

The input file formats supported by EBSeq are .csv, .tab, or .txt (tab delimited). In your input file, the rows should be the genes either with or without column names. In other words, the first column shows your gene names.

Note: This example does not use isoform level expression data.
An example of isoform expression analysis is shown in Section 9.

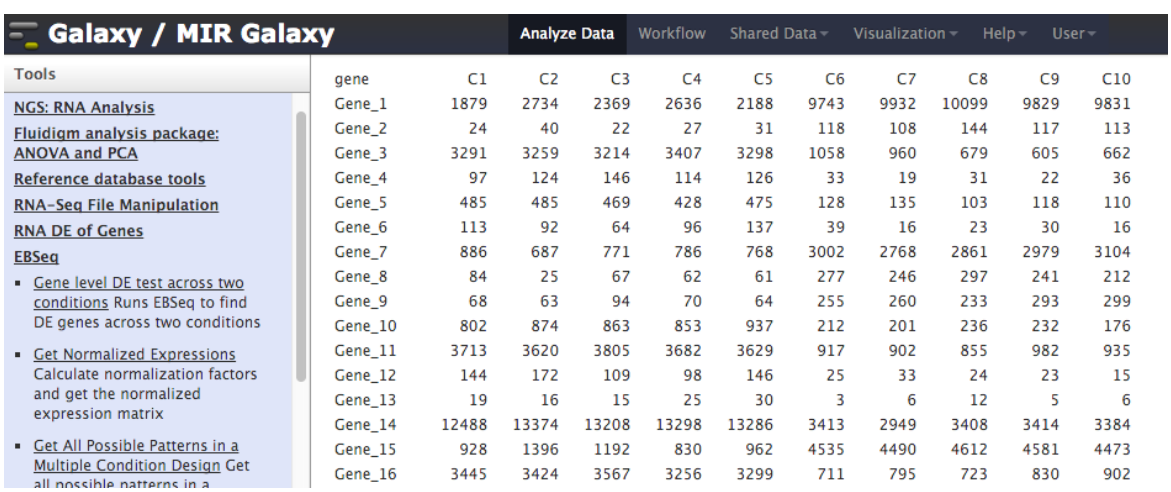
Example data set in .csv format:

GeneMat.csv (without column names): we will use this to show following screenshot



"Gene_1"	1879	2734	2369	2636	2188	9743	9932	10099	9829	9831
"Gene_2"	24	40	22	27	31	118	108	144	117	113
"Gene_3"	3291	3259	3214	3407	3298	1058	960	679	605	662
"Gene_4"	97	124	146	114	126	33	19	31	22	36
"Gene_5"	485	485	469	428	475	128	135	103	118	110
"Gene_6"	113	92	64	96	137	39	16	23	30	16
"Gene_7"	886	687	771	786	768	3002	2768	2861	2979	3104
"Gene_8"	84	25	67	62	61	277	246	297	241	212
"Gene_9"	68	63	94	70	64	255	260	233	293	299
"Gene_10"	802	874	863	853	937	212	201	236	232	176
"Gene_11"	3713	3620	3805	3682	3629	917	902	855	982	935
"Gene_12"	144	172	109	98	146	25	33	24	23	15
"Gene_13"	19	16	15	25	30	3	6	12	5	6
"Gene_14"	12488	13374	13208	13298	13286	3413	2949	3408	3414	3384
"Gene_15"	928	1396	1192	830	962	4535	4490	4612	4581	4473
"Gene_16"	3445	3424	3567	3256	3299	711	795	723	830	902
"Gene_17"	32	23	25	24	31	96	106	110	133	78
"Gene_18"	2465	2574	2269	2382	2286	555	599	586	556	505
"Gene_19"	575	497	459	706	713	2036	2007	2120	2246	2093
"Gene_20"	2391	2547	2639	2677	2551	524	749	598	520	504

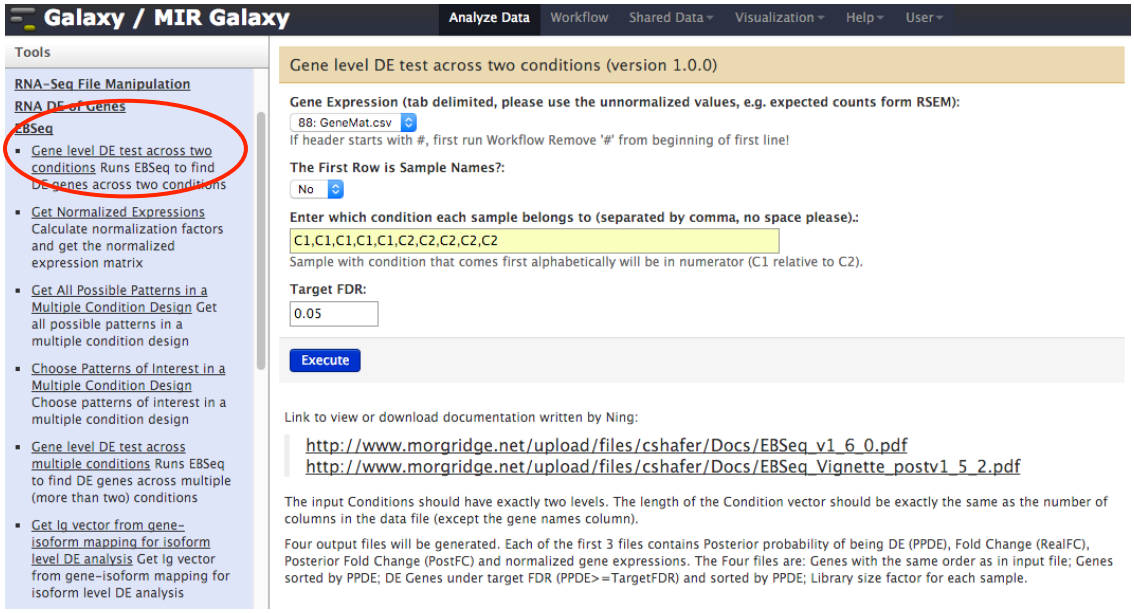
GeneMat.csv (with column names): It fails if the first cell from the first row is empty



gene	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Gene_1	1879	2734	2369	2636	2188	9743	9932	10099	9829	9831
Gene_2	24	40	22	27	31	118	108	144	117	113
Gene_3	3291	3259	3214	3407	3298	1058	960	679	605	662
Gene_4	97	124	146	114	126	33	19	31	22	36
Gene_5	485	485	469	428	475	128	135	103	118	110
Gene_6	113	92	64	96	137	39	16	23	30	16
Gene_7	886	687	771	786	768	3002	2768	2861	2979	3104
Gene_8	84	25	67	62	61	277	246	297	241	212
Gene_9	68	63	94	70	64	255	260	233	293	299
Gene_10	802	874	863	853	937	212	201	236	232	176
Gene_11	3713	3620	3805	3682	3629	917	902	855	982	935
Gene_12	144	172	109	98	146	25	33	24	23	15
Gene_13	19	16	15	25	30	3	6	12	5	6
Gene_14	12488	13374	13208	13298	13286	3413	2949	3408	3414	3384
Gene_15	928	1396	1192	830	962	4535	4490	4612	4581	4473
Gene_16	3445	3424	3567	3256	3299	711	795	723	830	902

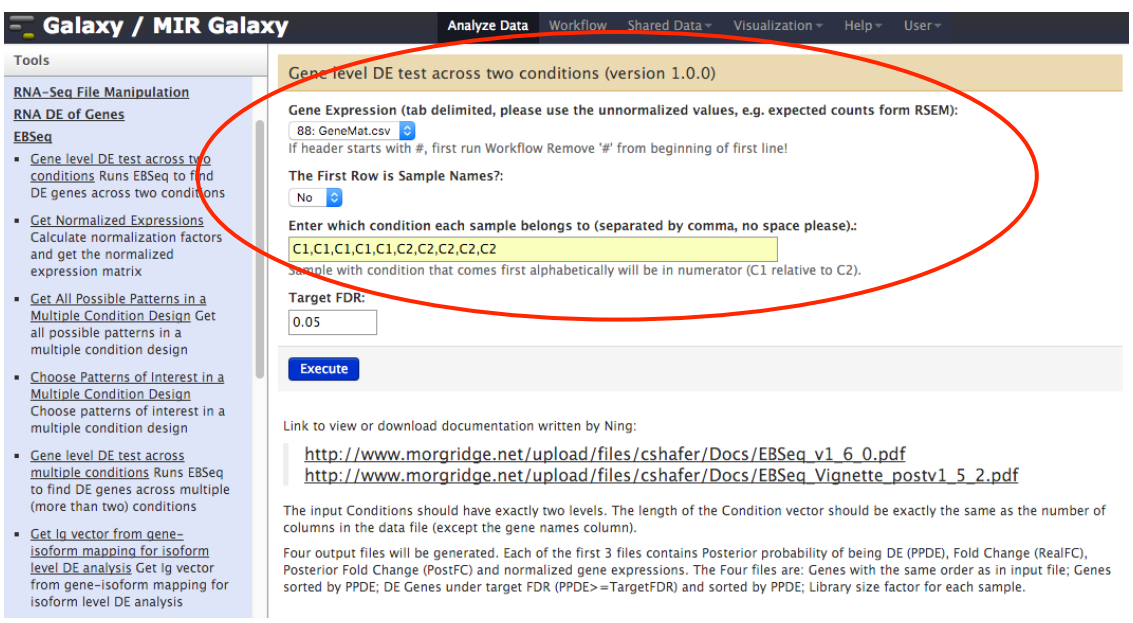
EBSeq in Galaxy:

Go to EBSeq and click Gene level DE test across two conditions



Next, a user can customize:

1. The input file. All uploaded files will be shown to choose input file.
2. Select whether the first row is sample names. "No" in this example.
3. Enter which condition each sample belongs to: The number of typed condition separated by comma (,) needs to be matched the number of columns in GeneMat.csv. Here we call the first condition as C1 and the second condition as C2. In this example, the first 5 samples are from condition 1 and the other 5 are from condition 2.
4. Target false discovery rate (FDR); the default is 0.05
5. Press "Execute" button



Explaining the Outputs

Five files will be generated for GeneMat.csv example (screenshot below):

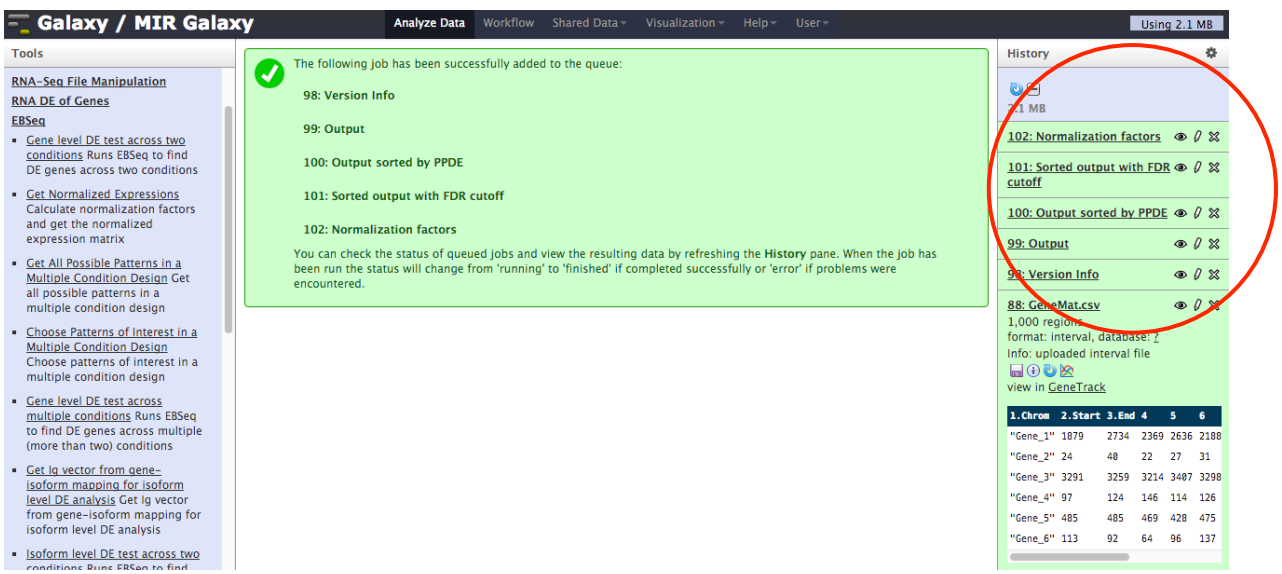
(1) Normalization factors: estimated library size for each sample that is used for median-by-ratio normalization.

(2) Sorted output with FDR cutoff: Columns are posterior probability of being DE (PPDE), Fold Change (RealFC), posterior Fold Change (PosteriorFC), and median-by-ratio normalized gene expressions (cell orders are the same with input cell order). Only genes with PPDE >= 1 - Target_FDR are listed. The RealFC calculates (mean_condition1 + 0.01)/(mean_condition2 + 0.01), in which the within condition mean is calculated using normalized data. The posterior fold changes are estimated from the empirical bayes model. The posterior FC estimations will give less extreme values for low expressers. e.g. if gene1 has mean1 = 5000 and mean2 = 1000, its FC and PostFC will both be 5. If gene2 has mean1 = 5 and mean2 = 1, its FC will be 5 but its PostFC will be < 5 and closer to 1. Therefore, when we sort the PostFC, gene2 will be less significant than gene1.

(3) Output sorted by PPDE: Columns are the same as in (1). Genes are sorted decreasingly by PPDE.

(4) Output: Columns are the same as in (1) and (2). Rows are the genes in the same order as the input file.

(5) Version Info: Shows EBSeq version info. Input parameters (e.g. FDR chosen, list of conditions) are shown.



4. Get Normalized Expressions

Normalized expressions can be obtained using the 'Get Normalized expressions' module without performing DE analysis (Section 3).

Input requirement:

The input file formats are similar to that of Section 3.

Explaining the Outputs

Three files will be generated for GeneMat.csv example:

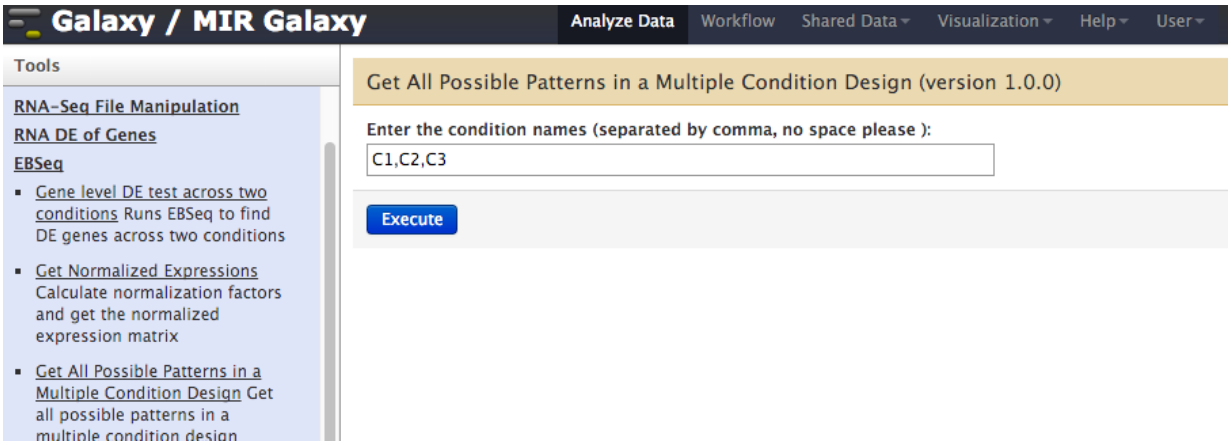
(1) Normalization factors: Library size for each sample that is used for median-by-ratio normalization
(2) Boxplots: Boxplot of normalized expression for each sample. Y axis shows normalized expression values.
(3) Normalized expression: median-by-ratio normalized (library size adjusted) gene expressions

5. Get All Possible Patterns in a Multiple Condition Design

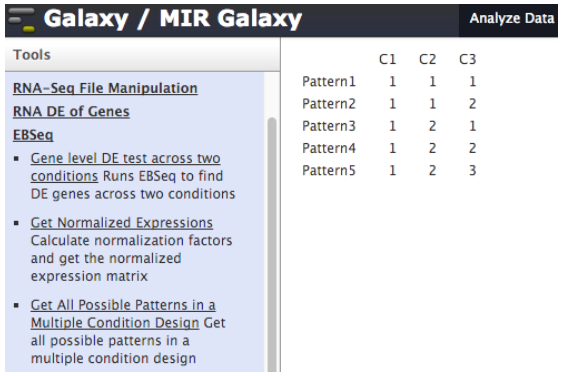
In a two condition DE analysis, a gene can only be either DE or EE. In a case with more than 2 conditions, more than multiple patterns are possible. For example, in a case with 3 conditions, there are total 5 possible patterns: C1 = C2 = C3, C1 = C2 ≠ C3, C1 ≠ C2 = C3, C2 ≠ C1 = C3, C1 ≠ C2 ≠ C3. Before performing DE test across multiple conditions, we need to construct the possible patterns, first. This can be done by the "Get All Possible Patterns" module.

Input:

Enter the condition names: Comma separated multiple conditions need to be typed



Explaining the Outputs



Output shows the possible patterns in a multiple condition design

For example, the first pattern is C1 = C2 = C3
the second pattern is C1 = C2 ≠ C3

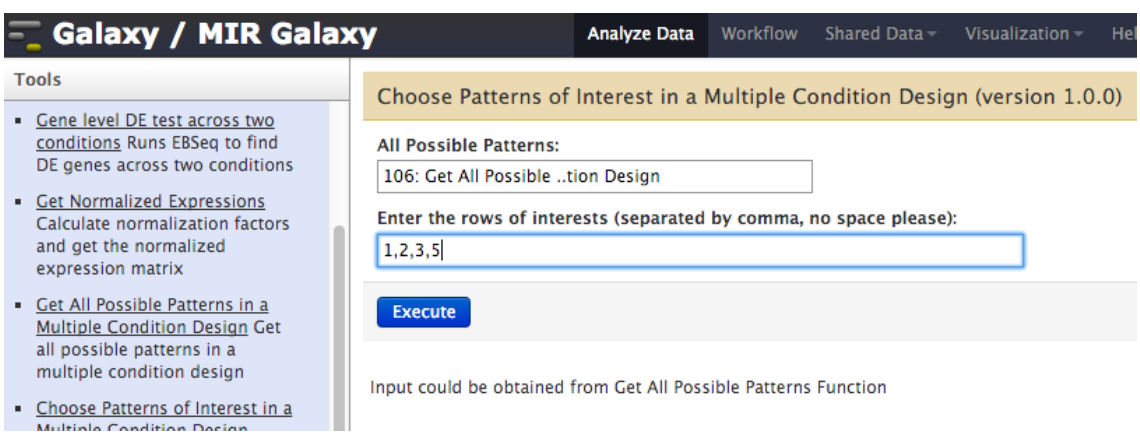
For data sets with more than 3 conditions, the number of possible patterns increase exponentially. To reduce the runtime and to make the output more easily interpreted, we suggest the user to choose a small set of patterns that related to the experimental hypothesis (Note we suggest the user to always include the all EE pattern and all DE pattern, usually the first pattern and the last pattern).

A user can choose patterns that is of interest directly from this output before continuing on to further analysis (Section 6)

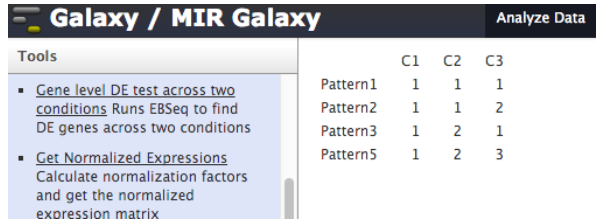
6. Choose Patterns of interest in a Multiple Condition Design

Input:

Use the output from Section 5 and enter the rows of interests. If we are interested in identifying genes that follows pattern 2, 3, and 5 but not 4, we can type "1,2,3,5". Note it is always suggested to include the all EE pattern (the first one here) and the all DE pattern (the last one) as background patterns. (screenshot below)



Output:
Output shows the chosen patterns.



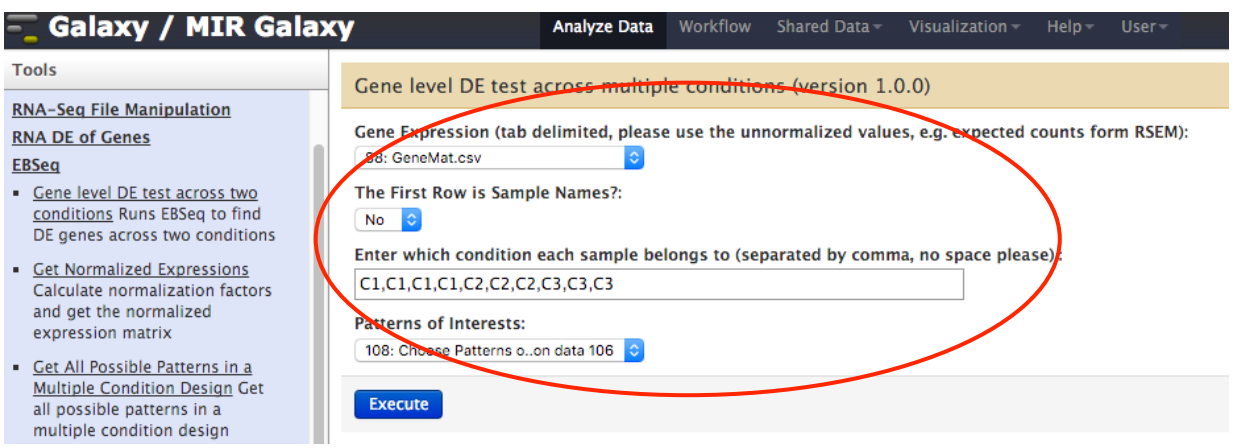
7. Gene level DE test across multiple Condition Design

Input:

The input file formats are similar to that of Section 3.

Next, a user can customize:

1. The input file. All uploaded files will be shown to choose input file.
2. Select whether the first row is sample names. "No" in this example.
3. Enter which condition each sample belongs to: The number of typed condition separated by comma (,) needs to be matched by the number of columns in GeneMat.csv
4. Patterns of interest. Constructed patterns from Section 5 or Section 6 can be chosen.
5. Press "Execute" button.



Explaining the Outputs

Four files will be generated for GeneMat.csv example (screenshot below):

(1) Normalization factors: Library size for each sample that is used for median-by-ratio normalization

(2) Pattern with highest PP: Column 1 shows the pattern with the highest posterior probability for each gene (MAP). The other columns are the median-by ratio normalized gene expressions. Rows are the genes with the same order as input.

(3) PP of each pattern: Columns are posterior probability of being each pattern. Rows are the genes with the same order as input. The higher the PP is, the more likely that this gene is following this specific pattern.

(4) Version Info: Shows EBSeq version info. Input parameters (e.g. FDR chosen, list of conditions) are shown.

8. Get Ig vector from gene-isoform mapping for isoform level DE analysis

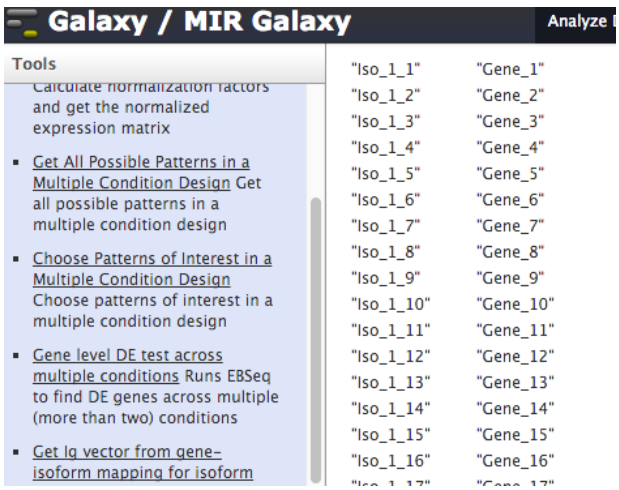
For isoform level analysis, an *Ig* vector is required (see Leng *et al*., 2013, or the EBSeq vignette for details on *Ig*). If you have the *Ig* vector file generated from RSEM, please ignore this subsection.

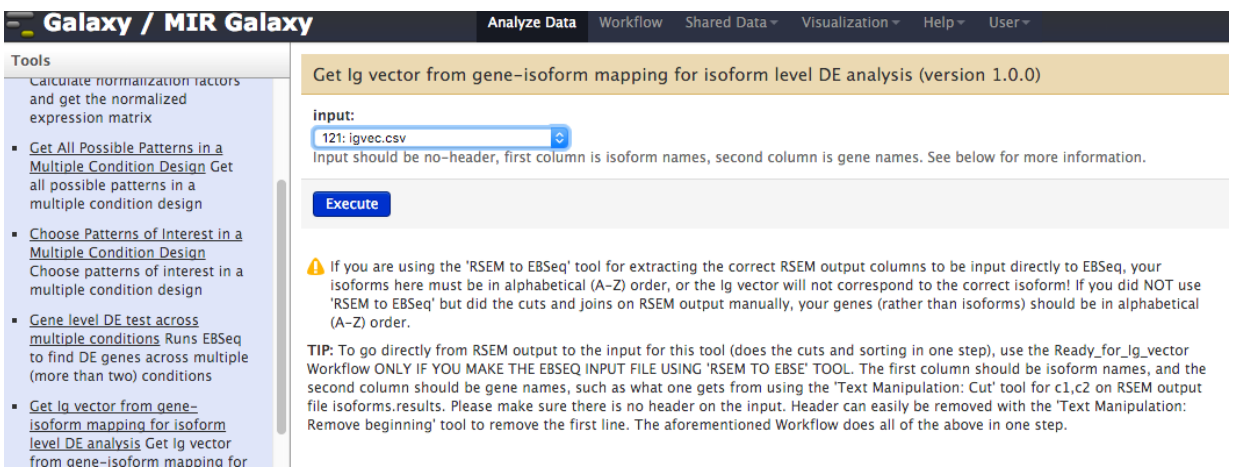
Input:

Again, csv, xls, or xlsx files are accepted. The first column specifies the isoform names and the second column specifies the corresponding gene names.

Example data set in .csv format:

igvec.csv





Explaining the Outputs

"Ig vector" file will be generated

9. Isoform level DE test across two conditions

Input:

The *Ig* vector file from Section 4.1 or RSEM rsem-generate-ngvector function (http://deweylab.biostat.wisc.edu/rsem/rsem-generate-ngvector.html).

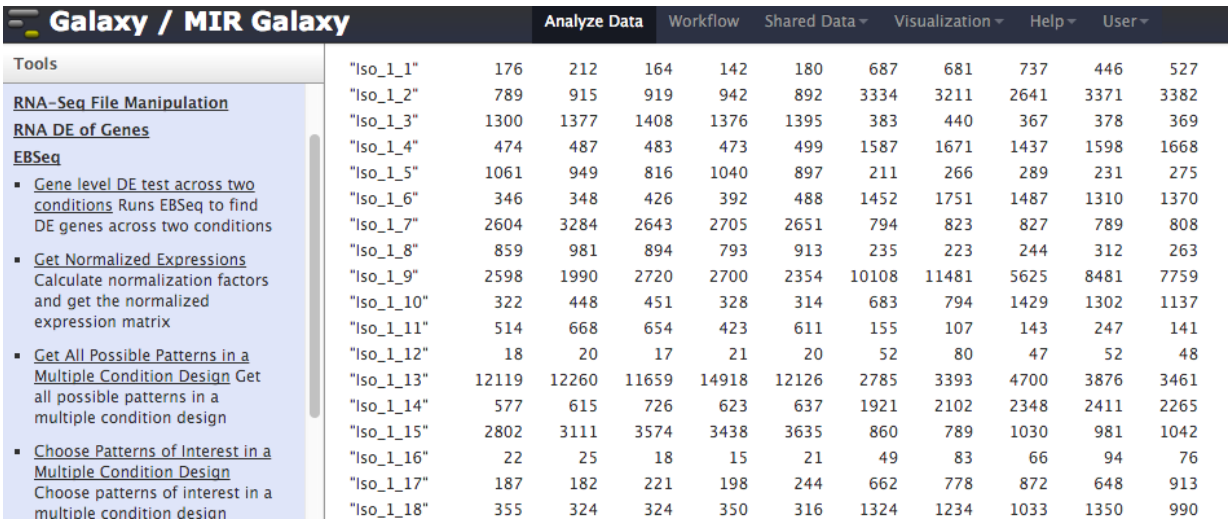
The data input could be .csv, .txt, or .tab files (tab delimited).

Rows are isoforms and columns are samples.

- The first column shows the isoform names

Example data set

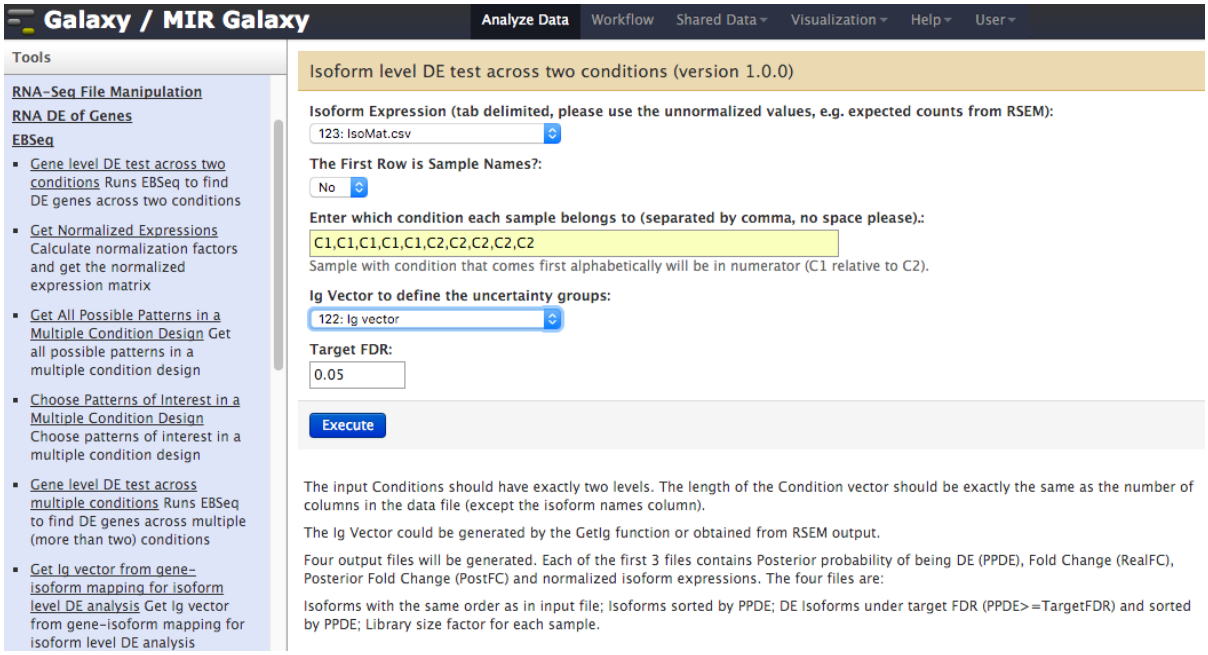
IsoMat.csv



"Iso_1_1"	176	212	164	142	180	687	681	737	446	527
"Iso_1_2"	789	915	919	942	892	3334	3211	2641	3371	3382
"Iso_1_3"	1300	1377	1408	1376	1395	383	440	367	378	369
"Iso_1_4"	474	487	483	473	499	1587	1671	1437	1598	1668
"Iso_1_5"	1061	949	816	1040	897	211	266	289	231	275
"Iso_1_6"	346	348	426	392	488	1452	1751	1487	1310	1370
"Iso_1_7"	2604	3284	2643	2705	2651	794	823	827	789	808
"Iso_1_8"	859	981	894	793	913	235	223	244	312	263
"Iso_1_9"	2598	1990	2720	2700	2354	10108	11481	5625	8481	7759
"Iso_1_10"	322	448	451	328	314	683	794	1429	1302	1137
"Iso_1_11"	514	668	654	423	611	155	107	143	247	141
"Iso_1_12"	18	20	17	21	20	52	80	47	52	48
"Iso_1_13"	12119	12260	11659	14918	12126	2785	3393	4700	3876	3461
"Iso_1_14"	577	615	726	623	637	1921	2102	2348	2411	2265
"Iso_1_15"	2802	3111	3574	3438	3635	860	789	1030	981	1042
"Iso_1_16"	22	25	18	15	21	49	83	66	94	76
"Iso_1_17"	187	182	221	198	244	662	778	872	648	913
"Iso_1_18"	355	324	324	350	316	1324	1234	1033	1350	990

A user can customize:

1. The input file. All uploaded files will be shown to choose input file.
2. Select whether the first row is sample names. "No" in this example.
3. Enter which condition each sample belongs to: The number of typed condition separated by comma (,) needs to be matched the number of columns in IsoMat.csv
4. The name for the *Ig* vector file.
5. Target false discovery rate (FDR); the default is 0.05



Explaining the Outputs

Five files will be generated:

(1) Normalization factors: Library size for each sample that is used for median-by-ratio normalization.
(2) Sorted output with target FDR: Columns are posterior probability of being DE (PPDE), Fold Change (RealFC), posterior Fold Change (PosteriorFC), and median-by ratio normalized gene expressions. Only genes with PPDE >= 1 - Target_FDR are listed.
(3) Output sorted by PP: Columns are the same as in (1). Isoforms are sorted decreasingly by PPDE
(4) Output: Columns are the same as in (1) and (2). Rows are the isoforms in the same order as the input file
(5) Version Info: Shows EBSeq version info. Input parameters (e.g. FDR chosen, list of conditions) are shown.

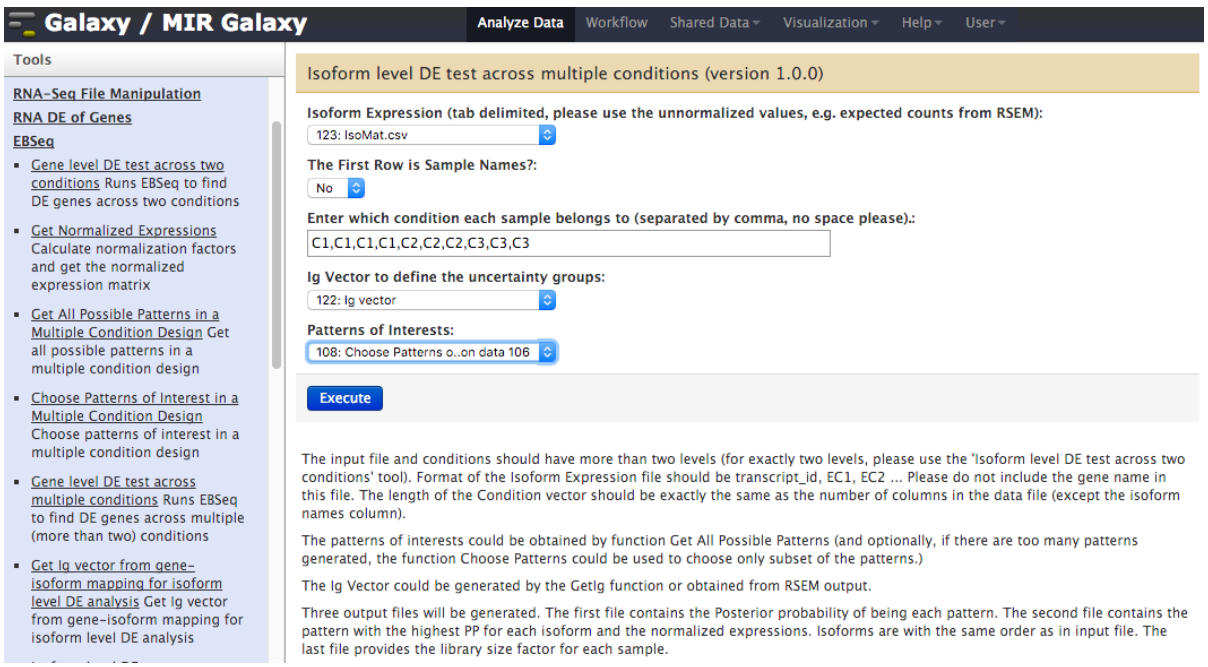
10. Isoform level DE test across multiple conditions

Input:

The input file formats are similar to that of Section 9.

Next, a user can customize:

1. The input file. All uploaded files will be shown to choose input file.
2. Select whether the first row is sample names. "No" in this example.
3. Enter which condition each sample belongs to: The number of typed condition separated by comma (,) needs to be matched the number of columns in IsoMat.csv
4. The name for the *Ig* vector file.
5. Pattern of interests. Constructed patterns from Section 5 or Section 6 can be chosen.
6. Press "Execute" button.



Explaining the Outputs

Four files will be generated for IsoMat.csv example (screenshot below):

(1) Normalization factors: Library size for each sample that is used for median-by-ratio normalization

(2) Pattern with highest PP: Column 1 shows the pattern with the highest posterior probability for each isoform (MAP). The other columns are median-by-ratio normalized isoform expressions. Rows are the isoforms with the same order as input.

(3) PP of each pattern: Columns are posterior probability of being each pattern. Rows are the isoforms with the same order as input.

(4) Version Info: Shows EBSeq version info. Input parameters (e.g. FDR chosen, list of conditions) are shown.

11. Trouble shooting

More details of the EBSeq implementation can be found at http://www.biostat.wisc.edu/~kendzior/EBSEQ/EBSeq_Vignette.pdf.

If you have additional questions not addressed in this manual regarding the EBSeq interface, please see the Q&A section on the EBSeq website biostat.wisc.edu/~kendzior/EBSEQ, or contact us at sswanson@morgridge.org.

Format problem: It fails if the first cell from the first row is empty.

Reference:

Leng, N., J.A. Dawson, J.A. Thomson, V. Ruotti, A.I. Rissman, B.M.G. Smits, J.D. Haag, M.N. Gould, R.M. Stewart, and C. Kendziorski. EBSeq: An empirical Bayes hierarchical model for inference in RNA-seq experiments, *Bioinformatics*, [e-pub ahead of print 21 February 2013] [Download].

Li, B., V. Ruotti, R.M. Stewart, J.A. Thomson, and C. Dewey. RNA-Seq gene expression estimation with read mapping uncertainty. *Bioinformatics* 26(4): 493-500, 2010.