**Access to AltAnalyze scripts**

For new labs using AltAnalyze: Download AltAnalyze scripts from github to get all of the pre- and post-processing scripts:

* <https://github.com/nsalomonis/altanalyze>

For Salomonis lab members, all necessary scripts are located in: */Volumes/salomonis2/software*

***NOTE:*** AltAnalyze currently uses python 2.7. When running any of these scripts on the cluster, specifically module load **python/2.7.5** ! Migration into python 3 is currently ongoing.

**Generate splice junction information from bam files**

Run *AltAnalyze/import\_scripts/****BAMtoJunctionBED.py*** and *AltAnalyze/import\_scripts/****BAMtoExonBED.py*** on each of your input bam files.

Specify the input bam file (--i), species (--species), and path to the appropriate Ensemble exon reference file (--r).

Ex:

* python /data/salomonis2/software/AltAnalyze/import\_scripts/BAMtoJunctionBED.py --i *[bam file]* --species Hs --r /data/salomonis2/software/AltAnalyze-91/AltAnalyze/AltDatabase/EnsMart91/ensembl/Hs/Hs\_Ensembl\_exon.txt
* python /data/salomonis2/software/AltAnalyze/import\_scripts/BAMtoExonBED.py --i *[bam file]* --r /data/salomonis2/Genomes/exonrefdir\_hg38/Hs.bed --s Hs

The outputs of these scripts, per bam file, include:

* Bam index file (bam.bai)
* Junction annotation file (\_junction.bed)
* Intron annotation file (\_intronJunction.bed)

The two different junction.bed files provide genomic coordinates, read counts for unique splice junctions. The junction.bed file contains information on known exon-exon splice juntions. The \_intronJunction.bed file will contain information instances of retained introns and novel splice junctions.

Bash scripts to process all bam files within the same directory:

* hg19: *salomonis2/LabFiles/Tutorials/RunAltAnalyze.from.bams/****index-junction\_hg19.sh***
* hg38: *salomonis2/LabFiles/Tutorials/RunAltAnalyze.from.bams/****index-junction\_hg38.sh***

**Create Accessory Files**

1. Generate “groups” file:

The groups file specifies which experimental “group” is associated with each bam file. This is a 3-column tab-delimited text file containing the following information:

* Column 1: The name of your bam file. ***NOTE:*** Change the file name from Name.bam to Name.bed. There isn’t a particular reason for this, but just do it anyway to avoid an error when running AltAnalyze.
* Column 2: The number associated with each group. Start with “1” and continue until you go through each of your experimental groups. It is recommended to have your control group(s) be the largest number(s) in your groups files.
* Column 3: The shorthand name for this group.

Example groups file layout:

Example\_1.bed 1 ExperimentalCondition\_1

Example\_2.bed 1 ExperimentalCondition \_1

Example\_3.bed 2 ExperimentalCondition \_2

Example\_4.bed 2 ExperimentalCondition\_2

Example\_5.bed 3 Control

Example\_6.bed 3 Control

If there are no comparison groups of interest (e.g., processing all cancer patient samples with no controls), then randomly assign half of the dataset as “1” and the other half as “2”. Generate whatever shorthand of choice for the third column.

***NOTE:*** You cannot have one bam file assigned to more than one group at the same time. If there are multiple ways to bin your samples (e.g., male vs. females; mutant vs. wt), you will need to run those additional analyses separately using different a different groups/comps file.

1. Generate “comps” file:

The comps file will assign your comparisons of interest. Specifically, with dictate your input groups for the differential splicing and gene expression analysis done directly after completion of the splicing annotation file.

Comparison groups are denoted based on the assigned number in your groups file. This file is a two-column text file. The first column specifies the group of interest, while the second column species the comparison group. For example, if you wanted to compare ExperimentalCondition\_1 vs. Control, your comps file would look like the following:

1 3

**The order of your groups into the first or second column matters.** Make sure your control group is **always** in the second column.

You can set up multiple comparisons in the same comps file by adding a different comparison into a new row. For example, to do ExperimentalCondition\_1 vs. Control and ExperimentalCondition\_2 vs. Control, set up the following comps file:

1. 3
2. 3
3. Create a new folder labeled “ExpressionInput”. Then place your “groups” and “comps” files within this new folder. It’s simplest to place the “ExpressionInput” folder within your bam/bed directory, but this is not required to run AltAnalyze.

**Run AltAnalyze**

Run AltAnalyze/**AltAnalyze.py**. Set the following parameters:

--species (e.g Hs )

--platform (e.g. RNASeq )

--version (e.g. EnsMart91 )

--bedDir Directory of your junction bed/bam files

--groupdir Filepath for your groups.txt file

--compdir Filepath for your comps.txt file

--expname Shorthand experiment name

--runGOElite This will run a Gene Ontology analysis using AltAnalyze databases (e.g. no)

Bash scripts to process all bam files within the same directory:

* hg19: *salomonis2/LabFiles/Tutorials/RunAltAnalyze.from.bams/* ***AltAnalyze.sh***
* hg38: *salomonis2/LabFiles/Tutorials/RunAltAnalyze.from.bams/* ***AltAnalyze-91.sh***

**QC Results**

First, check the log files to make sure AltAnalyze completed successfully. AltAnalyze can stall during generation of the splicing matrix file and have the job time out with a partially generated splicing annotation file. This is not inherently obvious when just looking at the output directories.

The DataPlots folder will contain four important QC plots:

1. PCA based on gene expression data (Clustering-*experiment name*-PCA.pdf)

**What to look for:** If you have experimental groups, your replicates in a given group (e.g. Control RepA; Control RepB) should cluster together. Outlier samples that don’t cluster with the other replicates could indicate batch effects.

1. Counts distribution (QC-counts. *experiment name*-distribution.pdf)

**What to look for:** All the samples should have a similar read count distribution

1. Splice junction counts per sample

(QC-counts.*experiment name*-BoxPlot-junction.pdf)

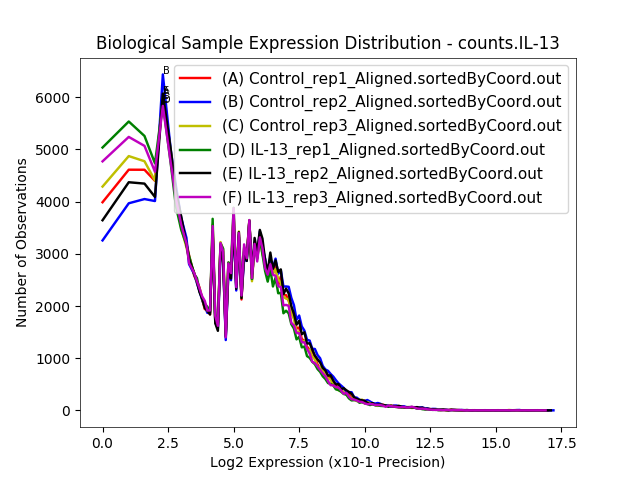
**What to look for:** Similar to the gene counts, all the samples should be in the same range for the total transcript feature counts. Samples with very low counts may have very sparse splicing information.

1. Total transcript feature counts

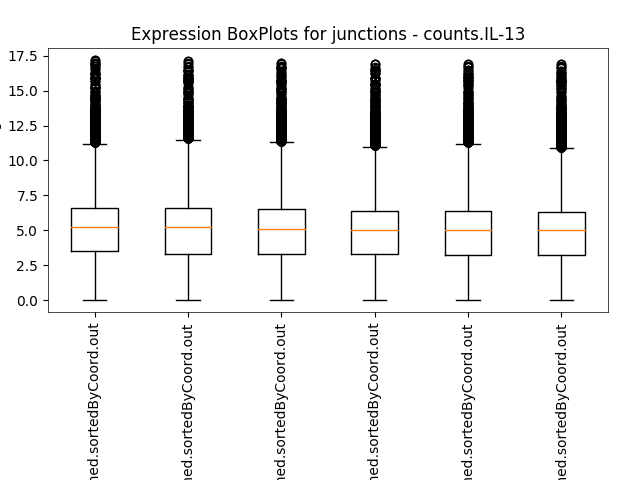
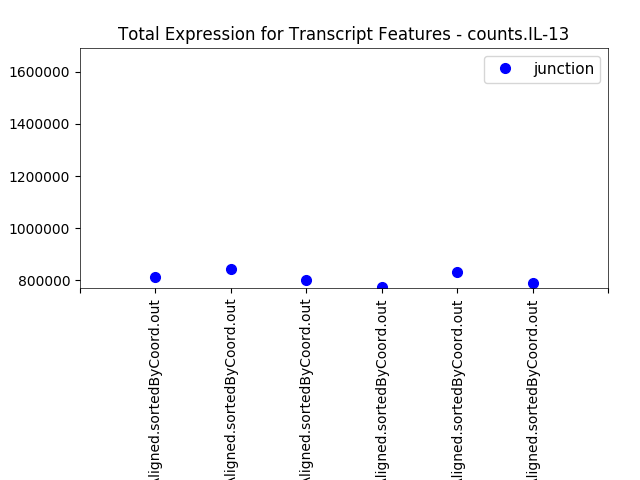
(QC-counts. *experiment name*-TotalFeatureExpression.pdf)

**What to look for:** Similar to the gene counts, all the samples should be in the same range for the total transcript feature counts. Samples with very low counts may have very sparse splicing information.

Example: gene expression QC plots



Example: splicing QC plots



**Splicing Output Results**

The */AltResults/AlternativeOutput* directory will contain the splicing annotation matrix and calculated PSI values (Hs\_RNASeq\_top\_alt\_junctions-PSI\_EventAnnotation.txt).

Additionally, results of the differential splicing analysis using the default parameters used by AltAnalyze. These default parameters include:

* dPSI (change in percent spliced in) cutoff of |.1| ***NOTE:*** This is standard for the field.
* raw p-value cutoff of .05
* Filtering out for splicing events that are missing in 25% or more of the samples, per group. ***NOTE:*** For small sample sizes (e.g. 3 replicates), this means that all samples much contain a given splicing event. This may be over restrictive with sample(s) that are sparse.

The output files in this directory contain:

* Events-dPSI: Directory contains a text file with the differential splicing events, the magnitude of the change (dPSI), the raw and adjusted p-value, and event coordinates.
* ExpressionProfiles: Directory contains a text file with the PSI values for the differentially spliced events across all samples included in this analysis.
* Two different graphs to visualize the types of splicing events, separated by inclusion and exclusion events.
* MarkerHeatmaps: Takes the top 50 correlated splicing events per group and generates a heatmap to view how different/similar the comparison groups are.

**Gene Expression Output Results**

The */ExpressionInput* directory will contain the output gene expression file (exp.IL-13-steady-state.txt).

**What Now?**

* I want to do more differential splicing analysis!

Run *AltAnalyze/stats\_scripts/****metaDataAnalysis.py*.** Set the platform to --p PSI. You can also change the missing-value filter from the default of with –percentExp. Ex) --percentExp 50 means that the splicing event must be expressed (i.e. not missing) in 50% of the samples. Reminder that a zero value is **not** the same as a missing value in the splicing world.

* I want to do more differential gene expression analysis!

Run *AltAnalyze/stats\_scripts/****metaDataAnalysis.py*** with the --p set to RNAseq.