ALTRE: vignette

Ewy Mathe, Elizabeth Baskin, and Rick Farouni

2016-07-21

Introduction

ALTRE (ALTered Regulatory Elements) is an R software package that streamlines and simplifies the analysis of data generated from genome-wide chromatin accessibility assays such as DNase-seq (a.k.a. DHS-seq), FAIRE-seq, and ATAC-seq.

Chromatin accessibility data maps the location of regulatory elements (REs) such as enhancers and promoters. REs are involved in regulating gene transcription – they control genes and pathways that can be investigated as putative therapeutic targets, or they may serve as targets themselves. Chromatin accessibility assays allows us to identify the location of regulatory regions genome-wide by identifying "open" chromatin (i.e. euchromatin). Identifying regulatory regions that differ between cell types, such as cancerous and noncancerous cell lines and tissues, holds promise for discovering new mechanisms involved in cellular development and disease progression. While assays for defining regulatory regions are well established, currently there are few workflows that guide researchers though the process of analysing data, starting with aligned reads and peak calls, all the way to obtaining meaningful results such as determining a set of putative pathways and genes for further investigation.

Workflow

The complete pipeline for data generated by genome-wide chromatin accessibility assays generally starts with the FASTQ files containing raw data of the sequencing experiments. The sequencing reads are then mapped into a reference genome using short read aligners (e.g. bowtie2 or STAR) returning aligned reads in the form of SAM/BAM file. The next step involves finding enriched peaks, genomic regions where we detect more sequencing reads than we would expect by chance. The output of peak calling software is a BED file format containing genomic regions of signal enrichment obtained from pooled and normalized data. The initial processing of the FASTQ file must be accomplished outside of ALTRE, as the R package requires BAM and BED files as inputs. The BED files we use to run the analysis in the vignette are ENCODE broadPeak BED files for DNase-seq experiments. These signal peaks represent DNaseI hypersensitive sites (DHSs), regions of accessible chromatin in which regulatory elements such as promoters and enhancers can be found.

ALTRE defines altered peaks using both binary data (presence/absence of peaks) and quantitative data (peak intensity). Moreover, the package includes functions that merge and annotate peaks (e.g. as promoters and enhancers), perform enrichment analysis for REs of interest, and provide visualization functions. The analysis can be conducted through the R command line or through an RShiny web interface for those who are not familiar with the R statistical language. This vignette further explains the purpose of the package and establishes the workflow.

Example data

To demonstrate the abilities of ALTRE, we have provided an example dataset (BAM and BED files) of cancerous (A549) and normal (SAEC) lung cell types downloaded from the ENCODE project. This dataset will guide users through the workflow of the package in this vignette. This vignette uses only a subset of the data corresponging to Chromosome 21. This data is available for download at https://mathelab.github.io/ALTREsampledata/

Start of pipeline

```
library(ALTRE)
```

Obtaining Reproducible peaks

In he first step, we try to identify consensus peaks among sample replicates – peaks that are present in all (or, less stringently, the majority) of sample replicates. The function *getConsensusPeaks* processes an input of two or more BED files per sample and returns a GRanges list. At least two BED replicates are required in order to differentiate sample noise from robust, reproducible peaks.

Many of the functions within ALTRE rely on sample metadata that is supplied to the workflow in a CSV file. First, we need to download all the necessary sample files for this analysis and then read in a CSV file that contains the meta-data about our samples. This CSV file is included in the bundle of subsetted data. Please modify the CSV file such that the file paths refer to your local directory.

```
csvfile <- loadCSVFile("/home/rick/Documents/AltreDataRepo/DNaseEncodeChr21.csv")
```

For the *getConsensusPeaks* function, we need to read in the BED files for analysis. The *loadBedFiles* function will only read the first three columns of the BED files corresponding to the *chr*, *start*, *stop* variables. Additional columns are allowed, but will be ignored.

```
samplePeaks <- loadBedFiles(csvfile)
samplePeaks
```

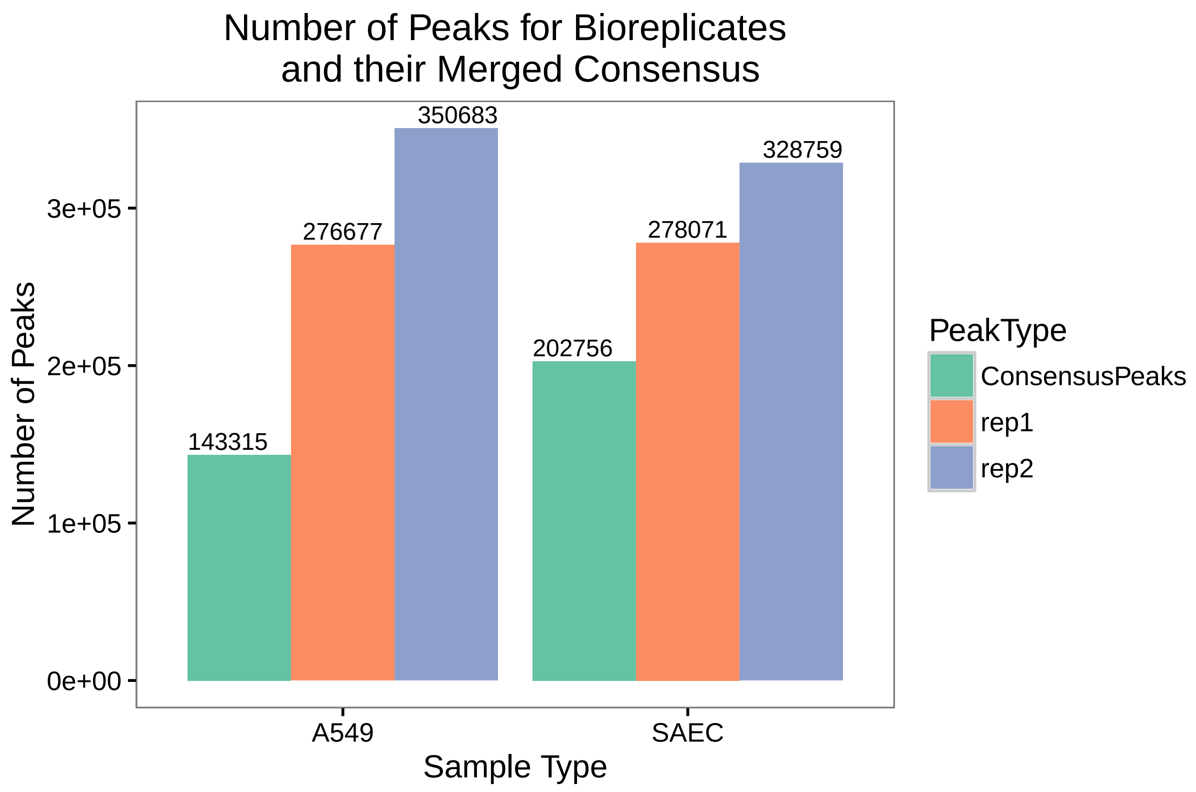
```
## GRangesList object of length 4:
## $A549_I
## GRanges object with 276677 ranges and 2 metadata columns:
##            seqnames               ranges strand |   sample replicate
##               <Rle>            <IRanges>  <Rle> | <factor>  <factor>
##        [1]     chr1     [ 10241,  10349]      * |     A549         I
##        [2]     chr1     [237719, 237872]      * |     A549         I
##        [3]     chr1     [564496, 564831]      * |     A549         I
##        [4]     chr1     [565253, 566084]      * |     A549         I
##        [5]     chr1     [566586, 567276]      * |     A549         I
##        ...      ...                  ...    ... .      ...       ...
##   [276673]     chrY [59024237, 59024354]      * |     A549         I
##   [276674]     chrY [59027624, 59027997]      * |     A549         I
##   [276675]     chrY [59028579, 59028819]      * |     A549         I
##   [276676]     chrY [59029626, 59029819]      * |     A549         I
##   [276677]     chrY [59031344, 59031507]      * |     A549         I
##
## ...
## <3 more elements>
## -------
## seqinfo: 24 sequences from an unspecified genome; no seqlengths
```

Finally, we use the *getConsensusPeaks* function to identify the peaks that are present in both replicates of each sample type.

```r
consensusPeaks <- getConsensusPeaks(samplepeaks = samplePeaks,
                                    minreps = 2)
```

The function *plotConsensusPeaks* creates a bar graph comparing the number of replicate peaks identified vs. total number of peaks.

```r
plotConsensusPeaks(samplepeaks = consensusPeaks)
```



Peak Annotation

The output of the function *getConsensusPeaks* is then further processed by the *combineAnnotatePeaks* function, which in turn accomplishes three main tasks:

1. Combines the reproducible peaks from both samples type into one large master list of peaks. This enables peak height comparison across sample types.
2. Categorize the peaks as either promoters or enhancers based on the distance of the RE from a transcription start site (TSS). In this vignette we will use the default promoter distance argument: REs within 1,500 bp of a TSS are considered promoters; those not within 1,500 bp of a TSS are considered enhancers.
3. Optionally, merges REs within a certain distance of each other. Merging close-by REs loosens the stringency of the region comparison. Multiple regions of active chromatin within ~1000 bp are likely to represent only one RE. In this vignette we will use the defaults of the function, which merges regions within 1000 bp of each other and merges only within RE type (i.e. only enhancers will only be merged with other enhancers, and likewise, promoters only with promoters). Both these functionalities can be changed via the "mergedistenh", "mergedistprom" and "regionspecific" arguments.

This function requires a list of TSSs for the promoter/enhancer annotation:

```r
TSSannot <- getTSS()
```

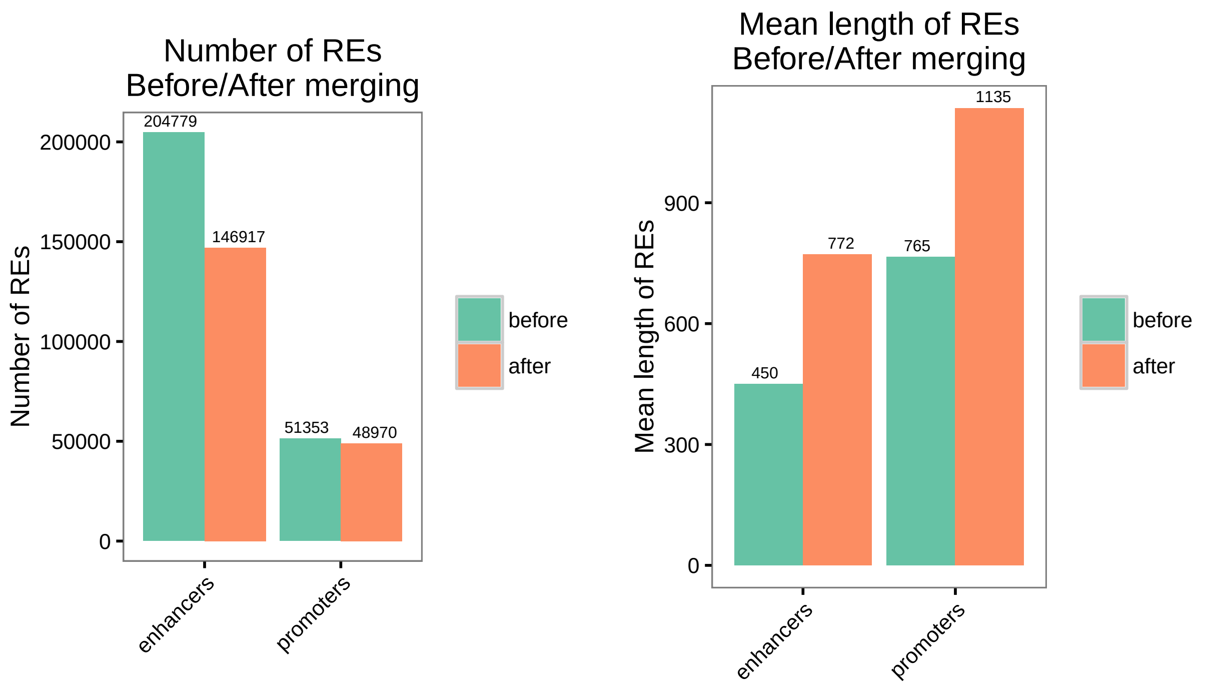
The output of the previous function "getConsensusPeaks" is fed as input:

```r
consensusPeaksAnnotated <- combineAnnotatePeaks(conspeaks = consensusPeaks,
                                                TSS = TSSannot,
                                                merge = TRUE,
                                                regionspecific = TRUE,
                                                mergedistenh = 1500,
                                                mergedistprom = 1000)
```

The function produces a GRanges object containing the annotated, merged peaks – all peaks present in cancerous or normal lung.

Additionally, the function *plotCombineAnnotatePeaks* creates a bar graph to showcase the changes in the number and size of enhancers and promoters before and after merging close-by REs.

```r
plotCombineAnnotatePeaks(consensusPeaksAnnotated)
```



The number of REs decreases and the size of the regions increases, as expected.

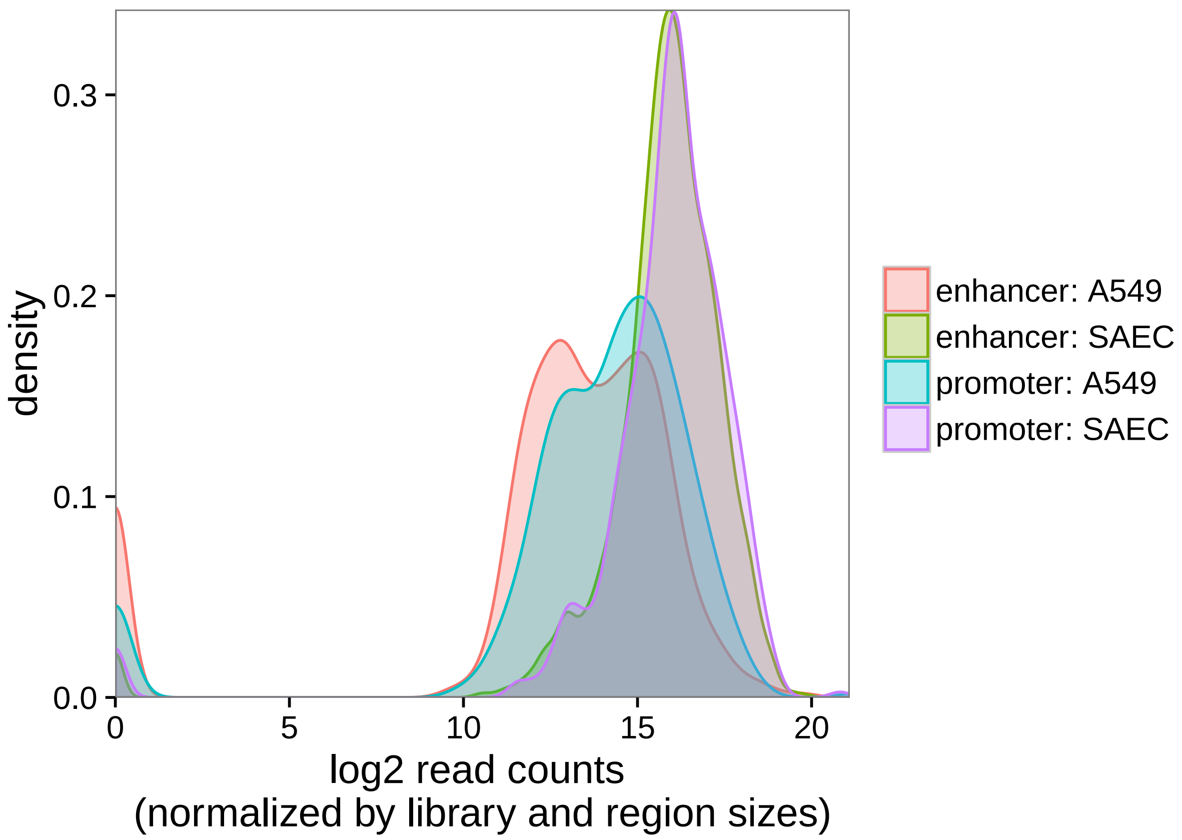
Quantifying RE Accessibility

The next function *getCounts* counts the number of reads in each RE for each sample type – this determines the peak height/intensity and approximates the accessibility of the RE in question.

Read count information across regions of interest is only available in the BAM file format – BAM files must be supplied to this function for each sample type. At least two replicates per sample type are required in order to distinguish reproducible peaks from sample noise. The names of the BAM files are not supplied to the function directly. Instead, the information is included in the CSV file containing sample meta-information. We have already extracted that information from the CSV file in the sampleinfo object.

One cell type must be designated as the "reference type", to which the other cell types will be compared. In this example, the reference type is the normal lung cell line (SAEC).

```
consensusPeaksCounts <- getCounts(annotpeaks = consensusPeaksAnnotated,
                              sampleinfo = csvfile,
                              reference = 'SAEC',
                              chrom = 'chr21')
plotgetcounts(consensusPeaksCounts)
```



Identification of cell-type specific REs

The count information from the function *getCounts* is then processed by the function *countanalysis* – this is the point in the pathway where the changes in peak height across cell types are quantified with an algorithm from the *DESeq2* R package.

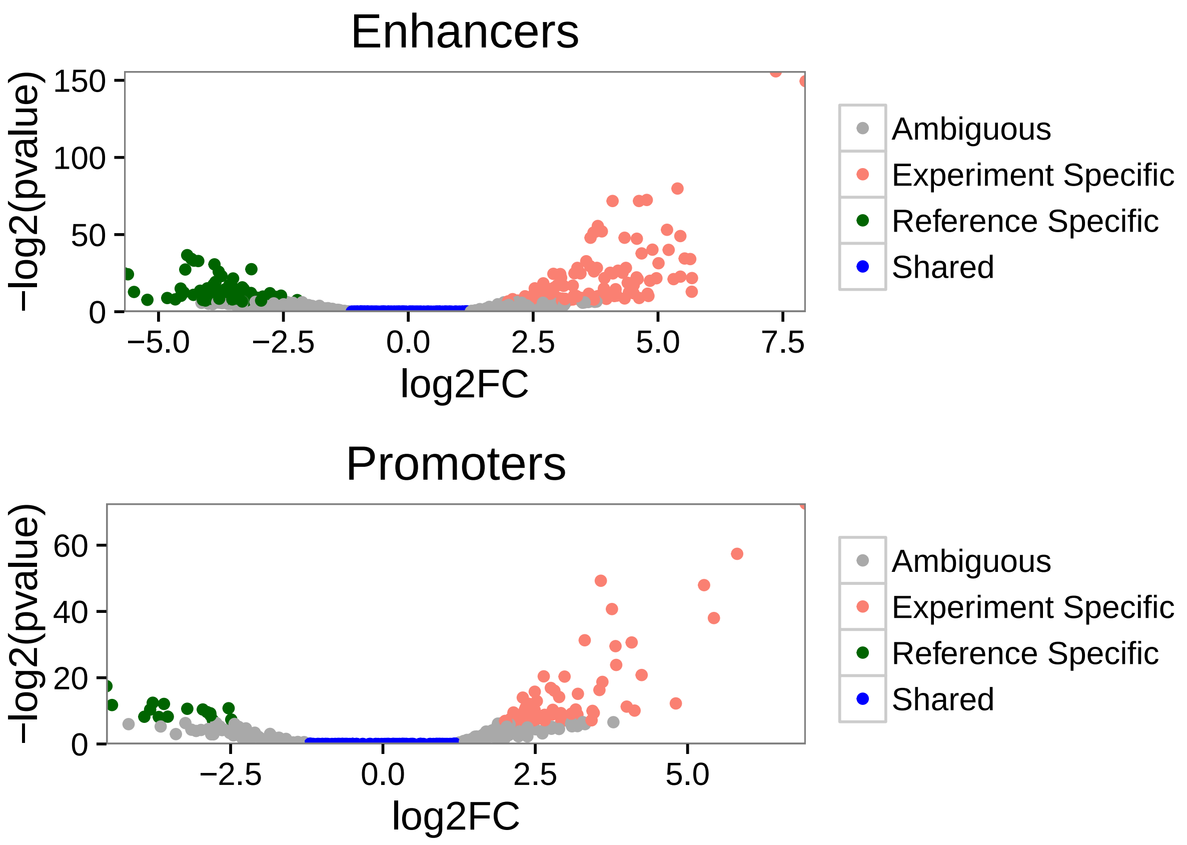
```
alteredPeaks <- countanalysis(counts = consensusPeaksCounts,
                             pval = 0.01,
                             lfcvalue = 1)
```

Finally, altered enhancers and promoters are identified. Larger log2fold changes and smaller p-values signify greater differences between cell types. How large a change is considered cell-type specific is left up to the user.

```r
alteredPeaksCategorized <- categAltrePeaks(alteredPeaks,
                                     lfctypespecific = 1.5,
                                     lfcshared = 1.2,
                                     pvaltypespecific = 0.01,
                                     pvalshared = 0.05)
plotCountAnalysis(alteredPeaksCategorized)
```

```
## Warning: Removed 544 rows containing missing values (geom_point).

## Warning: Removed 123 rows containing missing values (geom_point).
```



REs with large, positive log2fold changes and small p-values are more open in the lung cancer cell type, and therefore, their activity is associated with the cancer phenotype. REs with large, negative log2fold changes and small p-values are more open in the normal lung cell type, and may therefore be associated with tumor suppressive properties. REs with minimal log2fold change and high p-values are equally open in both cell types, and therefore, their activity is more likely to contribute to the house-keeping functions required in every cell.

Pathway enrichment

Finally, the results of the function *categAltrePeak* can be used to identify whether there is an enrichment of certain pathways in the type-specific or shared regions using the "enrichment" function. This is accomplished by linking each RE to the closest gene, and then linking each gene's product to the pathway the gene product participates in. Pathways that recur many times in the gene cluster ("enriched" pathways) are likely to be regulated by the REs linked to the gene cluster (whether type-specific or shared).

Pathways that are unlikely to be of interest can be filtered out of the enrichment results. ALTRE has two metrics to identify "low information" pathways.

1. Vague pathways such as "DNA binding" are removed – they encompass a number of other, more precise pathways (termed "offspring" pathways) already present in the enrichment results. The "offspring" argument limits how many offspring a pathway can contain. Generally, the more offspring, the vaguer and less informative the pathway.
2. Removal or pathways with low gene counts – these pathways are more likely to be false positives. This is because they are more likely to be elevated in the pathway rankings due to the nature of the statistical test – it is much easier to contain half of a pathway's genes in the gene cluster under analysis when the pathway contains only four genes, and not 100 genes. The "gene" argument limits how few genes a pathway can contain.

Pathways fall into three categories (Molecular Function (MF), Biological Process (BP), and Cellular Component (CC)), and each category must be analyzed separately. The category is selected by supplying an argument to "ontoltype". No pathways will have low p-values ($p<0.05$) since we are using only a chr21 subset of the data in this vignette – the p-value argument is set very high so that the results do not come up as "none" in this example. During a real analysis the p-value should be set to a more scientifically meaningful p-value cut-off, such as 0.05. The enriched pathways for this entire dataset at a low p-value cut-off can be seen in the published paper.

```
MFenrich <- pathenrich(analysisresults = alteredPeaksCategorized,
                       ontoltype = 'MF',
                       enrichpvalfilt = 0.99)
```

```
## [1] "Number of rows 9"
```

Post-processing summarization and visualization

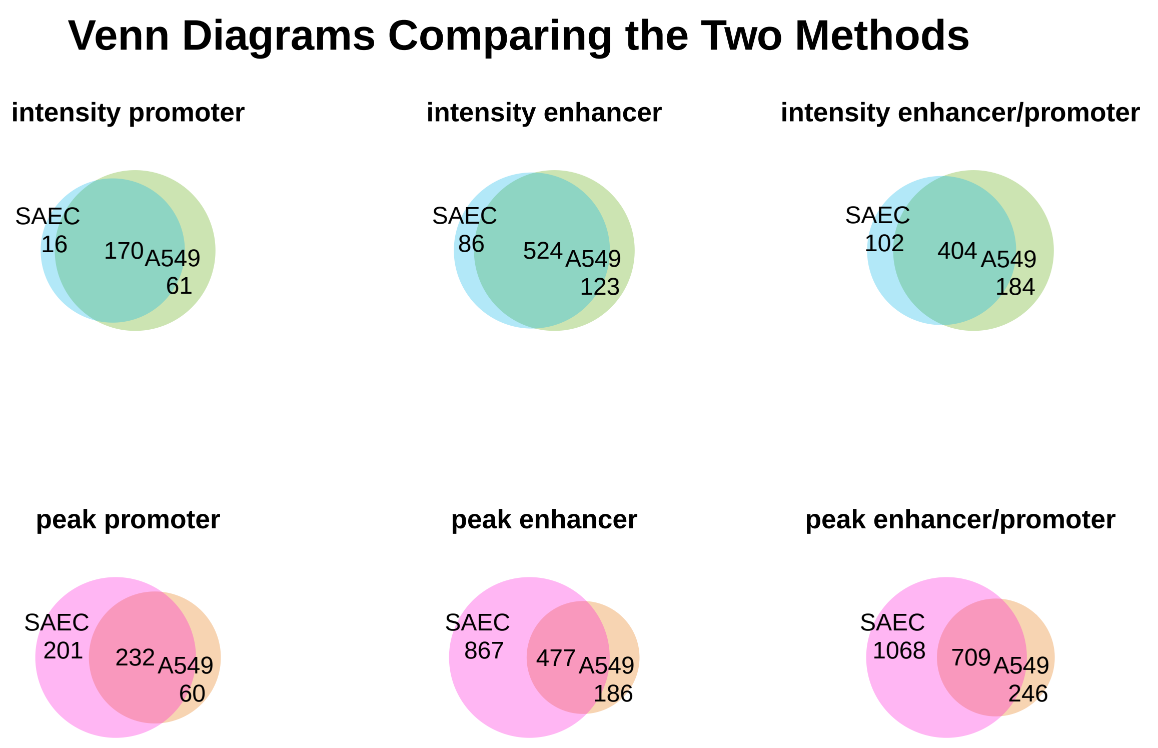
Once results have been obtained, there are a number of additional functions in ALTRE to summarize and better depict the results and raw data in tables and graphs. In this section of the vignette we will walk through these functions.

The function *comparePeaksAltre* creates a table to compare the two methods of identifying altered REs – one based on peak intensity, the other on peak presence or absence as determined by peak calling algorithms. The intensity-based method identifies much fewer type-specific regions.

```
analysisresults <- comparePeaksAltre(alteredPeaksCategorized, reference = "SAEC")
```

The function *plotallvenn* takes the results from *comparePeaksAltre* and translates the table into Venn diagrams.

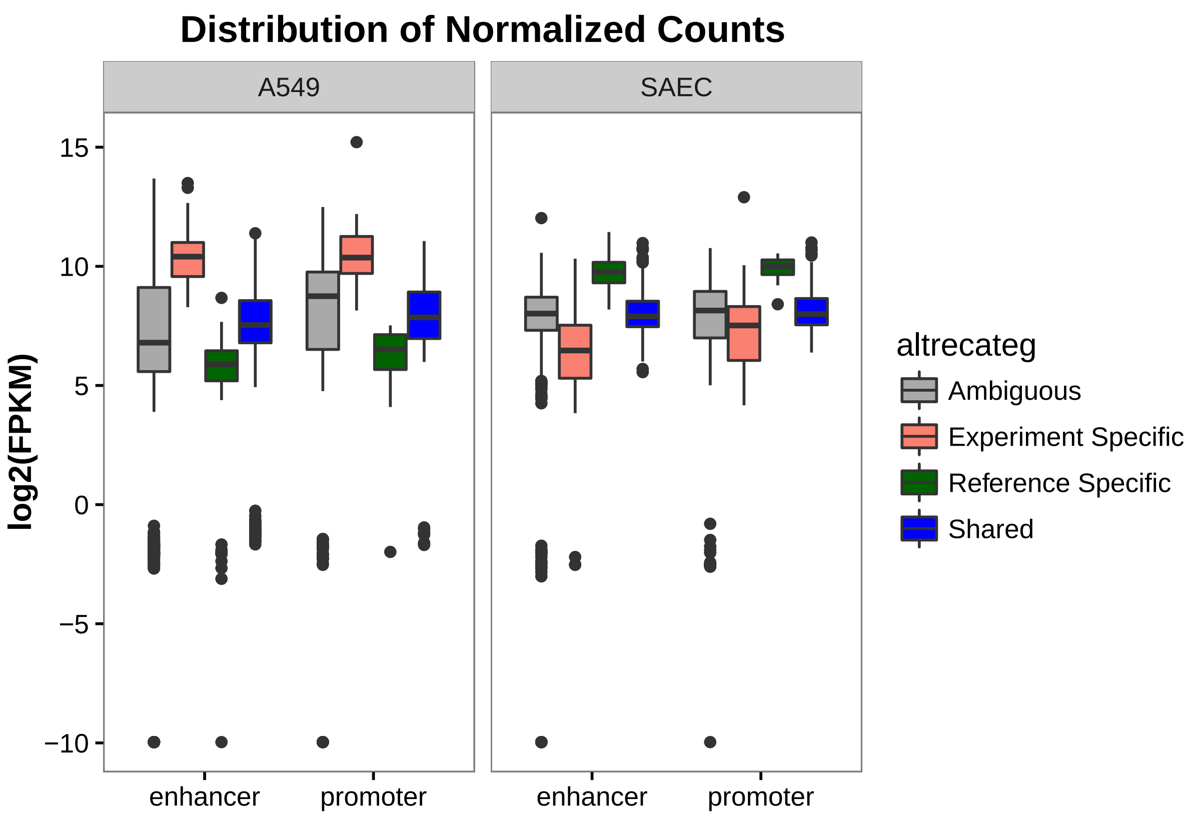
```
plotallvenn(analysisresults)
```



The function *plotDistCountAnalysis* enables users to view the raw count data in regions identified as type-specific or shared. The log2 transformation of reads per kilo-base of RE per million is plotted. As expected, for REs identified as higher log-fold change than normal, the lung cancer cell line has much higher counts than he normal lung cell line. The opposite is true for the lower log-fold change regions. For RE with little log-fold change between cell types, the counts are similar. This visual depiction confirms the results of *countanalysis* and enables users to change parameters if needed.

```
plotDistCountAnalysis(alteredPeaksCategorized, consensusPeaksCounts)
```

```
## Using PEcateg, altrecateg as id variables
```



The function *writeBedFile* allows raw data to be visualized as tracks in the UCSC by creating hotspot files color-coded by type-specificity. A bed file will be created in the working directory, there is no output to the function that can be saved as an object.

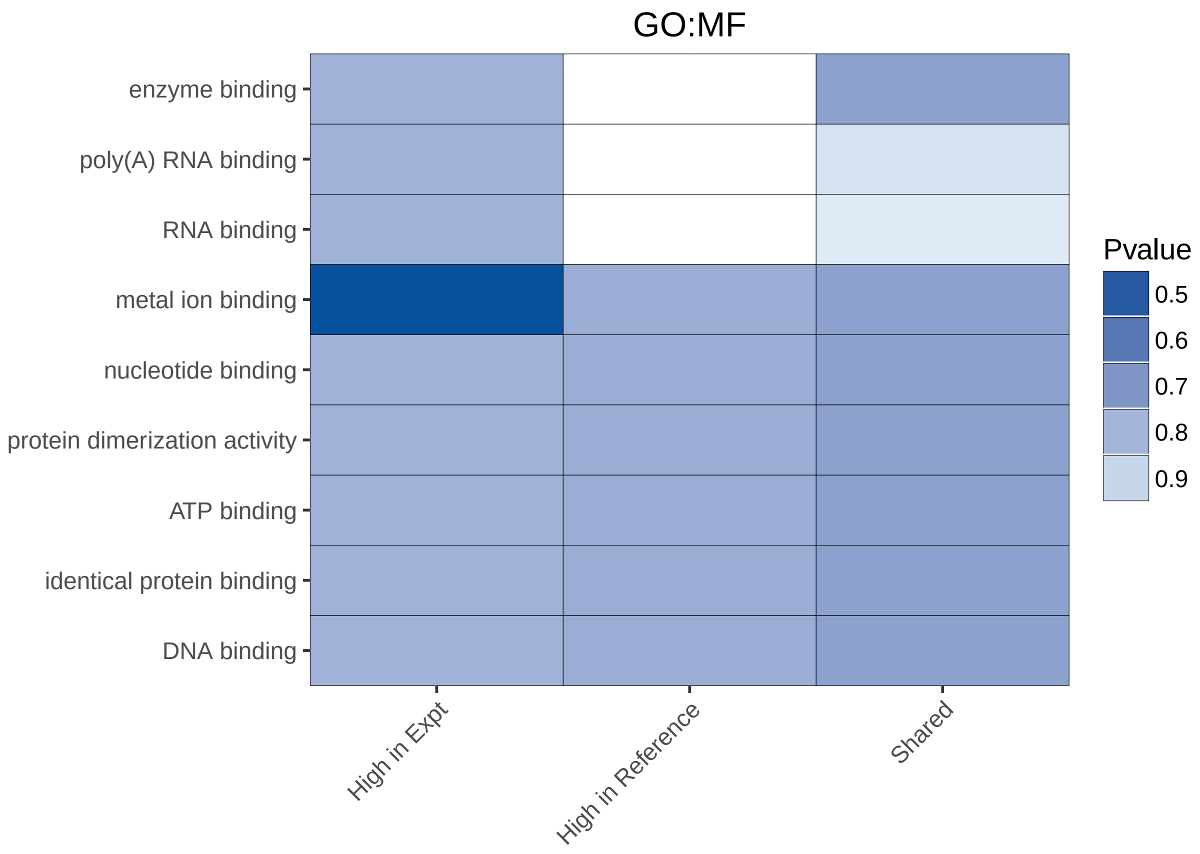
```r
writeBedFile(alteredPeaksCategorized, "output.txt")
```

The function *enrichHeatmap* creates a heatmap plot of the analysis for all three RE types (SAEC-specific, A549-specific, and shared), with the color-coding corresponding to the significance of the pathway – the lighter the blue, the lower the p-value. No pathways will have low p-values (p<0.05) since we are using only a chr21 subset of the data in this vignette – the p-value argument is set very so that the results do not come up as "none" in this example. During a real analysis the p-value should be set to a more scientifically meaningful p-value cut-off, such as 0.05. The enriched pathways for this entire dataset at a low p-value cut-off can be seen in the published paper.

```r
enrichHeatmap(MFenrich, title = "GO:MF", pvalfilt = 0.99)
```

```
## [1] "Pathways enzyme binding"
## [2] "Pathways DNA binding"
## [3] "Pathways identical protein binding"
## [4] "Pathways ATP binding"
## [5] "Pathways protein dimerization activity"
## [6] "Pathways RNA binding"
## [7] "Pathways poly(A) RNA binding"
## [8] "Pathways nucleotide binding"
## [9] "Pathways metal ion binding"
## [1] "Dim heatmapmatrix 9" "Dim heatmapmatrix 3"
## [1] "enzyme binding"
## [1] "DNA binding"
## [1] "identical protein binding"
## [1] "ATP binding"
## [1] "protein dimerization activity"
## [1] "RNA binding"
## [1] "poly(A) RNA binding"
## [1] "nucleotide binding"
## [1] "metal ion binding"
```

```
## Using id as id variables
```



Pathways that are identified as enriched in all three RE types can be filtered out at this step.