An Introduction to *exomePeak*

Jia Meng, PhD

Modified: 18 August, 2013. Compiled: April 12, 2014

1 Introduction

The *exomePeak* R-package has been developed based on the MATLAB "exomePeak" package, for the analysis of RNA epitranscriptome sequencing data with affinity-based shotgun sequencing approach, such as MeRIP-Seq or m6A-Seq. **The exomePeak package is under active development, please don't hesitate to contact me @ jia.meng@hotmail if you have any questions.** The inputs of the main function "exomepeak" are the IP BAM files and input control BAM files:

- From one experiment condition: for peak calling to identify the RNA methylation sites
- From two experimental conditions: for peak calling and differential analysis to unveil the post-transcriptional regulation of RNA modifications.

Gene annotation can be provided as a GTF file, a transcriptDb object, or automatically downloaded from UCSC through the internet. Let us firstly load the package and get the toy data (came with the package) ready.

```
> library("exomePeak")
> gtf=system.file("extdata", "example.gtf", package="exomePeak")
> f1=system.file("extdata", "IP1.bam", package="exomePeak")
> f2=system.file("extdata", "IP2.bam", package="exomePeak")
> f3=system.file("extdata", "IP3.bam", package="exomePeak")
> f4=system.file("extdata", "IP4.bam", package="exomePeak")
> f5=system.file("extdata", "Input1.bam", package="exomePeak")
> f6=system.file("extdata", "Input2.bam", package="exomePeak")
> f7=system.file("extdata", "Input3.bam", package="exomePeak")
> f8=system.file("extdata", "treated_IP1.bam", package="exomePeak")
> f9=system.file("extdata", "treated_Input1.bam", package="exomePeak")
>
```

We will in the next see how the two main functions can be accomplished in a single command.

The first main function of "exomePeak" R-package is to call peaks (enriched binding sites) to detect RNA methylation sites on the exome. Inputs are the gene annotation GTF file, IP and Input control samples in BAM format. This function is used when data from only one condition is available.

```
> result = exomepeak(GENE_ANNO_GTF=gtf,
+                    IP_BAM=c(f1,f2,f3,f4),
+                    INPUT_BAM=c(f5,f6,f7))

[1] "Divide transcriptome into chr-gene-batch sections ..."
[1] "Get Reads Count ..."
[1] "This step may take a few hours ..."
[1] "100 %"
[1] "Get all the peaks ..."
[1] "Get the consistent peaks ..."
[1] "---------------------------------"
[1] "The bam files used:"
[1] "4 IP replicate(s)"
[1] "3 Input replicate(s)"
[1] "---------------------------------"
[1] "Peak calling result: "
[1] "13 peaks detected on merged data."
[1] "Please check 'peak.bed/xls' under /tmp/RtmpfckdtS/Rbuild2ae96dc65f3e/exomePeak/vignette
[1] "10 consistent peaks detected on every replicates. (Recommended list)"
[1] "Please check 'con_peak.bed/xls' under /tmp/RtmpfckdtS/Rbuild2ae96dc65f3e/exomePeak/vign

> names(result)

[1] "all_peaks" "con_peaks"
```

The results will be saved in the specified output directory, including the identified (consistent) peaks in BED/table format. The BED format can be visualized in genome browser directly and the peaks may span one or multiple introns. The function also returns two GRangesList objects, in which there are called peaks and consistent peaks.

The consistent peaks in the latter appear on all the IP replicates compared with the merged Input control sample, and is thus recommended. The log p-value, log fdr and fold enrichment of the identified peaks are stored as metadata, which can be extracted with command *mcols*.

```
> recommended_peaks = result$con_peaks # consistent peaks (Recommended!)
> peaks_info = mcols(recommended_peaks) # information of the consistent peaks
> head(peaks_info)

DataFrame with 6 rows and 3 columns
       lg.p    lg.fdr fold_enrchment
  <numeric> <numeric>      <numeric>
```

```
1     -47.8     -46.5           8.05
2     -15.1     -14.0           9.55
3     -15.0     -13.9           3.78
4    -221.0    -219.0          15.50
5     -14.6     -13.6           5.81
6    -163.0    -161.0          17.50
```

or to get all the peak detected (some of them do not consistently appear on all replicates.):

```
> all_peaks = result$all_peaks # get all peaks
> peaks_info = mcols(all_peaks) # information of all peaks
> head(peaks_info)
```

```
DataFrame with 6 rows and 3 columns
       lg.p    lg.fdr fold_enrchment
  <numeric> <numeric>      <numeric>
1      -6.9     -6.04           2.66
2     -47.8    -46.50           9.55
3     -15.0    -13.90           3.78
4    -221.0   -219.00          15.50
5     -14.6    -13.60           5.81
6    -163.0   -161.00          17.50
```

When there are MeRIP-Seq data available from two experimental conditions, the “exomepeak” function may can unveil the dynamics in post-transcriptional regulation of the RNA methylome. In the following example, the function will report the sites that are post-transcriptional differentially methylated between the two tested conditions (TREATED vs. UNTREATED).

```
> result = exomepeak(GENE_ANNO_GTF=gtf,
+                    IP_BAM=c(f1,f2,f3,f4),
+                    INPUT_BAM=c(f5,f6,f7),
+                    TREATED_IP_BAM=c(f8),
+                    TREATED_INPUT_BAM=c(f9))
```

```
[1] "Divide transcriptome into chr-gene-batch sections ..."
[1] "Get Reads Count ..."
[1] "This step may take a few hours ..."
[1] "100 %"
[1] "Comparing two conditions ..."
[1] "Get all the peaks ..."
[1] "Get the consistent peaks ..."
[1] "---------------------------------"
[1] "The bam files used:"
[1] "4 IP replicate(s)"
[1] "3 Input replicate(s)"
```

```
[1] "1 TREATED IP replicate(s)"
[1] "1 TREATED Input replicate(s)"
[1] "---------------------------------"
[1] "Peak calling and differential analysis result: "
[1] "13 peaks detected."
[1] "Please check 'diff_peak.bed/xls' under /tmp/RtmpfckdtS/Rbuild2ae96dc65f3e/exomePeak/vig
[1] "---------------------------------"
[1] "0 significantly differential methylated peaks are detected."
[1] "Please check 'sig_diff_peak.bed/xls' under /tmp/RtmpfckdtS/Rbuild2ae96dc65f3e/exomePeak
[1] "---------------------------------"
[1] "0 consistent significantly differential methylated peaks are detected.(Recommended list
[1] "Please check 'con_sig_diff_peak.bed/xls' under /tmp/RtmpfckdtS/Rbuild2ae96dc65f3e/exome
[1] "---------------------------------"
```

The algorithm will firstly identify reads enriched binding sites or peaks, and then check whether the sites are differentially methylated between the two experimental conditions. The results will be saved in the specified output directory, including the identified (consistent) peaks in BED and table formats, along with the differential information indicating whether the site is hyper- or hypo-methylated under the treated condition. Similar to the peak calling case, the BED format can be visualized in genome browser directly and the peaks may span one or multiple introns.

Similar to the peak calling case, the function will report a set of consistent differentially methylated peaks saved in the specified folder, which is the recommended set. The function also returns 3 GRangesList object, containing all the peaks, the differentially methylated peaks with the given threshold on the merged data, consistently differentially methylated peaks. The consistent differentially methylated peaks in the last appear to be differential for all the replicates and is thus recommended. The information of the identified peaks and the differential analysis are stored as metadata, which can be extracted.

```
> names(result)
```

```
[1] "diff_peaks"         "sig_siff_peaks"
[3] "con_sig_diff_peaks"
```

```
> is.na(result$con_sig_diff_peaks) # no reported consistent differnetial peaks
```

```
[1] TRUE
```

Unfortunately, there is no reported consistent differnetial peaks on the toy data, to get the information of all the peaks and the differential analysis information:

```
> diff_peaks = result$diff_peaks # consistent differential peaks (Recommended!)
> peaks_info = mcols(diff_peaks) # information of the consistent peaks
> head(peaks_info[,1:3]) # peak calling information
```

```
DataFrame with 6 rows and 3 columns
       lg.p    lg.fdr fold_enrchment
  <numeric> <numeric>      <numeric>
1     -5.91     -5.05           2.47
2    -50.70    -49.50           9.19
3    -24.70    -23.50           5.00
4   -222.00   -220.00          14.40
5    -15.40    -14.40           5.97
6   -171.00   -170.00          14.70
```

```
> head(peaks_info[,4:6]) # differential analysis information
```

```
DataFrame with 6 rows and 3 columns
  diff.lg.fdr diff.lg.p diff.log2.fc
    <numeric> <numeric>    <numeric>
1      -0.262    -0.402       -1.180
2      -0.262    -0.296        0.775
3      -0.274    -0.689        1.780
4      -1.280    -2.090       -1.750
5      -0.262    -0.342       -1.050
6      -1.280    -2.200       -1.450
```

Gene annotation may be alternatively downloaded directly from internet, but will take a really long time due to the downloading time and huge transcriptome needed to be scanned.