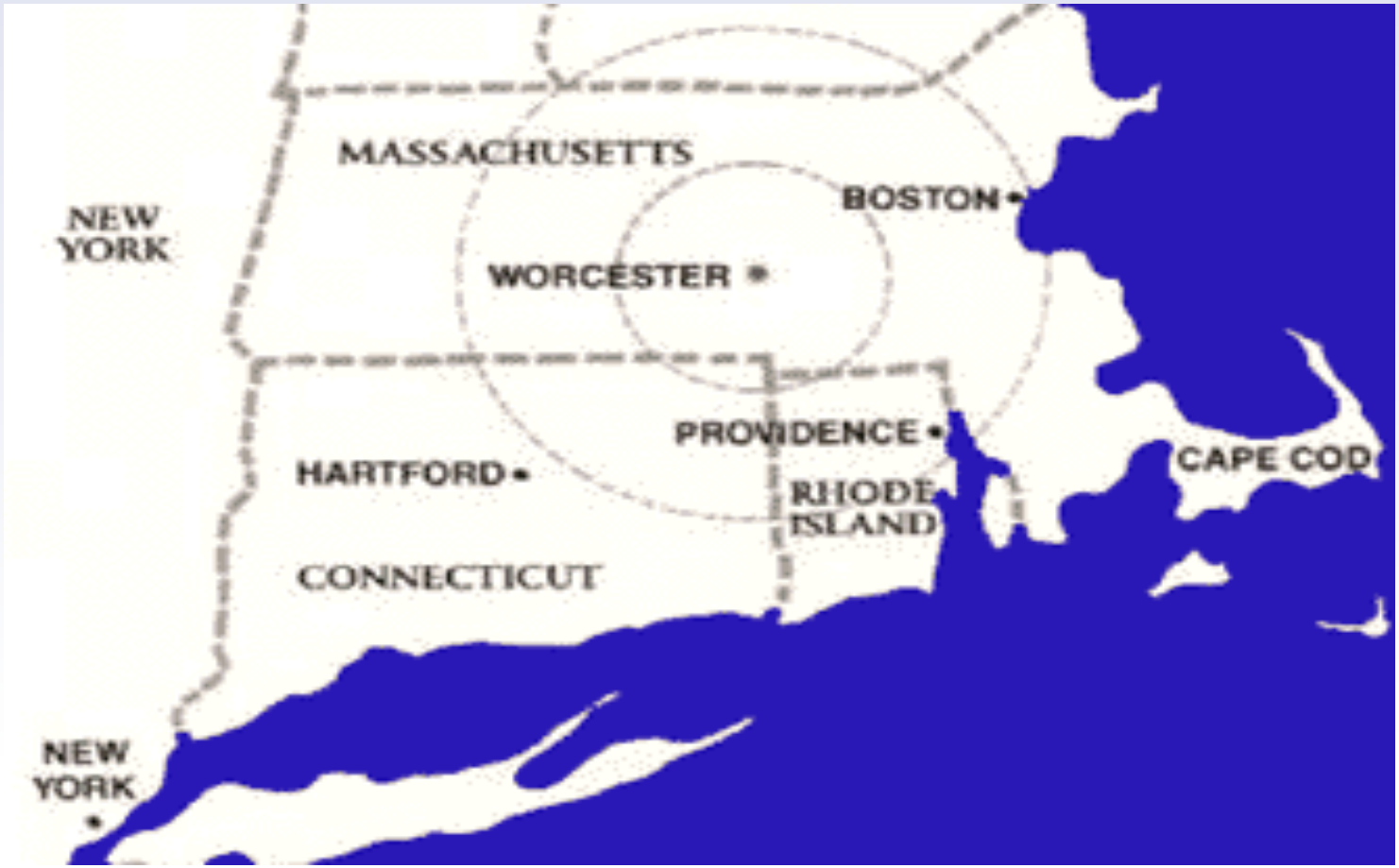
University of Massachusetts Medical School
UMASS

BioC2010: ChIPpeakAnno Practical

July 29th 2010

Lihua Julie Zhu

THE HEART OF MASSACHUSETTS



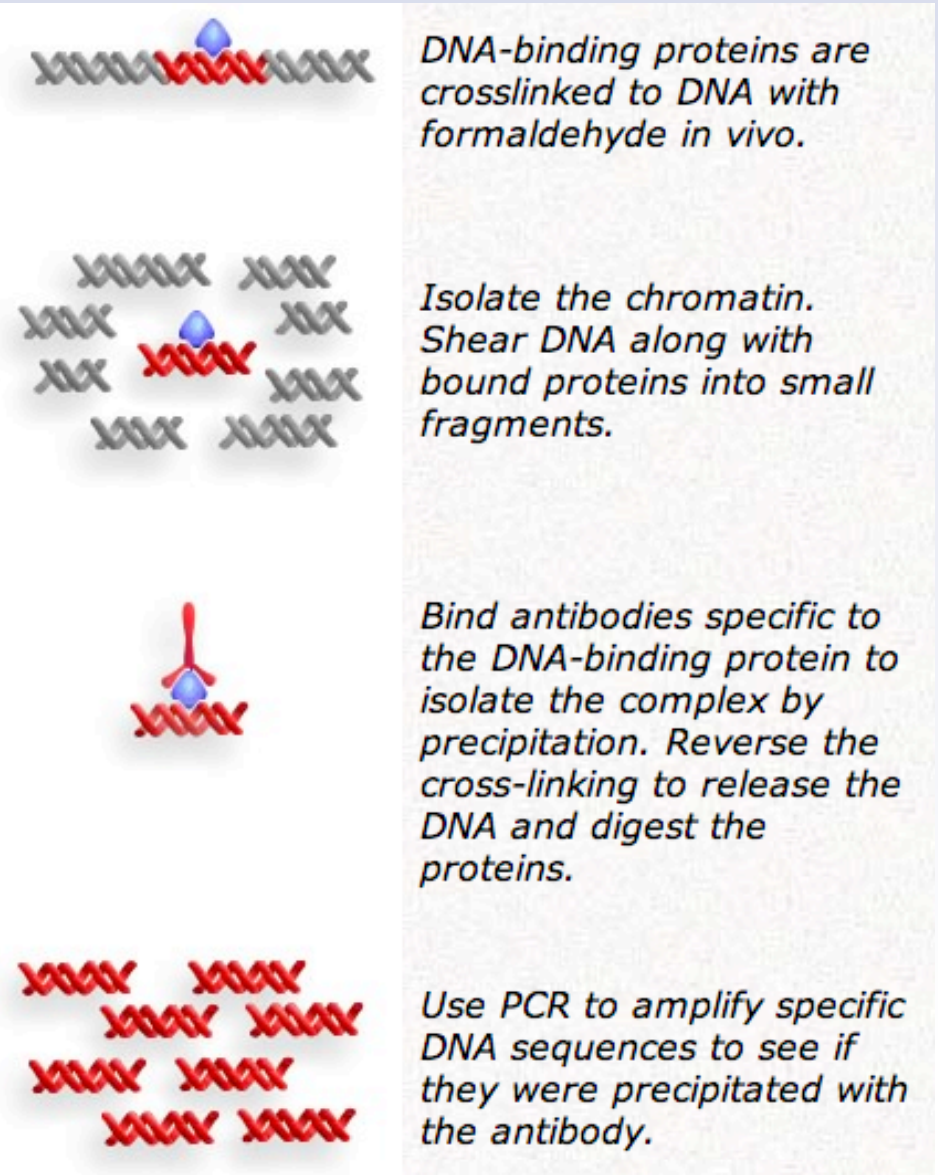
Outline

- Motives
- Functionality
- Dependency
- Installation
- Demo
- Get help and Reference

Protein and DNA Interaction

- Interactions between proteins and DNA are fundamental to life.
- They mediate
  - Transcription
  - DNA replication and recombination
  - DNA repair

- ChIP (Chromatin immunoprecipitation) Procedure For Determining the DNA Binding Sites
- Adopted from http://www.bio.brandeis.edu/haberlab



High-throughput Identification of DNA Binding Sites

- ChIP-seq
  - ChIP followed by high-throughput sequencing
- ChIP-chip
  - ChIP followed by genome tiling array analysis

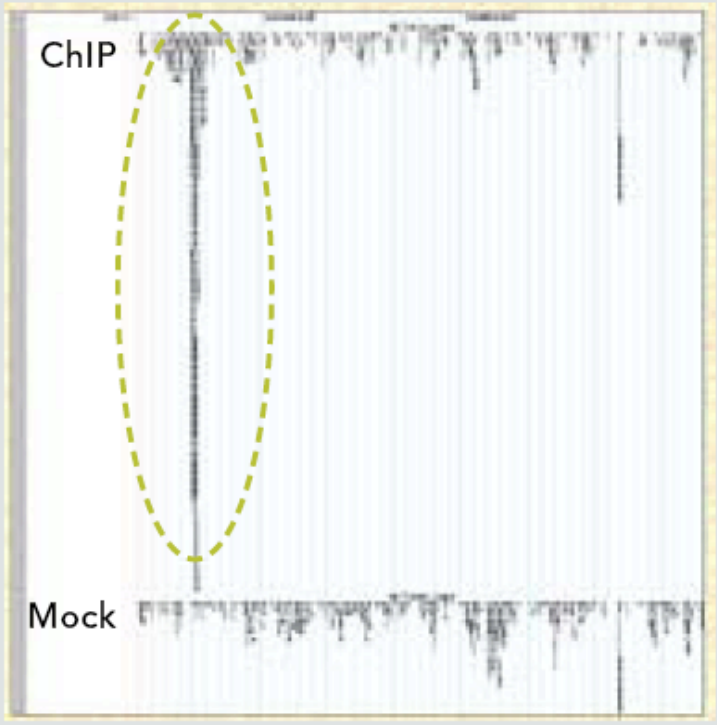
PEAK IDENTIFICATION

- Johnson DS, Mortazavi A, Myers RM, Wold B: **Genome-wide mapping of in vivo protein-DNA interactions**. *Science* 2007, **316**(5830):1497-1502.
- Fejes AP, Robertson G, Bilenky M, Varhol R, Bainbridge M, Jones SJ: **FindPeaks 3.1: a tool for identifying areas of enrichment from massively parallel short-read sequencing technology**. *Bioinformatics* 2008, **24**(15): 1729-1730.
- Ji H, Jiang H, Ma W, Johnson DS, Myers RM, Wong WH: **An integrated software system for analyzing ChIP-chip and ChIP-seq data**. *Nat Biotechnol* 2008, **26**(11):1293-1300.
- Zhang Y, Liu T, Meyer CA, Eeckhoute J, Johnson DS, Bernstein BE, Nussbaum C, Myers RM, Brown M, Li W *et al*: **Model-based analysis of ChIP-Seq (MACS)**. *Genome Biol* 2008, **9**(9):R137.
- Zhang ZD, Rozowsky J, Snyder M, Chang J, Gerstein M: **Modeling ChIP sequencing in silico with applications**. *PLoS Comput Biol* 2008, **4**(8):e1000158.
- Sharon E, Lubliner S, Segal E: **A feature-based approach to modeling protein-DNA interactions**. *PLoS Comput Biol* 2008, **4**(8):e1000154.
- Rozowsky J, Euskirchen G, Auerbach RK, Zhang ZD, Gibson T, Bjornson R, Carriero N, Snyder M, Gerstein MB: **PeakSeq enables systematic scoring of ChIP-seq experiments relative to controls**. *Nat Biotechnol* 2009, **27**(1): 66-75.

OUTPUT OF PEAK CALLING ALGORITHMS

- A file with at least the chromosome coordinate information if not in standard BED or GFF file format.
  - MACS (http://liulab.dfci.harvard.edu/MACS/) outputs BED file.
  - ChipSeq Peak Finder (http://woldlab.caltech.edu/html/chipseq_peak_finder) outputs BED file.
  - CisGenome (http://www.biostat.jhsph.edu/~hji/cisgenome/) for analyzing both ChIP-chip and ChIP-seq outputscod file format which is a text file format, e.g., peak1 chr1 1000 2000 +
  - SISSRS for analyzing ChIP-seq dataset (http://www.rajajothi.com/sissrs/) outputs a text file with chr binding-site-start-position binding-site-end-position NumTags [Fold] [p-value]
  - NimbleScan software for analyzing ChIP-chip dataset output gff file.
  - QuEST (http://mendel.stanford.edu/sidowlab/downloads/quest/) outputs a text file with the first three columns containing name of the peak, chromosome and start-end, e.g., R-2 chr11 47556767-47557549)

FIGURE 1: CHIP-SEQ TAGS IN BED FORMAT, DISPLAYED IN UCSC BROWSER



Example of custom tracks submitted in BED format (upper track is from ChIP sample and lower track is from mock control sample). The peak on the left in the ChIP sample (green circle) is significant. However, the peak on the right side is detected in both the ChIP and mock samples and is not significant.

Adopted from http://www.illumina.com

ChIPpeakAnno

- Batch annotate enriched peaks
  - ChIP-seq
  - ChIP-chip
  - cap analysis of gene expression (CAGE)
  - any experiments resulting in a large number of enriched genomic regions

FUNCTIONALITY

- Find the nearest genes for each set of peaks.
- Find all genes within a certain distance from the peaks.
- Identify enriched Gene Ontology (GO) terms associated with adjacent genes of the peaks.
- Label peaks with any annotation of interest
  - a dataset from the literature
  - CpG island
  - conserved element
- Determine the significance of overlap and drawing Venn diagrams to visualize the extent of the overlap
  - binding sites among replicates
  - binding sites among transcription factors within a complex
  - binding sites among different experiments such as yours and the ones in literature
- Retrieve genomic sequences flanking putative binding sites
  - for motif discovery
  - for cloning
  - for PCR amplification

DEPENDENCY

- IRanges
  - Provides infrastructure for representing and manipulating sets of integer ranges, and implements algorithms for range-based calculations, matching and searching
- BSgenome
  - Supplies infrastructure for efficiently representing, accessing and analyzing whole genome
- Biostrings
  - Implements functions for pattern matching, sequence alignment and string manipulation
- GO.db
  - A set of annotation maps describing the entire Gene Ontology assembled using data from GO
- biomaR
  - Provides an R interface to a collection of databases implementing the BioMart software suite
- Multtest
  - Non-parametric bootstrap and permutation re-sampling-based multiple testing procedures
- Limma
  - Data analysis, linear models and differential expression for microarray data

INSTALLATION

Install R-2.11.1

Windows: http://cran.fhcrc.org/bin/windows/base/

OS X: http://cran.fhcrc.org/bin/macosx/

Source (Linux): http://cran.fhcrc.org/sources.html

INSTALLATION

All the dependent packages can be installed from R as:

```
source("http://bioconductor.org/biocLite.R")
biocLite(c("IRanges", "Biostring",
           "BSgenome", "biomaRt", "GO.db", "multtest",
           "limma"))
```

ChIPpeakAnno can be installed from R as:

```
source("http://bioconductor.org/biocLite.R")
biocLite("ChIPpeakAnno")
```

The lightweight organism-specific package BSgenome.Ecoli.NCBI.20080805 and org.Hs.eg.db were installed during build time for testing the code snippets in the vignette.

MAIN FUNCTIONS

annotatePeakInBatch

- Find the nearest genes for each set of peaks
- Find all genes within a certain distance from the peaks
- Label peaks with any annotation of interest
  - a dataset from the literature
  - CpG island
  - conserved element

```
annotatePeakInBatch(myPeakList, mart, featureType = c("TSS",
    "miRNA","Exon"), AnnotationData, output=c("nearestStart",
    "overlapping", "both"), multiple=c(FALSE,TRUE), maxgap=0)
```

PARAMETERS

- myPeakList
  - Peaks as RangedData
- mart
  - used if AnnotationData not supplied, a mart object, see useMart of bioMaRt package for details
- featureType
  - used if AnnotationData not supplied, TSS, miRNA or exon
- AnnotationData
  - annotation data obtained from getAnnotation or customized annotation of class RangedData. If not supplied, then annotation will be obtained from biomaRt automatically
- output
  - nearestStart: will output the nearest features calculated as peak start - feature start (feature end if feature resides at minus strand);
  - overlapping: will output overlapping features with maximum gap =maxgap between peak range and feature range
  - both: will output all the nearest features, in addition, will output any features that overlap the peak that is not the nearest features.
- Multiple: not applicable when output is nearestStart. TRUE: output multiple overlapping features for each peak. FALSE: output at most one overlapping feature for each peak
- Maxgap: Non-negative integer. Intervals with a separation of maxgap or less are considered to be overlapping

OUTPUT

- RangedData
  - start: the start position of the peak
  - end : the end position of the peak
  - space: the chromosome location where the peak is located
  - Feature: id of the feature such as ensembl gene ID
  - start_position: start position of the feature such as gene
  - end_position: end position of the feature such as the gene
  - shortestDistance: the shortest distance from either end of peak to either end the feature.
  - fromOverlappingOrNearest
    - NearestStart: indicates this feature's start (feature's end for features at minus strand) is closest to the peak start
    - Overlapping: indicates this feature overlaps with this peak although it is not the nearest feature start
  - strand
    - 1 or + for positive strand and -1 or - for negative strand where the feature is located
  - insideFeature:
    - upstream: peak resides upstream of the feature;
    - downstream: peak resides downstream of the feature;
    - inside: peak resides inside the feature;
    - overlapStart: peak overlaps with the start of the feature;
    - overlapEnd: peak overlaps with the end of the feature;
    - includeFeature: peak include the feature entirely
  - distancetoFeature
    - distance to the nearest feature such as transcription start site. The distance is calculated as the distance between the start of the binding site and the TSS that is the gene start for genes located on the forward strand and the gene end for genes located on the reverse strand

DEMO

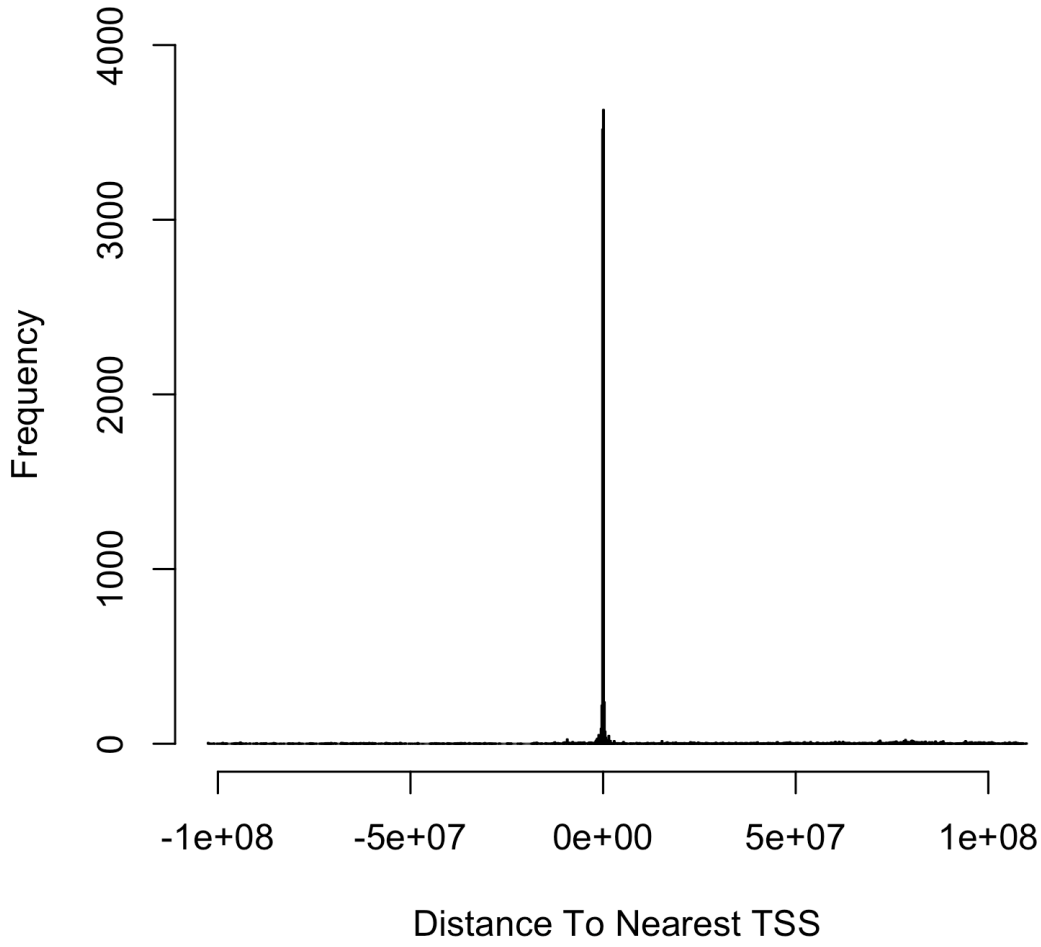
EXAMPLE 1: FINDING THE NEAREST GENE AND THE DISTANCE TO THE TRANSCRIPTION START SITE OF THE NEAREST GENE.

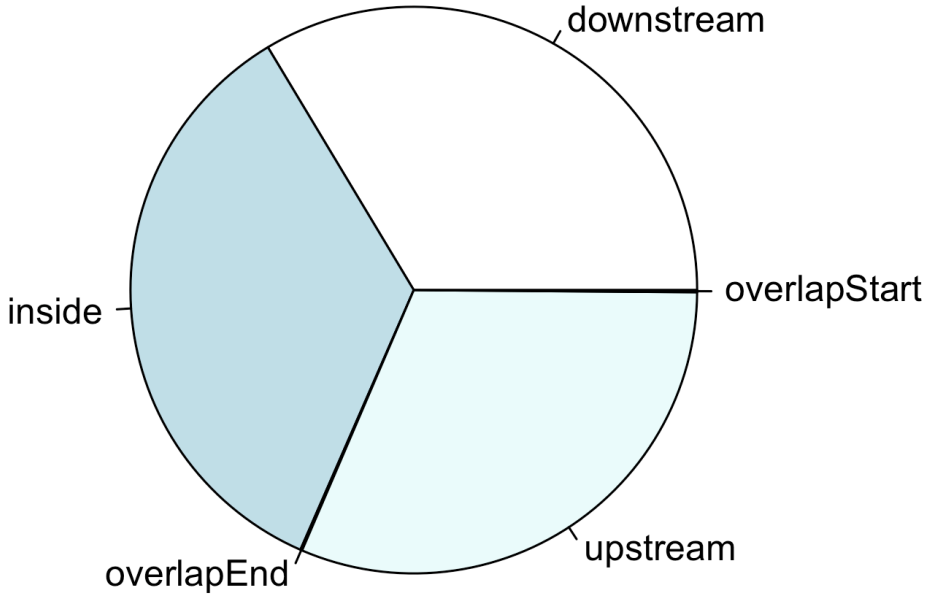
```
library(ChIPpeakAnno)
data(myPeakList)
data(TSS.human.GRCh37)
annotatedPeak = annotatePeakInBatch (myPeakList[1:6,],
    AnnotationData = TSS.human.GRCh37)
#The annotated peaks can be saved as an Excel file for biologists to
    view easily.
write.table(as.data.frame(annotatedPeak),
    file="annotatedPeakList.xls", sep="\t", row.names=FALSE)
```

Plot the distribution of the peaks relative to the TSS

Gives a birds-eye view of the peak distribution relative to the genomic features of interest.

```
data(annotatedPeak)
y = annotatedPeak$distancetoFeature[!is.na(annotatedPeak$distancetoFeature) & annotatedPeak$fromOverlappingOrNearest == "NearestStart"]
hist(y, xlab="Distance To Nearest TSS", main="", breaks=1000, xlim=c(min(y)-100, max(y)+100)) temp = as.data.frame(annotatedPeak)
pie(table(temp[as.character(temp$fromOverlappingOrNearest) == "Overlapping" | (as.character(temp$fromOverlappingOrNearest) == "NearestStart" & !temp$peak %in%  temp[as.character(temp$fromOverlappingOrNearest) == "Overlapping", ]$peak) ,]$insideFeature))
```


Frequency
0
1000
2000
3000
4000
-1e+08
-5e+07
0e+00
5e+07
1e+08
Distance To Nearest TSS


downstream
overlapStart
upstream
overlapEnd
inside

Obtain annotation on-line using *getAnnotation*

```
mart = useMart(biomart="ensembl",
    dataset="hsapiens_gene_ensembl")
#Annotation = getAnnotation(mart,  featureType="TSS")
Annotation = getAnnotation(mart,  featureType="miRNA")
as.data.frame(Annotation)[1:10,]
```

To obtain annotation with other genomic features, it is necessary to change the *featureType* (e.g., "exon" for exon, "miRNA" for miRNA, "5utr" for 5' utr, "3utr" for 3' utr, and "ExonPlusUtr" for Exon plus utr).

Example 2: Label the peaks from your experiment with a list of peaks in the literature

```
myexp =  RangedData(IRanges(start=c
(1543200,1557200,1563000,1569800, 167889600,100,1000), end=c
(1555199,1560599,1565199,1573799, 167893599,200,1200),
names=c("p1","p2","p3","p4","p5","p6", "p7")), strand=as.integer
(1),space=c(6,6,6,6,5,4,4))
literature = RangedData(IRanges(start=c
(1549800,1554400,1565000,1569400,167888600,120,800), end=c
(1550599,1560799,1565399,1571199,167888999,140,1400), names=c
("f1","f2","f3","f4","f5","f6","f7")), strand=c(1,1,1,1,1,-1,-1), space=c
(6,6,6,6,5,4,4))
annotatedPeak1= annotatePeakInBatch(myexp, AnnotationData =
literature, output="both", maxgap=1000, multiple=TRUE)
pie(table(as.data.frame(annotatedPeak1)$insideFeature))
as.data.frame(annotatedPeak1)
```

Example 2 – Cont. Different Parameter Setting

```
annotatedPeak1= annotatePeakInBatch(myexp,
AnnotationData = literature, output="overlapping",
maxgap=1000, multiple=TRUE)
as.data.frame(annotatedPeak1)
annotatedPeak1= annotatePeakInBatch(myexp,
AnnotationData = literature, output="nearestStart")
as.data.frame(annotatedPeak1)
```

* New Feature in ChIPpeakAnno version 1.5.4

```
PeakLocForDistance=c("start", "middle", "end")
FeatureLocForDistance=c("TSS", "middle", "start", "end")
```

BED2RangedData and GFF2RangedData

```
test.bed = data.frame(cbind(chrom = c("4", "6"), chromStart=c
("100", "1000"),chromEnd=c("200", "1100"), name=c("peak1",
"peak2")))
test.rangedData = BED2RangedData(test.bed)
as.data.frame(annotatePeakInBatch(test.rangedData,
AnnotationData = literature))
test.GFF = data.frame(cbind(seqname  = c("chr4", "chr4"),
source=rep("Macs", 2), feature=rep("peak", 2), start=c("100",
"1000"), end=c("200", "1100"), score=c(60, 26), strand=c(1, 1),
frame=c(".", 2), group=c("peak1", "peak2")))
test.rangedData = GFF2RangedData(test.GFF)
as.data.frame(annotatePeakInBatch(test.rangedData,
AnnotationData = literature))
```

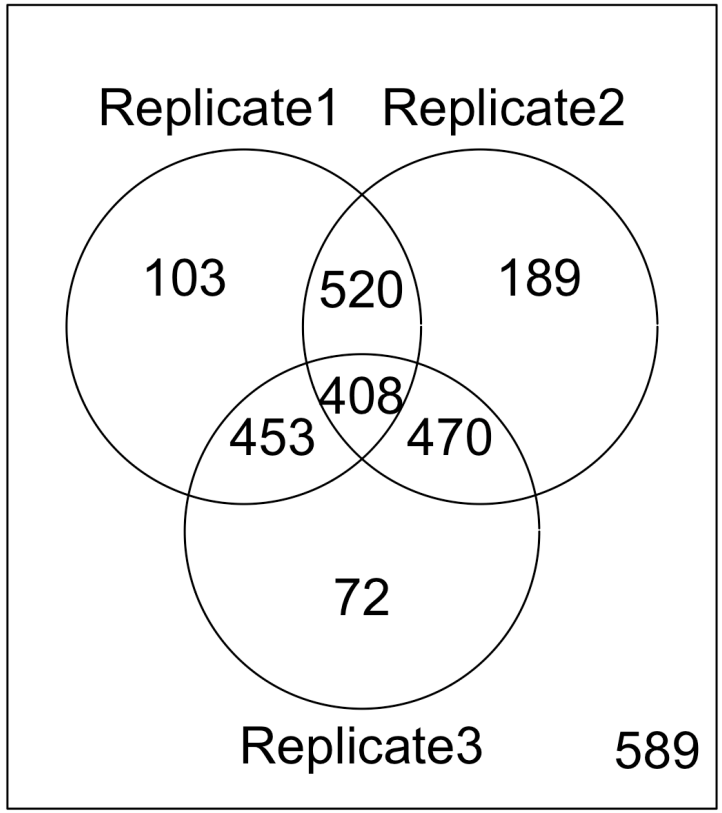
BED2RangedData and GFF2RangedData

```
test.bed = data.frame(cbind(chrom = c("4", "6"), chromStart=c
("100", "1000"),chromEnd=c("200", "1100"), name=c("peak1",
"peak2")))
test.rangedData = BED2RangedData(test.bed)
as.data.frame(annotatePeakInBatch(test.rangedData,
AnnotationData = literature))
test.GFF = data.frame(cbind(seqname  = c("chr4", "chr4"),
source=rep("Macs", 2), feature=rep("peak", 2), start=c("100",
"1000"), end=c("200", "1100"), score=c(60, 26), strand=c(1, 1),
frame=c(".", 2), group=c("peak1", "peak2")))
test.rangedData = GFF2RangedData(test.GFF)
as.data.frame(annotatePeakInBatch(test.rangedData,
AnnotationData = literature))
```

Example 3: Determine the significance of the overlapping and visualize the overlap as a Venn diagram among different datasets.

Overlap Significance Testing and Visualization

```
data(Peaks.Ste12.Replicate1)
data(Peaks.Ste12.Replicate2)
data(Peaks.Ste12.Replicate3)
makeVennDiagram(RangedDataList
  (Peaks.Ste12.Replicate1, Peaks.Ste12.Replicate2,
  Peaks.Ste12.Replicate3), NameOfPeaks = c
  ("Replicate1","Replicate2","Replicate3"), maxgap = 0,
  totalTest = 1580)
```


Replicate1
Replicate2
103
520
189
408
453
470
72
Replicate3
589

Combine the Overlapping Peaks Across Replicates

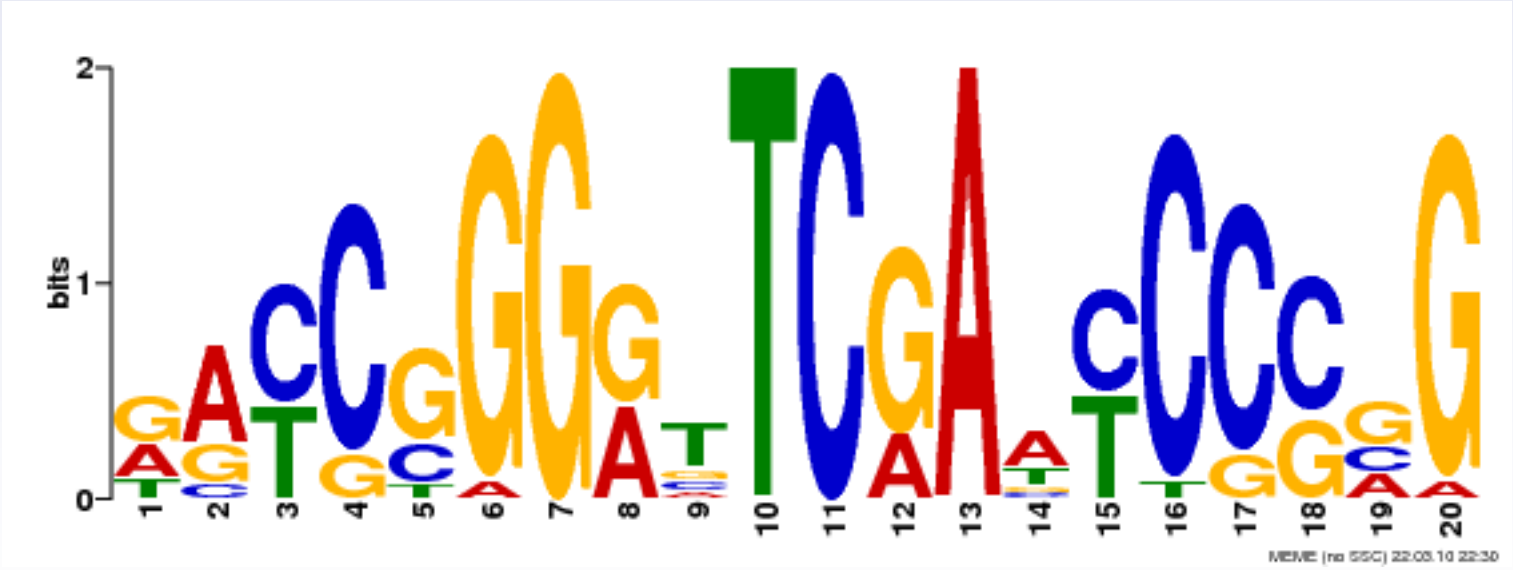
```
MergedPeaks = findOverlappingPeaks
  (findOverlappingPeaks(Peaks.Ste12.Replicate1,
  Peaks.Ste12.Replicate2, maxgap = 0, multiple =
  F, NameOfPeaks1 = "R1", NameOfPeaks2 =
  "R2")$MergedPeaks, Peaks.Ste12.Replicate3,
  maxgap = 0, multiple = F, NameOfPeaks1 =
  "R1R2",  NameOfPeaks2 = "R3")$MergedPeak
```

```
as.data.frame(MergedPeaks)
```

Example 4: Obtain the sequences around the binding sites for PCR amplification or motif discovery

Obtain Genomic DNA sequences

```
peaks = RangedData(IRanges(start = c(100, 500), end = c(300,
 600), names = c("peak1", "peak2")), space = c("NC_008253",
 "NC_010468"))
library(BSgenome.Ecoli.NCBI.20080805)
peaksWithSequences = getAllPeakSequence(peaks, upstream =
    100, downstream = 100, genome = Ecoli)
#To convert the sequences to a common FASTA file format, the
    following function is called.
write2FASTA(peaksWithSequences, file="test.fa", width=50)
available.genomes()
```


bits
2
1
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
MEME (no SSC) 22:05:10 22:30

EXAMPLE 5: OBTAIN ENRICHED GO TERMS NEAR THE PEAKS

Obtain Enriched GO

```
data(annotatedPeak)

library(org.Hs.eg.db)

enrichedGO <- getEnrichedGO (annotatedPeak[1:6,],
    orgAnn="org.Hs.eg.db", maxP=0.1, multiAdj =TRUE,
    minGOterm=1,  multiAdjMethod="BH")
```

Parameters

maxP is the maximum p-value required to be considered to be significant

multiAdj indicates whether to apply multiple hypothesis testing adjustment

minGOterm is the minimum count in a genome for a GO term to be included

multiAdjMethod is the multiple testing procedure to be applied

orgAnn is the organism specific GO annotation (http://www.bioconductor.org/packages/release/data/annotation/ for additional org.xx.eg.db packages)

Additional Parameters

annotatedPeak: RangedData or character vector

feature_id_type: "entrez_id" or "ensembl_gene_id"

Caveat

Not all species specific GO annotation package are Enrez ID centric

- `org.At.tair.db (Arabidopsis) is TAIR ID centric`
- `org.Sc.sgd.db (Yeast) is orf centric`

Work around:

- set annotatedPeak to a character vector of species-specific IDs
- set feature_type_id as entrez_id

```
enrichedGO.Arab <- getEnrichedGO (tarIDs, feature_id_type="entrez_id",
orgAnn="org.At.tair.db", maxP=0.05, multiAdj =TRUE, minGOterm=10,
multiAdjMethod="BH")
enrichedGO.Cse4 <- getEnrichedGO (orfs, feature_id_type="entrez_id",
orgAnn="org.Sc.sgd.db", maxP=0.05, multiAdj =TRUE, minGOterm=5,
multiAdjMethod="BH")
```

New Features and Future Plan

- New Features available in ChIPpeakAnno version 1.5.4
  - annotatePeakInBatch
    - PeakLocForDistance=c("start", "middle", "end")
    - FeatureLocForDistance=c("TSS", "middle", "start", "end")
- Need to add "TSS" option to PeakLocForDistance
- Nee to output a list of genes and peaks along with the enriched GO terms

REFERENCE AND HELP

- ?ChIPpeakAnno in a R session
- browseVignettes("ChIPpeakAnno")
- Zhu LJ*, Gazin C, Lawson ND, Pages H, Lin SM, Lapointe DS and Green MR. (2010) [* denotes corresponding author] ChIPpeakAnno: a Bioconductor package to annotate ChIP-seq and ChIP-chip data. BMC Bioinformatics 2010, 11:237.

ACKNOWLEDGEMENT

- The Bioconductor package reviewers
  - Nishant Gopalak
  - Marc Carlson
  - other anonymous reviewers
- Coauthors
  - Claude Gazin
  - Nathan Lawson
  - Hervé Pagès
  - Simon Lin
  - David Lapointe
  - Michael Green
- The users of the *ChIPpeakAnno*
- The Bioconductor core team, esp.,
  - Patrick Aboyoun
  - Martin Morgan
- Ivan Gregoretti
- Amy Molesworth
- Khademul Islam
- Hua Li
- Zhiping Weng, Sara Evans , Alan Ritacco, Glenn Maston, Ping Wan, Ellen Kittler