ChIPpeakAnno

Jianhong Ou

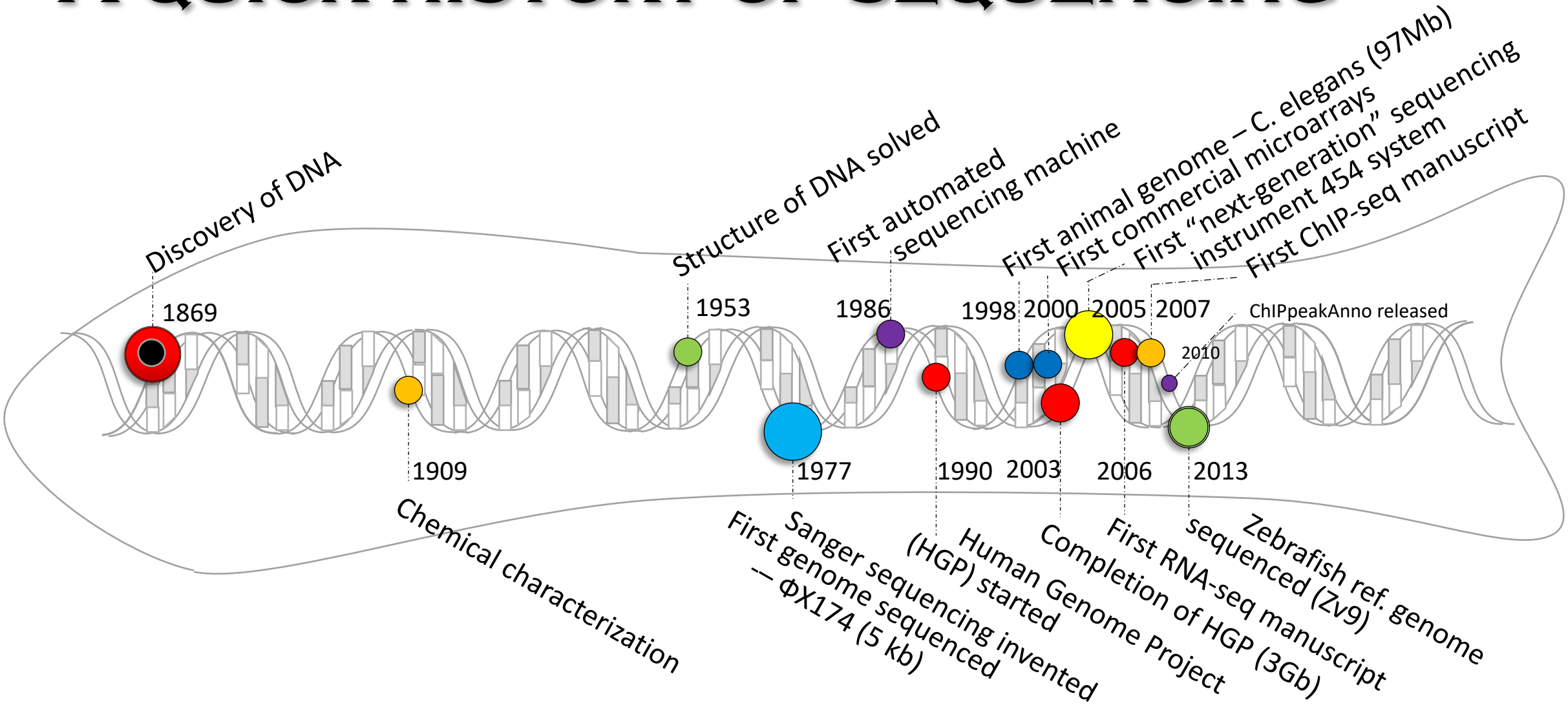
Julie Zhu

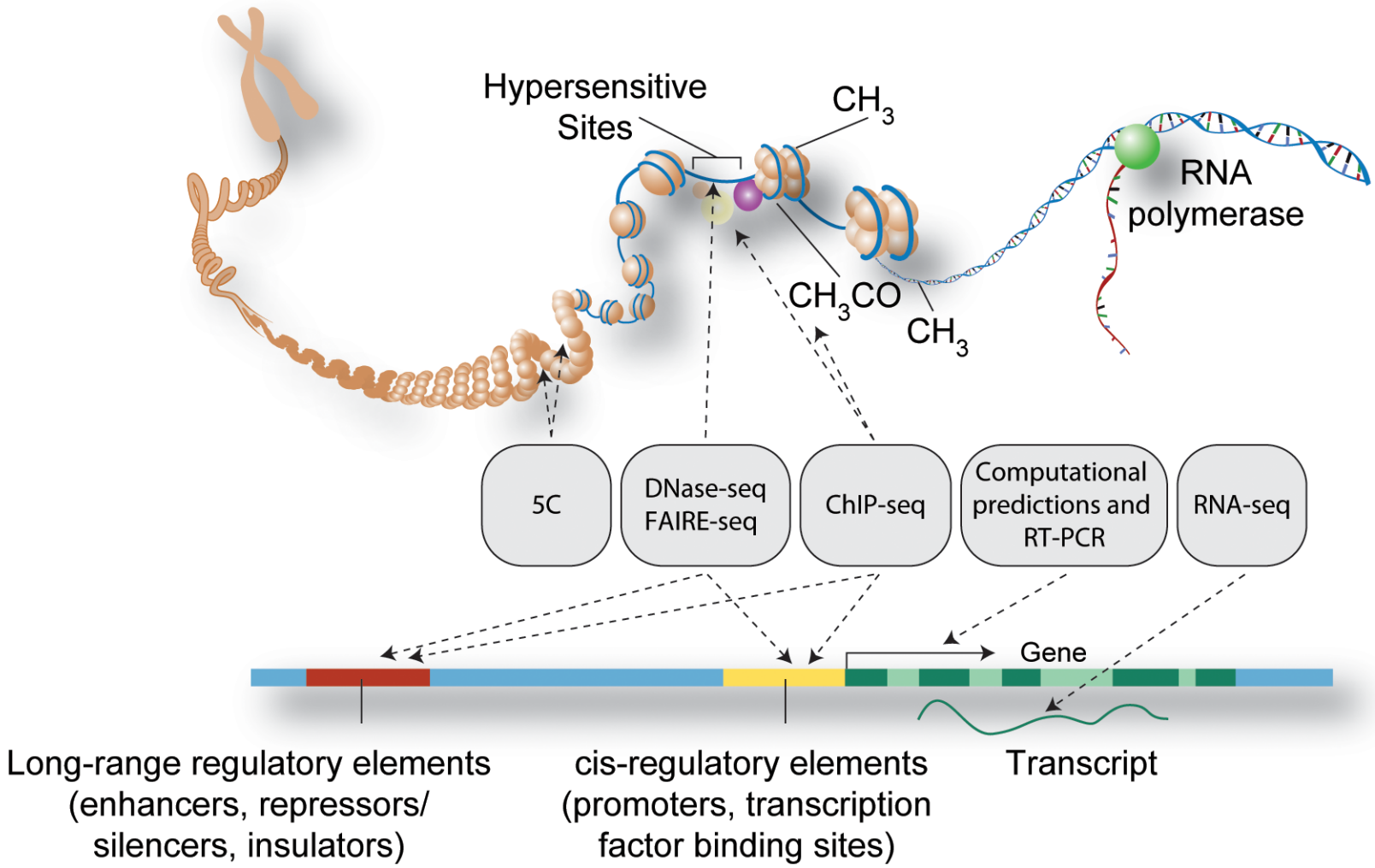
Install the workshop pkg

```
## set the working directory,
## replace "~/Downloads/ATACseqQCworkshop" by your path
wd <- "~/Downloads/workshop2020"
dir.create(wd)
setwd(wd)
library(BiocManager)
install("jianhong/workshop2020", build_vignettes = TRUE)
vignette("ChIPpeakAnno", package="workshop2020")
```

https://github.com/jianhong/workshop2020
https://bioconductor.org/packages/ChIPpeakAnno
Slides:
https://github.com/jianhong/workshop2020/blob/master/inst/extdata/ChIPpeakAnno_workshop2020.pdf

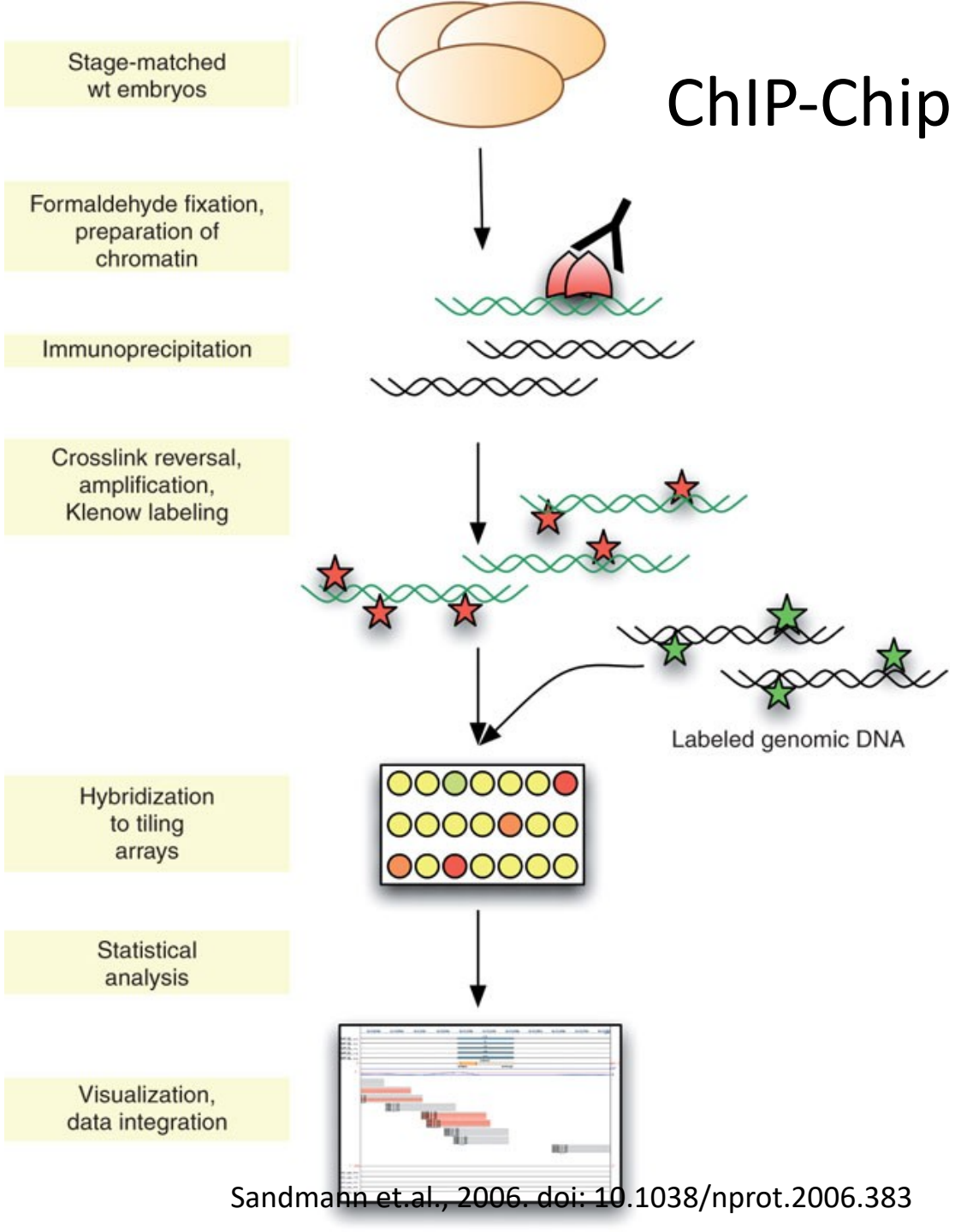
A QUICK HISTORY OF SEQUENCING





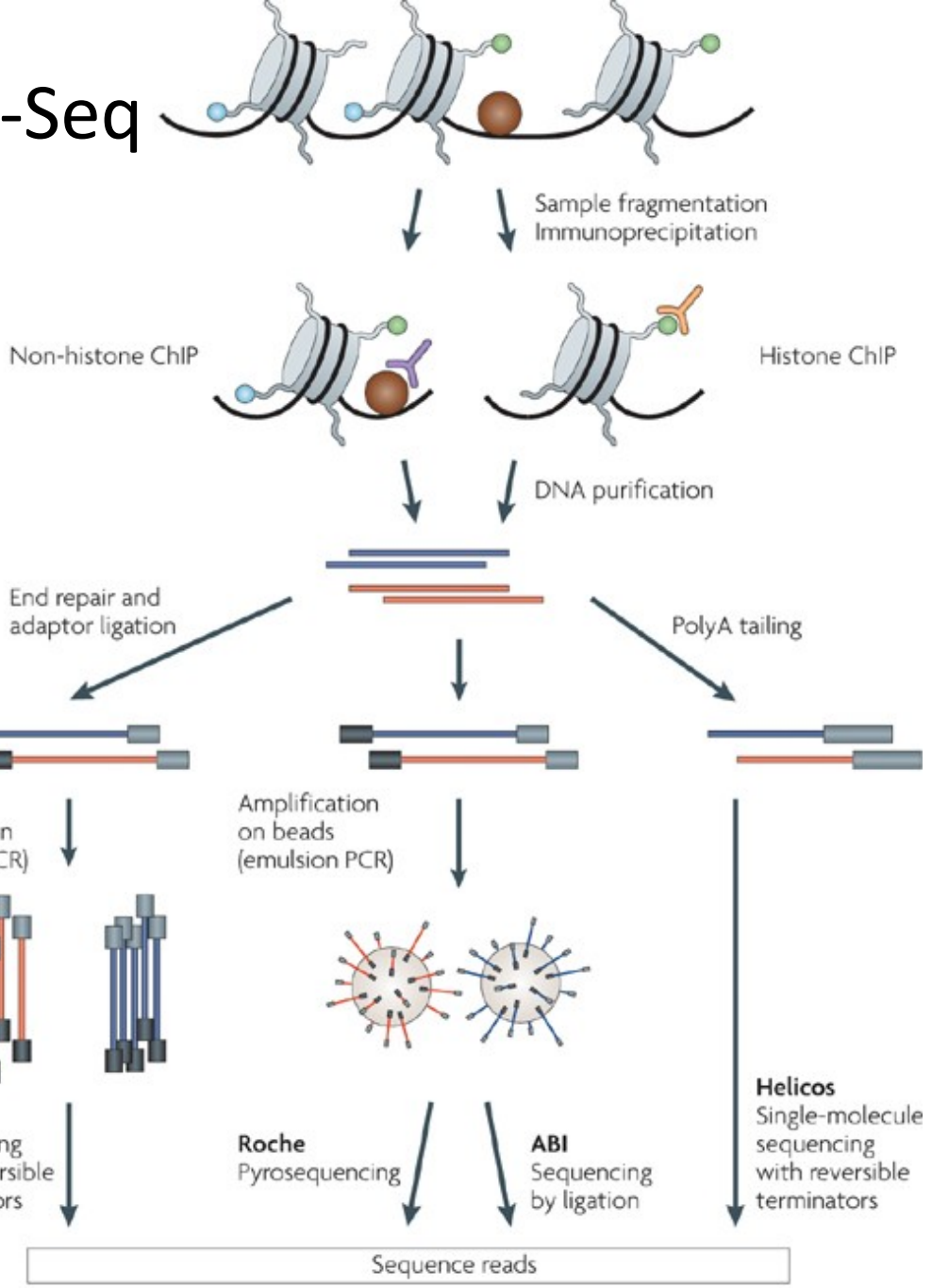
The ENCODE Project Consortium 2011. doi: 10.1371/journal.pbio.1001046

ChIP-Chip



Sandmann et.al., 2006. doi: 10.1038/nprot.2006.383

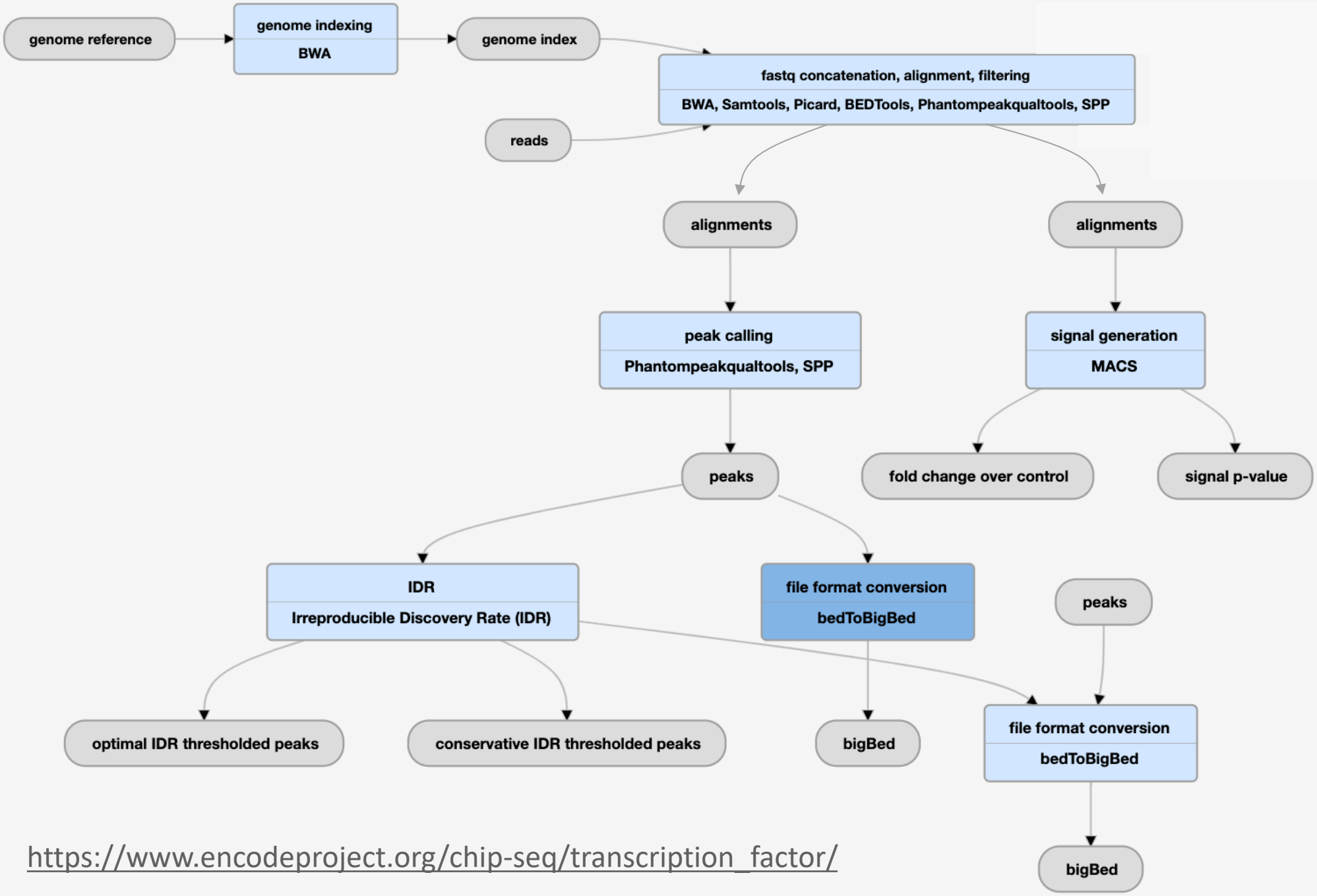
ChIP-Seq



Park 2009. doi:10.1038/nrg2641

Nature Reviews | Genetics

ChIP-seq Data Processing Pipeline



https://www.encodeproject.org/chip-seq/transcription_factor/

Why we need an annotation tool like ChIPpeakAnno?

Reproducibility

Accuracy of an experimental claim can be checked by complete obedience to the protocol in the published analysis. The program or portable source code involved in each published analysis should be trackable by version control system.

Transparency

Credible work for high-throughput experiment requires exposure of the entire process. There should not be any hidden step. The program or source code and the data for each publication should be accessible.

Efficiency

Software and data resources in an open-source environment can be re-used, modified or extended to achieve new functionalities. This is particularly effective when good documentation protocols are established.

Avoidance of traps

Software submitted to Bioconductor accepts comments, questions and challenges. Over 8 years continuous development, ChIPpeakAnno become strong and informative to avoid general mistakes in annotation.

Main function list

I/O: toGRanges

The *toGRanges* function Convert UCSC BED format and its variants, such as GFF, or user defined dataset such as MACS output file to GRanges

Annotation: annotatePeakInBatch

The *annotatePeakInBatch* function will obtain the distance to nearest/given-range TSS, miRNA, and /or exons for a list of peak.

Set operation: findOverlapsOfPeaks

The *findOverlapsOfPeaks* function will findthe overlapped peaks among two or more (less than five) set of peaks.

Enrichment: getEnrichedGO

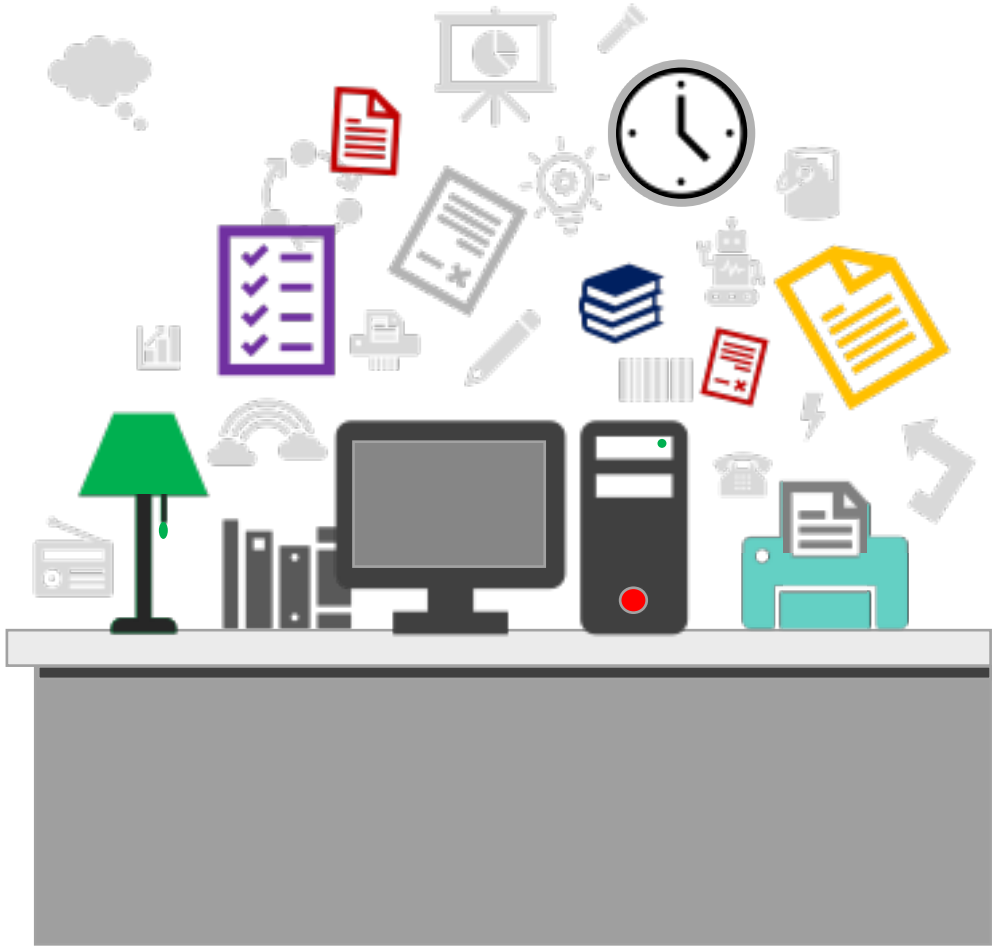
The *getEnrichedGO* function obtain the enriched gene ontology (GO) terms that near the peaks.

Metagene: featureAlignedExtendSignal

The *featureAlignedExtendSignal* function obtain extract signals in given ranges from bam files of DNA-seq. For RNA-seq, use *featureAlignedSignal* instead.

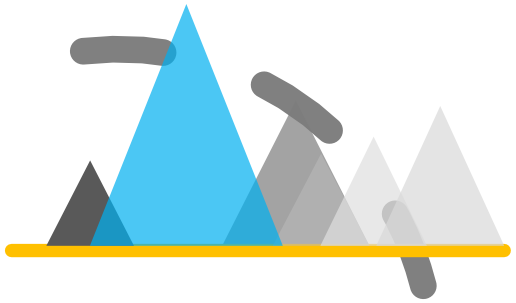
Motif search: summarizePatternInPeaks

The *summarizePatternInPeaks* function Output a summary of the occurrence of each pattern in the sequences



Steps of annotation by ChIPpeakAnno

The functions, *toGRanges*, *annotatePeakInBatch*, and *addGeneIDs* in the **ChIPpeakAnno**, make the annotation of ChIP-Seq peaks streamlined into four major steps:



Read peak data
with *toGRanges*

Generate annotation data
with *toGRanges*

Annotate peaks
with *annotatePeakInBatch*

Add additional
information
with *addGeneIDs*

0-BASED VS. 1-BASED COORDINATES SYSTEMS
CLOSED VS. HALF-OPEN COORDINATES SYSTEMS

0/1

0-based vs. 1-based coordinates systems

0-based coordinates system is used by programmer, 1-based coordinates system is used by biologists. The difference is that 0-based coordinates system start counting from 0 where 1-based coordinates system start counting from 1.

1/)

Closed vs. Half-open coordinates systems

A coordinates system use two number to indicate start and end position. There are two kinds of combination for the numbers: 1. start and end; 2 start and step. The following examples use start and end combination.

Closed coordinates: [start, end] == start ≤ x ≤ end, where width = end – start + 1

Half-open means half-closed-half-open coordinates: [start, end) == start ≤ x < end, where width = end – start

∞

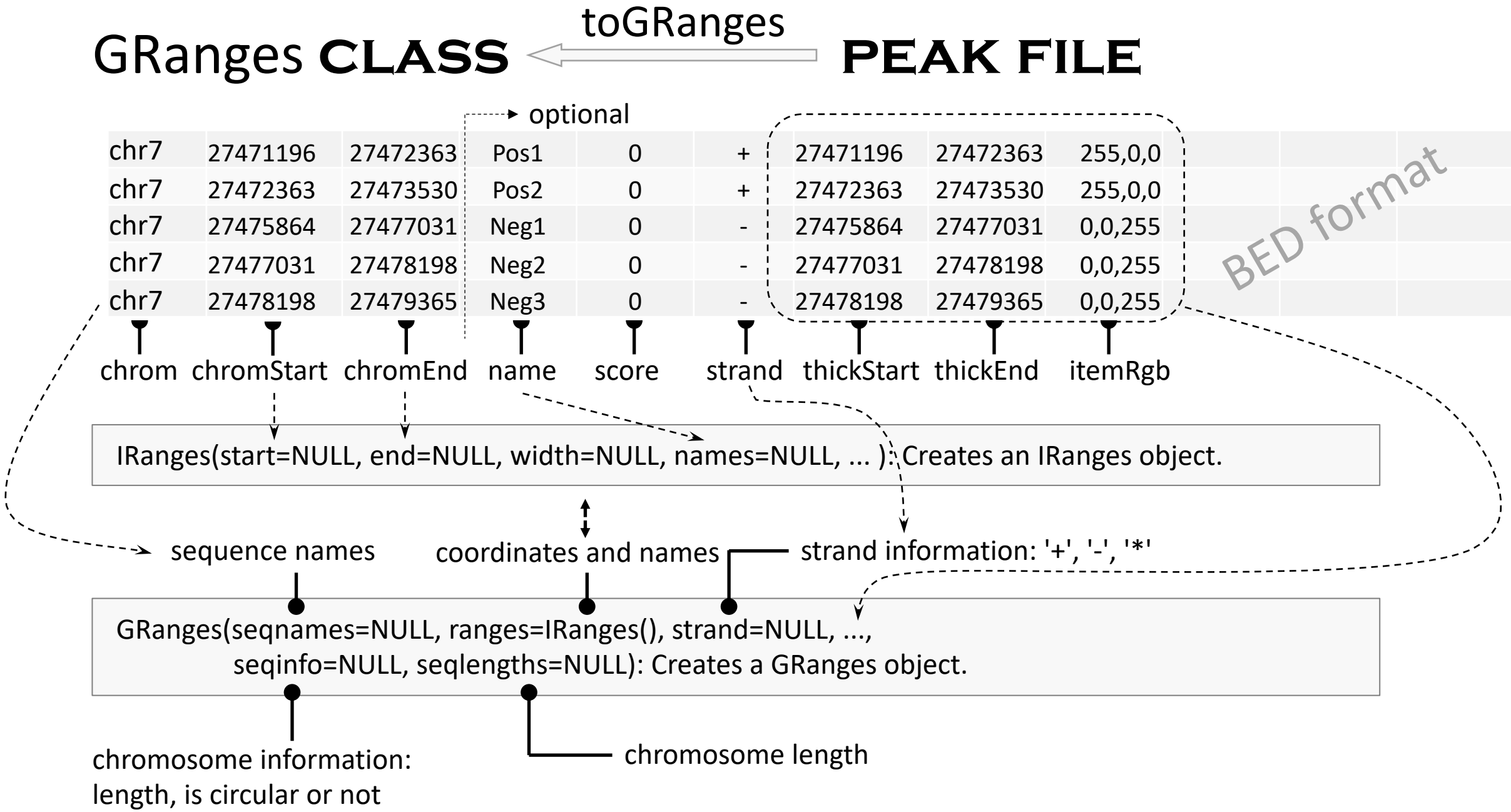
Most of 0-based coordinates system use half-open coordinates, where 1-based coordinates system use closed coordinates.

1-based coordinates system: SAM, VCF, GFF/GTF and Wiggle formats.

0-based coordinates system: BAM, BCFv2, BED, and PSL formats.

IRanges/GRanges are using **closed** coordinates system. There is no definition about 0-based or 1-based coordinates for IRanges/GRanges (IRanges(0, 0)). The *rtracklayer* package treat it as **1-based** coordinates and *ChIPpeakAnno* follows this rule.

https://genome.ucsc.edu/FAQ/FAQformat.html


GRanges CLASS
toGRanges
PEAK FILE
optional
chr7 27471196 27472363 Pos1 0 + 27471196 27472363 255,0,0
chr7 27472363 27473530 Pos2 0 + 27472363 27473530 255,0,0
chr7 27475864 27477031 Neg1 0 - 27475864 27477031 0,0,255
chr7 27477031 27478198 Neg2 0 - 27477031 27478198 0,0,255
chr7 27478198 27479365 Neg3 0 - 27478198 27479365 0,0,255
BED format
chrom chromStart chromEnd name score strand thickStart thickEnd itemRgb
IRanges(start=NULL, end=NULL, width=NULL, names=NULL, ...): Creates an IRanges object.
sequence names
coordinates and names
strand information: '+', '-', '*'
GRanges(seqnames=NULL, ranges=IRanges(), strand=NULL, ...,
seqinfo=NULL, seqlengths=NULL): Creates a GRanges object.
chromosome information:
length, is circular or not
chromosome length

INPUT DATA FOR ANNOTATION, *toGRanges*

```
## the sample file is included in ChIPpeakAnno package.
## chage the file path into your own file path to handle your data
path <- system.file("extdata", "Tead4.broadPeak", package="ChIPpeakAnno")
## toGRanges is overlaoded method,
## by define the correct file format to import the file in correct coordinates
peaks <- toGRanges(path, format="broadPeak")
## see top 2 lines of the imported peaks.
## the imported peaks will be packaged into GRanges object
head(peaks, n=2)
```

```
## GRanges object with 2 ranges and 4 metadata columns:
##             seqnames        ranges strand |     score signalValue    pValue
##                <Rle>     <IRanges>  <Rle> | <integer>   <numeric> <numeric>
##   peak12338     chr2 175473-176697      * |       206      668.42        -1
##   peak12339     chr2 246412-246950      * |        31      100.23        -1
##               qValue
##             <numeric>
##   peak12338        -1
##   peak12339        -1
##   -------
##   seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

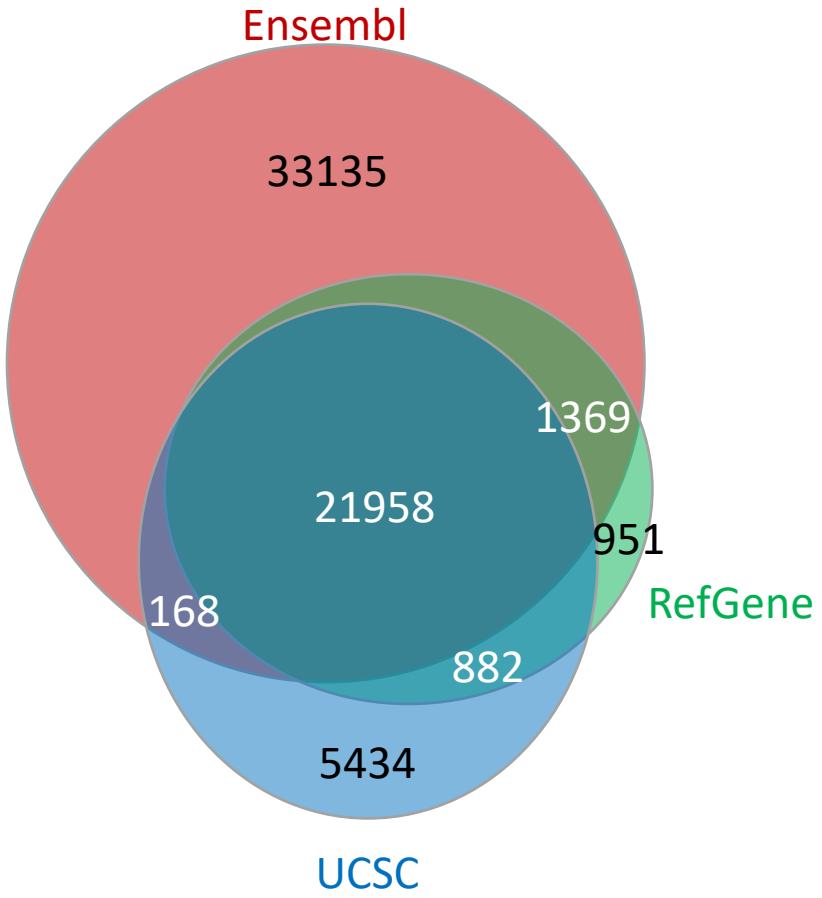
Important concepts in annotation

Annotation: Computational process of attaching biologically relevant information to a given data.

Assembly: Computational reconstruction of a longer sequence from smaller sequence reads.

Genome assembly: A computational representation of the sequence of a haploid genome, representative of a species or strain.

Ensembl, RefSeq and UCSC annotations



The overlap and intersection among RefGene, UCSC(hg18), and Ensembl (GRCh37) annotations

Zhao et.al., 2015. doi: 10.1186/s12864-015-1308-8

Ensembl: annotated by genebuild pipeline + manually annotated by the HAVANA team

UCSC: also called "Known Gene", is available only on assemblies before hg38. It was built with UCSC gene predictor which uses protein, EST and cDNA annotations.

RefSeq: most stringent. RefSeq also have additional sequences independent of the genome assembly. This has the important implication that the position of genome variants are harder to map to RefSeq transcripts.

For **RNA-seq**:

Less complex genome annotations, such as RefGene, are preferable for reproducible and robust gene expression estimates.

However, to discover and explain unknown biological mechanisms, more comprehensive and complex genome annotations are necessary, such as Ensembl.

Wu et.al., 2013. doi: 0.1186/1471-2105-14-S11-S8

Genome assembly versions

SPECIES	UCSC VERSION	RELEASE DATE	Ensembl VERSION	RELEASE DATE	ASSEMBLY NAME
Human	hg38	Dec. 2013	GRCh38	P13: Sep. 2019 (>=V98); P12: Apr. 2018 (>=V92, <=V97); P10: Mar. 2017 (>=V88, <=V91); P7: Jul. 2016 (>=V85, <=V87); P5: Dec. 2015 (>=V83, <=V84); P3: Jul. 2015 (>=V81, <=V82); P2: Mar. 2015 (>=V79, <=V80); Aug. 2014 (>=V76, <=V78);	Genome Reference Consortium GRCh38
	hg19	Feb. 2009	GRCh37	P13: Dec. 2013 (>=V74, <=V75); P12: Sep. 2013 (=V73); P11: Jun. 2013 (=V72); P10: Apr. 2013 (=V71); P8: Jul. 2012 (>=V68, <=V70); P7: May 2012 (=V67); P6: Feb. 2012 (=V66); P5: Sep. 2011 (>=V64, <=V65); P3: Jun. 2011 (=V63); Jul. 2009 (>=V55, <=V62);	Genome Reference Consortium GRCh37
Mouse	mm10	Dec. 2011	GRCm38	P6: Apr. 2018 (>=V92); P5: Dec. 2016 (>=V87, <=V91); P4: Jul. 2015 (>=V81, <=V86); P3: Dec. 2014 (>=V78, <=V80); P2: Dec. 2013 (>=V74, <=V77); P1: Jan. 2013 (>=V70, <=V73); Jul. 2012 (>=V68, <=V69)	Genome Reference Consortium GRCm38
	mm9	Jul. 2007	NCBIM37	Oct. 2007(>=V47)	NCBI Build 37
Zebrafish	danRer11	May 2017	GRCz11	Apr. 2018 (>=V92)	Genome Reference Consortium GRCz11
	danRer10	Sep. 2014	GRCz10	May 2015 (>=V80, <=V91)	Genome Reference Consortium GRCz10
D. melanogaster	dm6	Aug. 2014	BDGP6	Mar. 2015 (>=V79)	BDGP Release 6 + ISO1 MT
	dm3	Apr. 2006	BDGP5	NA	BDGP Release 5
C. elegans	ce11	Feb. 2013	WBcel235	Apr. 2013 (>=V71)	C. elegans Sequencing Consortium WBcel235
	ce10	Oct. 2010	WS220	Feb. 2011 (>=V61, <=V66)	WormBase v. WS220

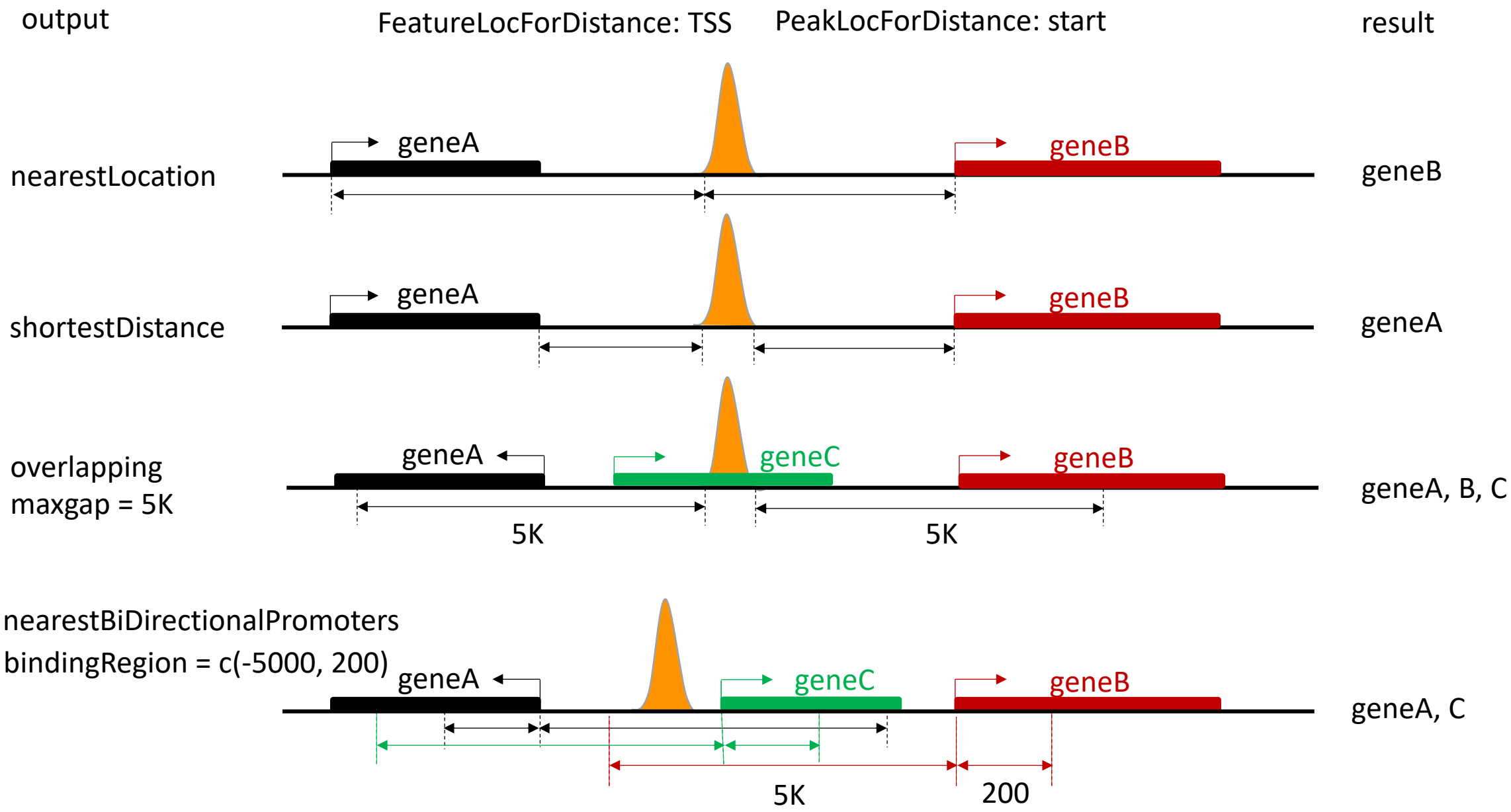
https://genome.ucsc.edu/FAQ/FAQreleases.html
https://useast.ensembl.org/info/website/archives/assembly.html

Prepare annotation data, *toGRanges*

```
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
annoData <- toGRanges(TxDb.Hsapiens.UCSC.hg19.knownGene, feature="gene")
annoData[1]
```

```
## GRanges object with 1 range and 0 metadata columns:
##    seqnames              ranges  strand
##       <Rle>           <IRanges>   <Rle>
## 1     chr19 58858172-58874214       -
## -------
## seqinfo: 93 sequences (1 circular) from hg19 genome
```

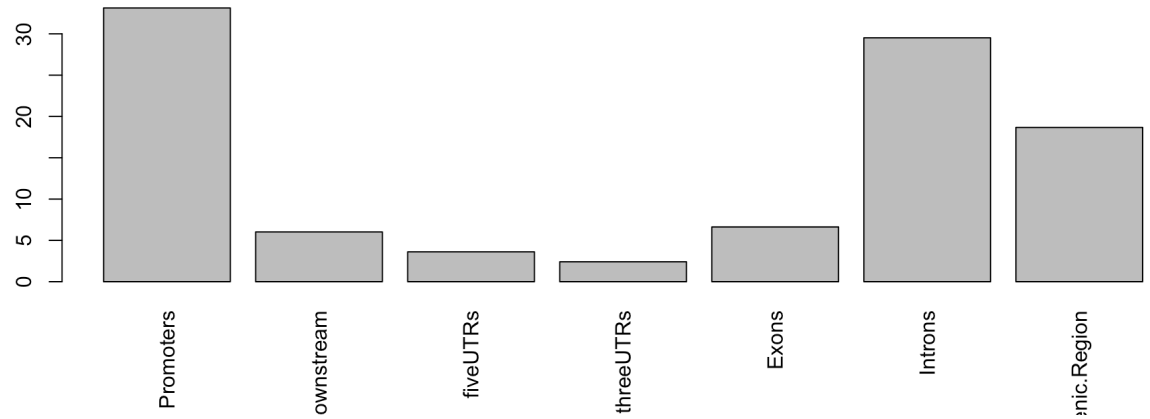
ANNOTATE BY annotatePeakInBatch



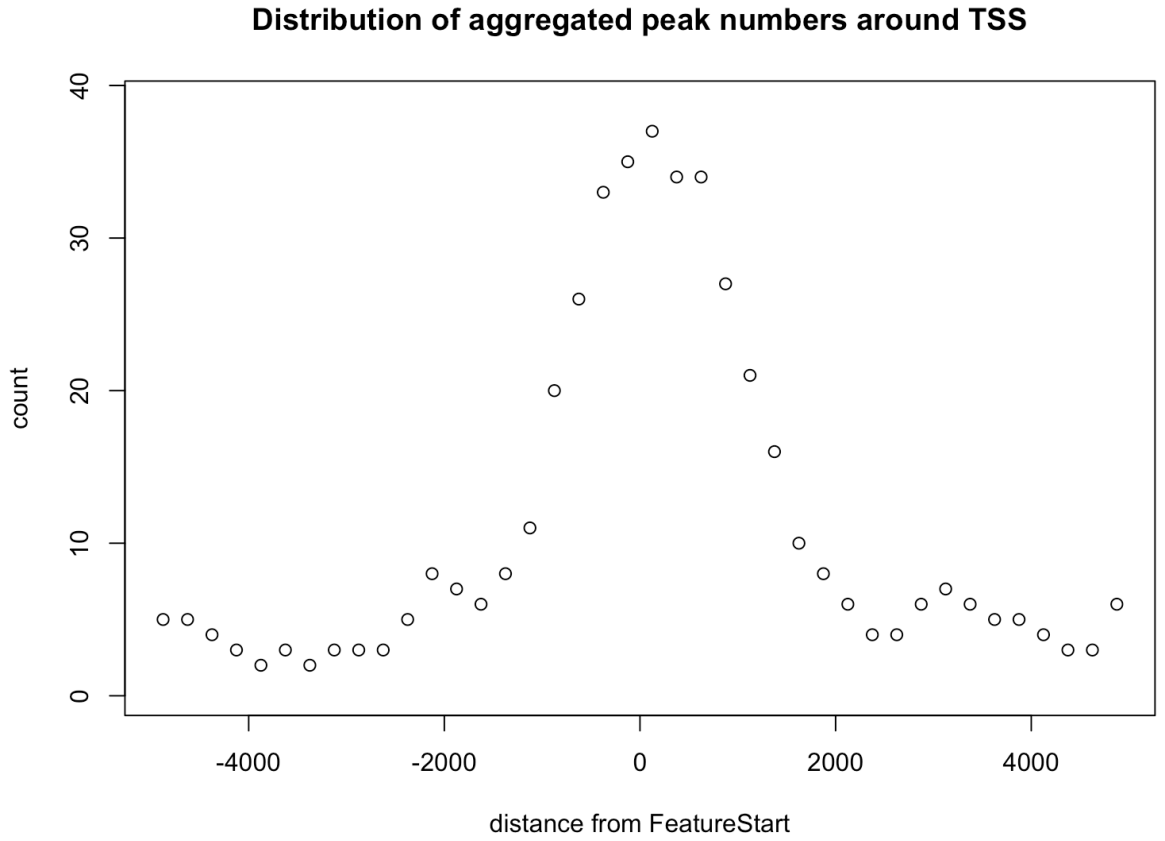
Select annotation method

Visualize binding site distribution relative to features

```
aCR<-assignChromosomeRegion(overlaps,
          nucleotideLevel=FALSE,
           precedence=c("Promoters",
"immediateDownstream", "fiveUTRs", "threeUTRs",
"Exons", "Introns"),
          TxDb=TxDb.Hsapiens.UCSC.hg19.knownGene)
barplot(aCR$percentage, las=3)
```




```
binOverFeature(overlaps, annotationData=annoDataTxDb,
                radius=5000, nbins=20, FUN=length,
                errFun=0, ylab="count",
                main="Distribution of aggregated peak numbers around TSS")
```



Annotate data by nearest features

```
## keep the seqnames in the same style
if(!identical(seqlevelsStyle(peaks), seqlevelsStyle(annoDataTxDb))){
 seqlevelsStyle(peaks) <- seqlevelsStyle(annoDataTxDb)[1]
}
## do annotation by nearest TSS of Ensembl GRCh37.p13
annoTxDb <- annotatePeakInBatch(peaks, AnnotationData=annoDataTxDb)
head(annoTxDb, n=1)
```

```
## GRanges object with 1 ranges and 13 metadata columns:
##                     seqnames        ranges strand |     score signalValue
##                        <Rle>     <IRanges>  <Rle> | <integer>   <numeric>
##   peak12338.26751      chr2 175473-176697      * |       206      668.42
##                       pValue    qValue        peak     feature start_position
##                    <numeric> <numeric> <character> <character>      <integer>
##   peak12338.26751         -1        -1   peak12338       26751         218136
##                   end_position feature_strand insideFeature distancetoFeature
##                      <integer>    <character>   <character>         <numeric>
##   peak12338.26751       264810              -    downstream             89337
##                   shortestDistance fromOverlappingOrNearest
##                          <integer>              <character>
##   peak12338.26751            41439          NearestLocation
##   -------
##   seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

Add additional annotation, *addGeneIDs*

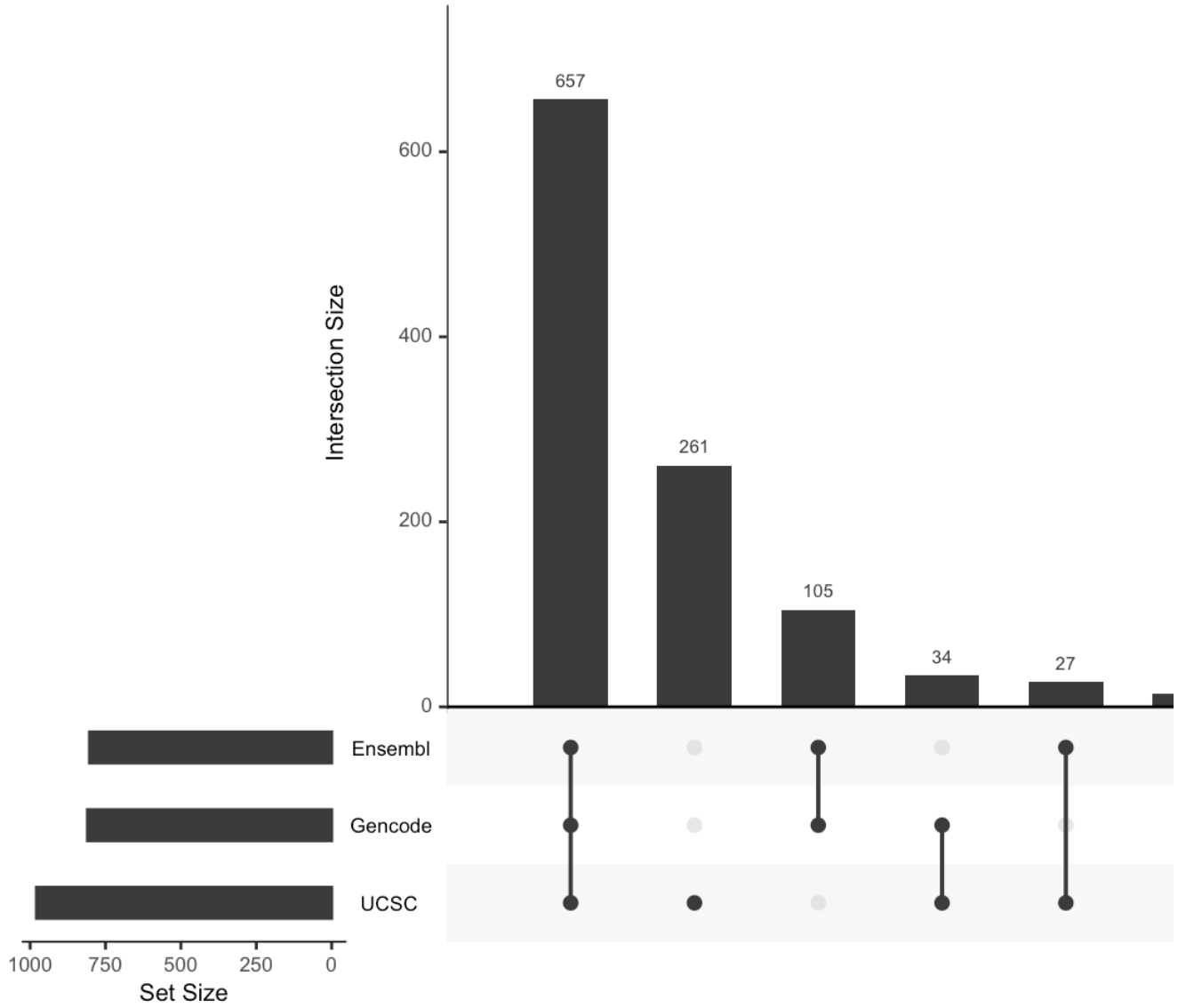
```
annoTxDb <- addGeneIDs(annoTxDb, orgAnn="org.Hs.eg.db", feature_id_type="entrez_id", IDs2Add=c("symbol"))
head(annoTxDb, n=2)
```

Use "entrez_id" for TxDb annotations;
Use ”ensemble_gene_id" for EnsDb annotations;

```
## GRanges object with 2 ranges and 13 metadata columns:
##                    seqnames        ranges strand |     score signalValue
##                       <Rle>     <IRanges>  <Rle> | <integer>   <numeric>
##   peak12338.26751     chr2 175473-176697      * |       206      668.42
##   peak12339.26751     chr2 246412-246950      * |        31      100.23
##                      pValue    qValue        peak     feature start_position
##                   <numeric> <numeric> <character> <character>      <integer>
##   peak12338.26751        -1        -1   peak12338       26751         218136
##   peak12339.26751        -1        -1   peak12339       26751         218136
##                   end_position feature_strand insideFeature distancetoFeature
##                      <integer>    <character>   <character>         <numeric>
##   peak12338.26751       264810              -    downstream             89337
##   peak12339.26751       264810              -        inside             18398
##                   shortestDistance fromOverlappingOrNearest
##                          <integer>             <character>
##   peak12338.26751            41439          NearestLocation
##   peak12339.26751            17860          NearestLocation
##   -------
##   seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

Annotations with different annotation scource

As a conclusion, annotate with different annotation resources, even the other parameter keep same, the annotations will be different from each other. To improve the reproducibility, accuracy of an annotation source should be provided.



upset plot of common gene symbols among annotations by Ensembl, UCSC and Gencode features.

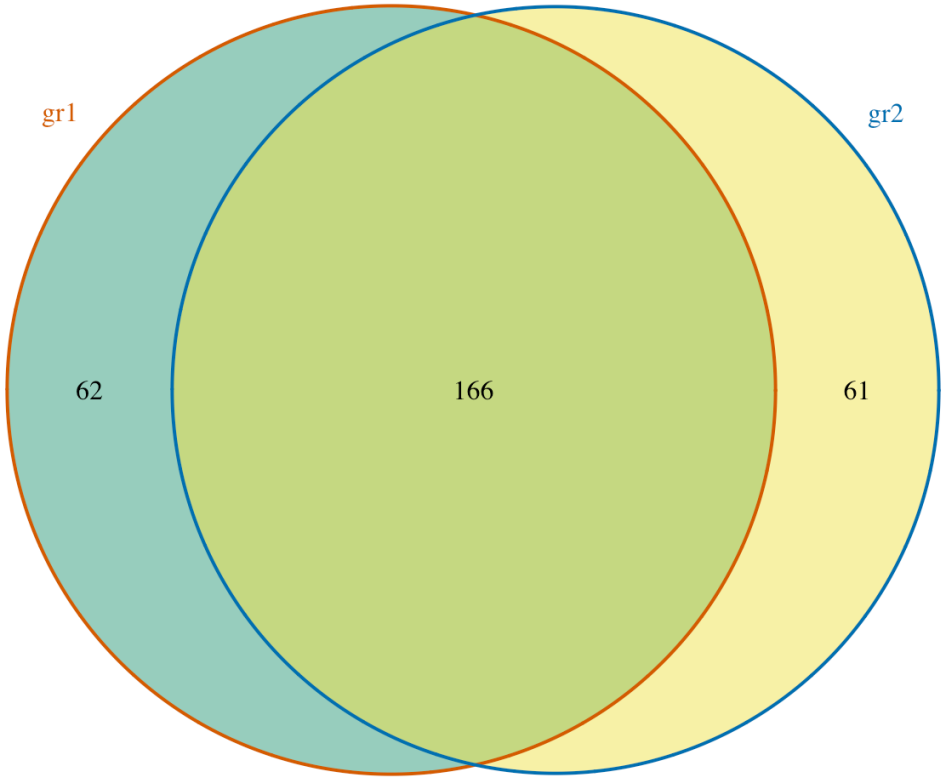
Find overlaps for replicates, *findOverlapsOfPeaks*

```
## import peaks from a bed file
bed <- system.file("extdata", "MACS_output.bed", package="ChIPpeakAnno")
gr1 <- toGRanges(bed, format="BED", header=FALSE)
## import peaks from a gff file
gff <- system.file("extdata", "GFF_peaks.gff", package="ChIPpeakAnno")
gr2 <- toGRanges(gff, format="GFF", header=FALSE, skip=3)
## find overlaps for replicates
ol <- findOverlapsOfPeaks(gr1, gr2, connectedPeaks = "keepAll")
head(ol$peaklist[["gr1///gr2"]], n=2)
```

```
## GRanges object with 2 ranges and 2 metadata columns:
##       seqnames          ranges strand |                              peakNames
##          <Rle>       <IRanges>  <Rle> |                        <CharacterList>
##   [1]     chr1 713791-715578      * | gr1__MACS_peak_13,gr2__001,gr2__002
##   [2]     chr1 724851-727191      * |          gr2__003,gr1__MACS_peak_14
##   -------
##   seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

Find overlaps for replicates, *findOverlapsOfPeaks*

```
makeVennDiagram(ol, fill=c("#009E73", "#F0E442"),
                col=c("#D55E00", "#0072B2"),
                cat.col=c("#D55E00", "#0072B2"))
```

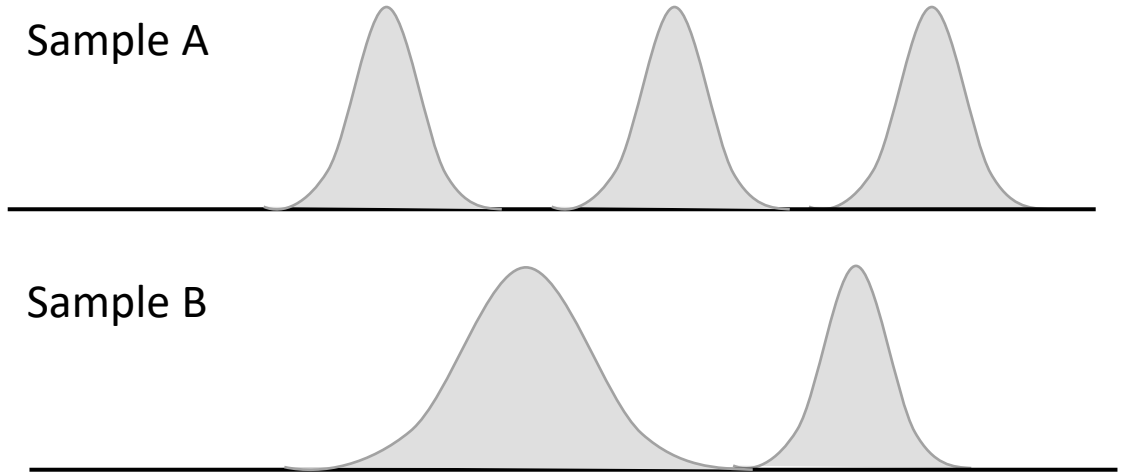


connectedPeaks:

"min" : minimal involved peaks → 2

"merge" : 1

"keepAll": keep all the original counts for each list while the final counts will be same as "min"



Enrichment: Obtain enriched GO terms and Pathways

Use *getEnrichedGO* to obtain a list of enriched GO terms with annotated peaks. For pathway analysis, please use function *getEnrichedPATH* with reactome or KEGG database.

```
getEnrichedGO(annotatedPeak, orgAnn,
              feature_id_type="ensembl_gene_id",
              maxP=0.01, minGOterm=10,
              multiAdjMethod=NULL,
              condense=FALSE,
              removeAncestorByPval=NULL,
              keepByLevel=NULL)
```

The *getEnrichedGO* function obtain enriched gene ontology (GO) terms based on the features near the enriched peaks using GO.db package and GO gene mapping package such as org.Hs.db.eg to obtain the GO annotation and using hypergeometric test (phyper) and multtest package for adjusting p-values.

```
getEnrichedPATH(annotatedPeak, orgAnn,
                pathAnn,
                feature_id_type="ensembl_gene_id",
                maxP=0.01,
                minPATHterm=10,
                multiAdjMethod=NULL)
```

The *getEnrichedPATH* function obtain enriched PATH that are near the peaks using path package such as reactome.db and path mapping package such as org.Hs.db.eg to obtain the path annotation and using hypergeometric test (phyper) and multtest package for adjusting p-values.

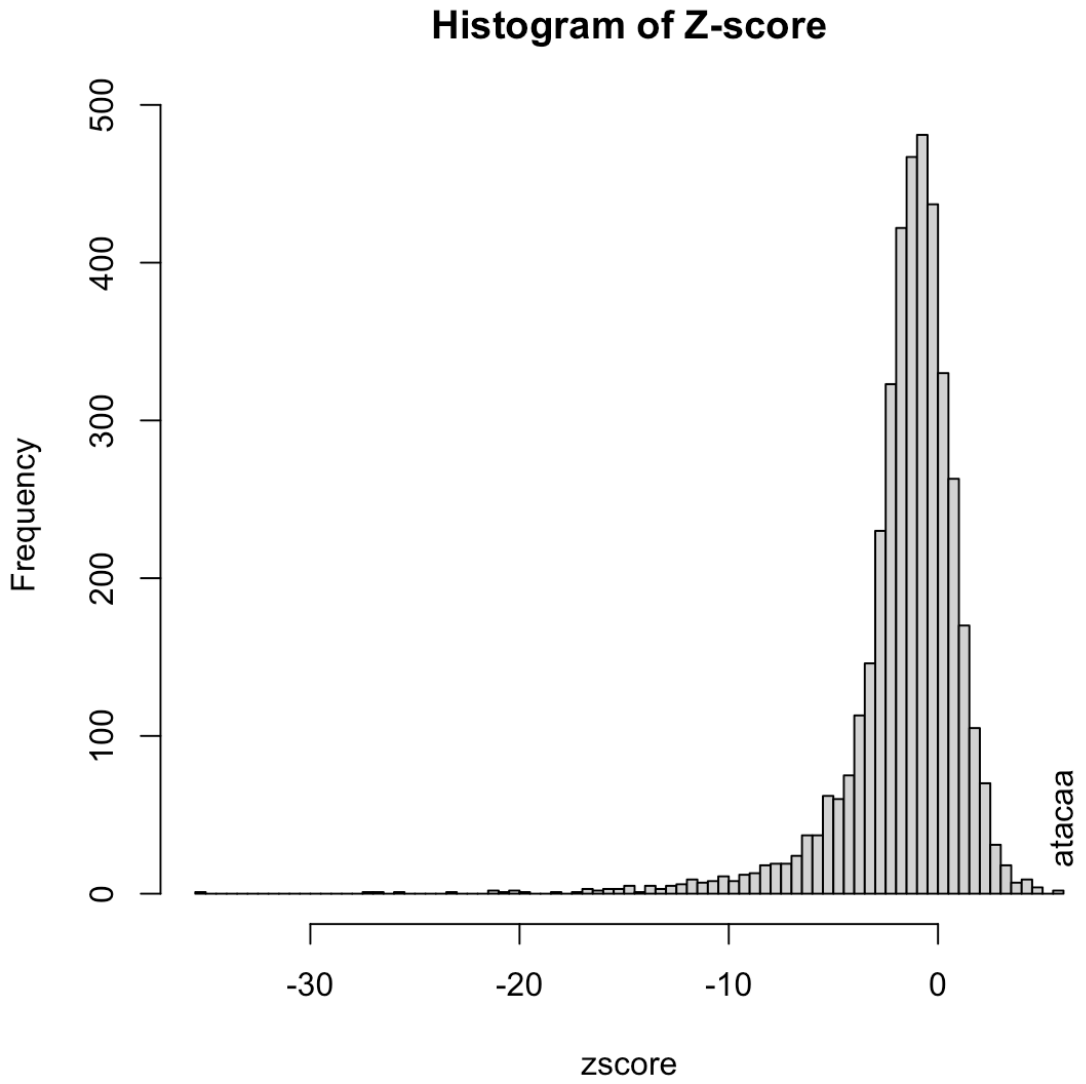
Motif search: Output a summary of consensus in the peaks

There are multiple methods to get the consensus in the peaks:

1. output the fastq file by the *getAllPeakSequence* function and search the motif by the 3rd program such as homer, MEME, and so on.

2. test the pre-defined consensus patterns to see if target consensus are enriched or not by the *summarizePatternInPeaks* function.

3. calculate the z-scores of all combinations of oligonucleotide in a given length by Markove chain by the *oligoSummary* function.

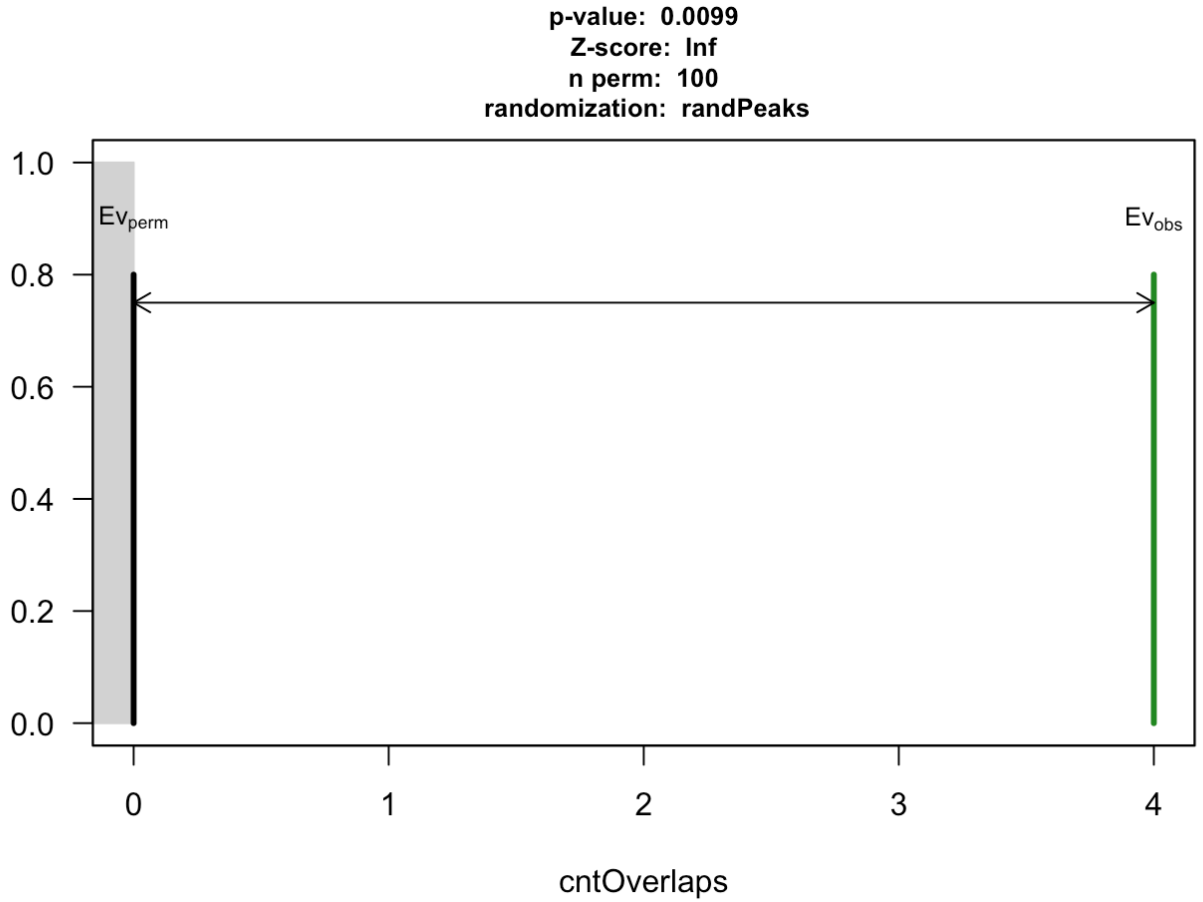


Determine if there is a significant overlap among multiple sets of peaks

makeVennDiagram: The p.value is calculated by **hypergeometric test** to determine whether the peaks or features are overlapped significantly. The number of all potential binding sites is required.

peakPermTest: Performs a **permutation test** to see if there is an association between two given peak lists.

```
## $p.value
##      TAF Tead4 YY1          pval
## [1,]   0     1   1  1.000000e+00
## [2,]   1     0   1 2.904297e-258
## [3,]   1     1   0  8.970986e-04
##
## $vennCounts
##      TAF Tead4 YY1 Counts count.TAF count.Tead4
## [1,]   0     0   0    849         0           0
## [2,]   0     0   1    621         0           0
## [3,]   0     1   0   2097         0        2097
## [4,]   0     1   1    309         0         310
## [5,]   1     0   0     59        59           0
## [6,]   1     0   1    166       172           0
## [7,]   1     1   0      8         8           8
## [8,]   1     1   1    476       545         537
## attr(,"class")
## [1] "VennCounts"
```



Save annotation results

The annotation results can be saved in XLS file format using **WriteXLS** package to avoid the gene name errors that can be inadvertently introduced when opened by Excel.

```
library(WriteXLS)
WriteXLS(as.data.frame(unname(overlaps.anno)), "anno.xls")
```

gene names | internal date format | default date format | gene names | internal date format | default date format | gene names | internal date format | default date format

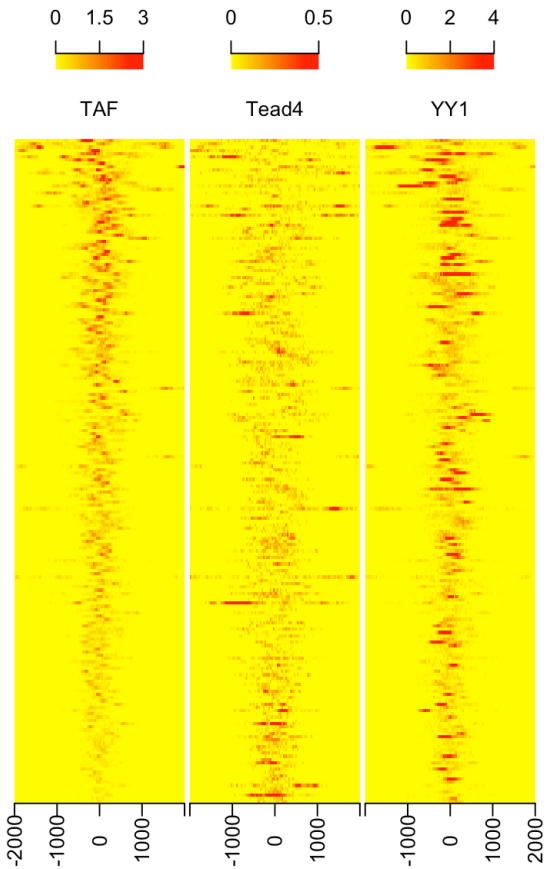
excel.gene2date.xls

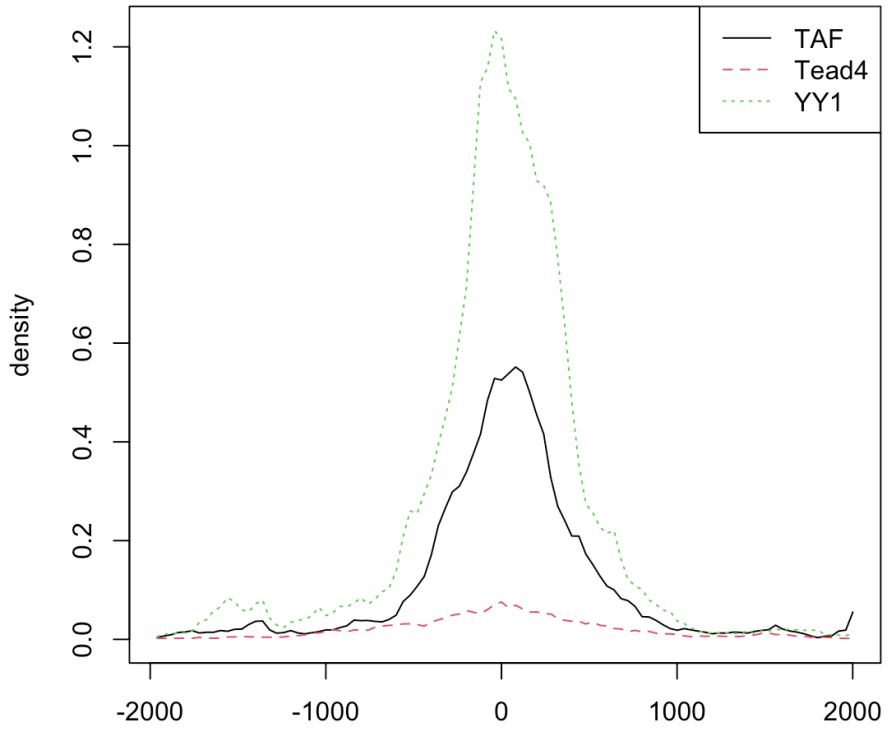
	A	B	C	D	E	F	G	H	I	J	K
1	APR-1	35885	1-Apr		OCT-1	36068	1-Oct		SEP2	36039	2-Sep
2	APR-2	35886	2-Apr		OCT-2	36069	2-Oct		SEP3	36040	3-Sep
3	APR-3	35887	3-Apr		OCT-3	36070	3-Oct		SEP4	36041	4-Sep
4	APR-4	35888	4-Apr		OCT-4	36071	4-Oct		SEP5	36042	5-Sep
5	APR-5	35889	5-Apr		OCT-6	36073	6-Oct		SEP6	36043	6-Sep
6	DEC-1	36129	1-Dec		OCT1	36068	1-Oct		SEPT1	36038	1-Sep
7	DEC-2	36130	2-Dec		OCT11	36078	11-Oct		SEPT2	36039	2-Sep
8	DEC1	36129	1-Dec		OCT2	36069	2-Oct		SEPT3	36040	3-Sep
9	DEC2	36130	2-Dec		OCT3	36070	3-Oct		SEPT4	36041	4-Sep
10	MAR1	35854	1-Mar		OCT4	36071	4-Oct		SEPT5	36042	5-Sep
11	MAR2	35855	2-Mar		OCT6	36073	6-Oct		SEPT6	36043	6-Sep
12	MAR3	35856	3-Mar		OCT7	36074	7-Oct		SEPT7	36044	7-Sep
13	NOV1	36099	1-Nov		SEP-1	36038	1-Sep		SEPT8	36045	8-Sep
14	NOV2	36100	2-Nov		SEP-2	36039	2-Sep		SEPT9	36046	9-Sep
15					SEP1	36038	1-Sep				

Sheet1 Sheet2

Ready Sum=0 SCRL CAPS NUM

Zeeberg et.al., 2004. doi: 10.1186/1471-2105-5-80





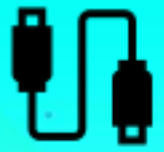
Metagene analysis for given features/peaks

The *featureAlignedExtendSignal* function obtain extract signals in given ranges from bam files of DNA-seq. For RNA-seq, use *featureAlignedSignal* instead.
Function featureAlignedHeatmap and featureAlignedDistribution can be used to visualize and compare the binding patterns of raw signals of multiple ChIP-Seq experiments.

ChIPpeakAnno CAN...

Annotation: annotatePeakInBatch

The *annotatePeakInBatch* function will obtain the distance to nearest/given-range TSS, miRNA, and /or exons for a list of peak.



I/O: toGRanges

The *toGRanges* function Convert UCSC BED format and its variants, such as GFF, or user defined dataset such as MACS output file to GRanges

Enrichment: getEnrichedGO/PATH

The *getEnrichedGO* and *getEnrichedPATH* function obtain the enriched gene ontology (GO) terms that near the peaks.

Set operation: findOverlapsOfPeaks

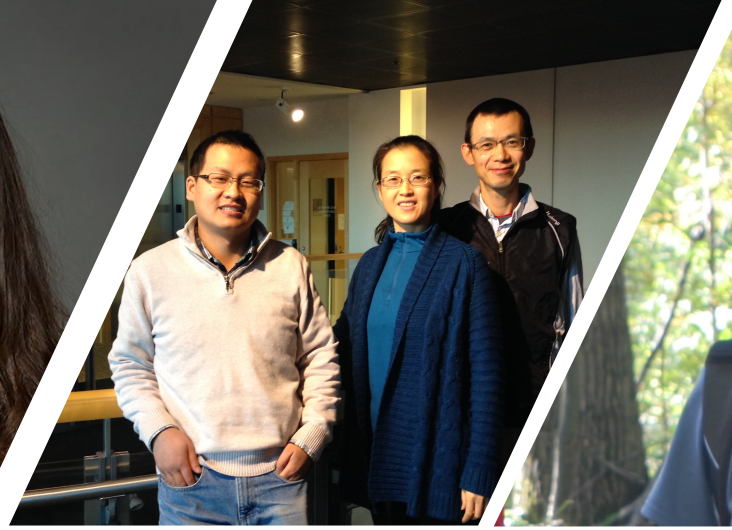
The *findOverlapsOfPeaks* function will findthe overlapped peaks among two or more (less than five) set of peaks.

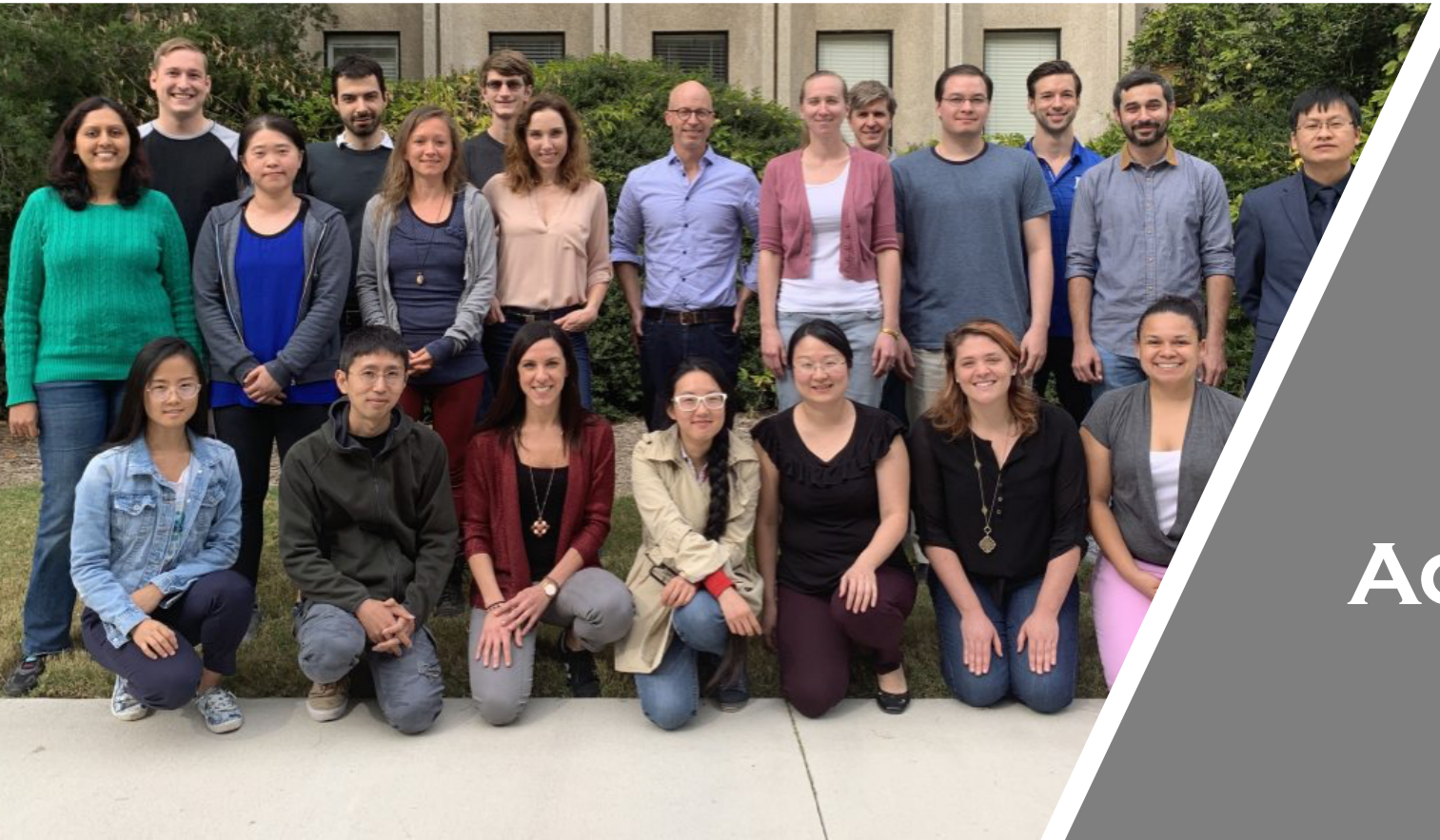
Metagene: featureAlignedExtendSignal

The *featureAlignedExtendSignal* function obtain extract signals in given ranges from bam files of DNA-seq. For RNA-seq, use *featureAlignedSignal* instead.

Motif search: oligoSummary

The *summarizePatternInPeaks* and *oligoSummary* function Output a summary of the occurrence of each pattern or all combinations of oligonucleotide in the sequences

Acknowledgement