Working with peaks in R

2022-8-25, Franziska Greulich

What type of peak files do we have?

1. MACS peak files

a.) _peaks.narrowPeak :

- BED6+4 format file

1. Chromosome name
2. Peak start
3. Peak end
4. Name
5. Score (UCSC visualization, 0…1000, encodes signal intensity)
6. Strand (usually unstranded .)

Standard BED6

7. Fold-change at peak summit
8. -log10(p-value) at peak summit
9. -log10(q-value) at peak summit
10. Relative summit position to start (0-based)

NarrowPeak file specific

b.) _summits.bed: peak summits locations for every peak in BED format (1-5 from above)

What type of peak files do we have?

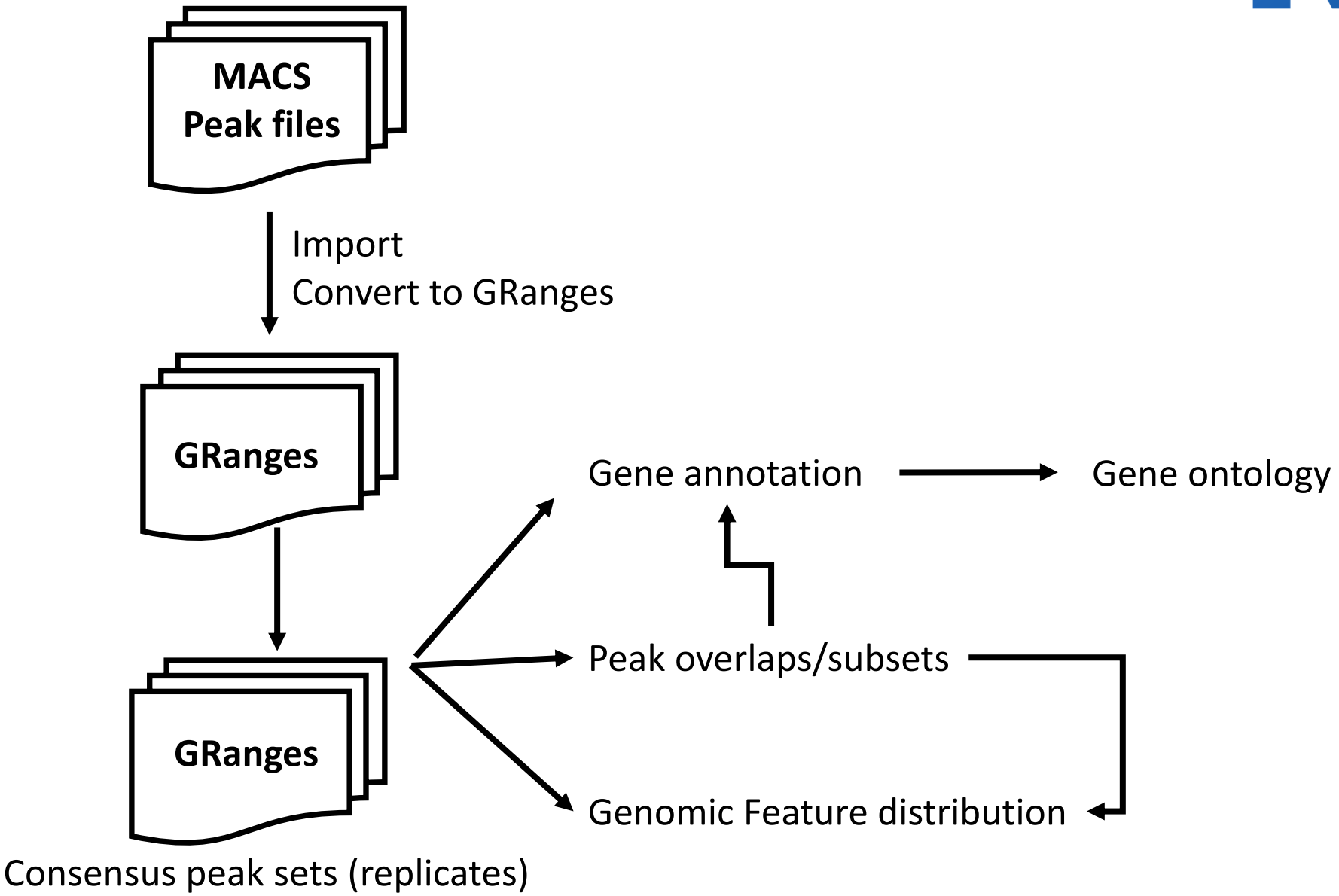
1. MACS peak files

c.) _peaks.xls:

- tabular file which contains information about called peaks.

  - Chromosome name
  - Peak start
  - Peak end
  - Peak width
  - Absolute peak summit
  - Pileup peak height at peak summit
  - -log10(p-value) at peak summit
  - Fold-enrichment for peak summit against random Poisson with local lambda
  - -log10(q-value) at peak summit

Workflow


MACS
Peak files
Import
Convert to GRanges
GRanges
GRanges
Gene annotation
Gene ontology
Peak overlaps/subsets
Genomic Feature distribution
Consensus peak sets (replicates)

Import peak files into R

1. import BED files

```
#read input data in BED format: Example of a _summit.bed MACS output.
peaks_bed <- read.table("./data/MACS/hMonocytes_4h1uMTA_aGR_rep1_SRR6497267overSRR6497265_SE_hg38_FDR005_summits.bed", heade
r=FALSE, sep='\t', stringsAsFactors=FALSE)
```

2. import narrowPeak files

Similar to BED files, narrowPeak files are standard tab-delimited and can be read using the `read.table` function of R. They do not contain a header, so we have to define the column names manually.

```
#read input data in narrowPeak format: Example of a _peaks.narrowPeak MACS output.
peaks_narrow <- read.table("./data/MACS/hMonocytes_4h1uMTA_aGR_rep1_SRR6497267overSRR6497265_SE_hg38_FDR005_peaks.narrowPea
k", header=FALSE, sep='\t', stringsAsFactors=FALSE)
```

3. import MACS .xls files

In contrast to BED and NarrowPeak files, MACS .xls files contain meta data within the first rows as well as column headers. By default the `read.table` function ignores commented rows. Set the header argument to `header=TRUE`.

```
#read input data in XLS format: Example of a _peaks.xls MACS output.
peaks_macs <- read.table("./data/MACS/hMonocytes_4h1uMTA_aGR_rep1_SRR6497267overSRR6497265_SE_hg38_FDR005_peaks.xls", header
=TRUE, sep='\t', stringsAsFactors=FALSE)
```

Import peak files into R – Excercise (20 mins)

MCF7 cells

	H3K27ac	H3K27me3	IgG
ChIPseq	6	6	6
Cut&Tag	6	6	6
Cut&Run	6	6	6
ATACseq		6	

Pic one sample and load all 6 replicates of this sample into R. Look at the chromosomes. What do you observe? Remove unconventional chromosomes.

Be mindful of the genome build/annotation

same genome build from GRC (Genome Reference Consortium)
e.g. GRCh38 = hg38; GRCh37 = hg19

UCSC/NCBI ≠ ENSEMBL

1. Different patch integration may alter sequence, but not coordinates
2. Different chromosome nomenclature in ENSEMBL (>2016) and UCSC

   ENSEMBL: numbers only: *1,2* … (! GENCODE uses *chr1, chr2* …)
   UCSC: „chr“ + number: *chr1, chr2* ….

3. **Different gene annotations**

   ENSEMBL: https://www.ensembl.org/index.html
   - Automated by aligning experimentally-validated sequences (*Curwen V et al. Genome Res. 2004*) + Manual curation (HAVANA) = GENCODE
   - ENSEMBL identifiers: ENSG…, ENST…., ENSP…, ENSE…

   UCSC: https://genome.ucsc.edu/goldenpath/help/ftp.html
   - NCBI RefSeq predicted + manually curated (NCBI/UCSC RefSeq curated) (alignments are recently integrated from GENCODE)
   - NCBI identifiers; NM_..., NP_..., NR_..., XP_...

Which genome build/version is correct?

' Who knows? '

Tip: Use the most recent annotations and genome build at the start of the project and stick to it!

The GenomicRanges package

= representation of genomic locations in R

- Bioconductor package
- Install via: $ *BiocManager::install(``GenomicRanges``)*

minimal GRanges object:

peaks <- GRanges(
seqnames = chromosome*,*
*ranges = IRanges(*start, end, names*),*
strand = strand (+,-,*) *)*

Additional data as meta data:

peaks <- GRanges(
seqnames = chromosome*,*
*ranges = IRanges(*start, end, names*),*
strand = strand (+,-,*),*
FC = fold-change*))*

The GenomicRanges package

Genomic Locations **Metadata**

```
## GRanges object with 6 ranges and 3 metadata columns:
##                             seqnames              ranges strand |        FC
##                                <Rle>           <IRanges>  <Rle> | <numeric>
##      chr1_33338963_33339095     chr1   33338963-33339095      * |   5.19504
##    chr1_111489759_111489936     chr1 111489759-111489936      * |   8.41439
##    chr1_198776741_198776902     chr1 198776741-198776902      * |   6.12064
##     chr10_20003900_20004006    chr10   20003900-20004006      * |   4.75220
##     chr10_72248903_72249062    chr10   72248903-72249062      * |   5.19504
##   chr11_114154531_114154770    chr11 114154531-114154770      * |   9.48454
##                               qvalue     score
##                             <numeric> <integer>
##      chr1_33338963_33339095   2.29619        22
##    chr1_111489759_111489936   8.37238        83
##    chr1_198776741_198776902   4.81979        48
##     chr10_20003900_20004006   2.83584        28
##     chr10_72248903_72249062   2.29619        22
##   chr11_114154531_114154770  10.84952       108
##   -------
```

- Components of the GRanges objects can be accessed using seqnames(), names(), ranges(), strand() … accessor functions
- Genomic ranges extracted with granges()
- Metadata extracted with mcols()

The GenomicRanges package – Exercise (20 min)

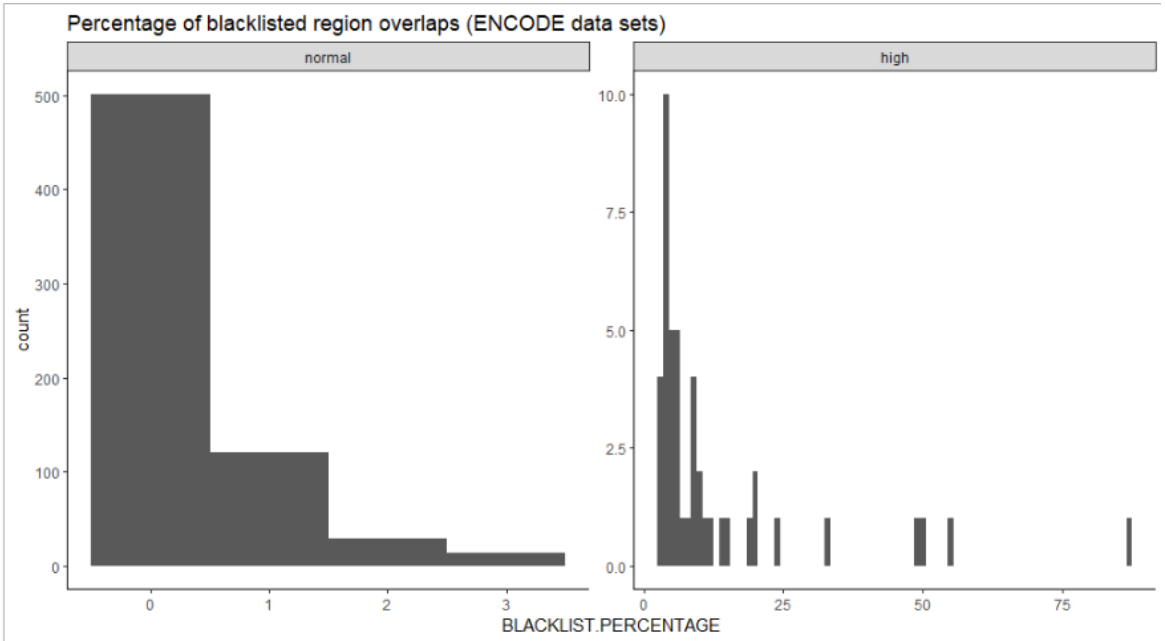
1. Pic one of your replicates and convert it into a GRange object.
2. How many peaks does your replicate contain?
3. Plot the distribution of peak sizes as a histogram.
4. Plot the p-value distribution of your peaks as histogram.

Advanced: Plot the peak size and p-value distribution of all replicates as a violin plot.

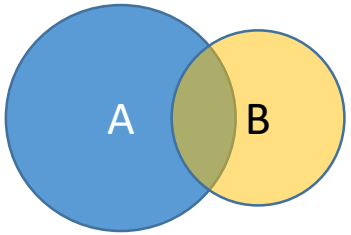
ENCODE Blacklisted Regions

= list of exclusive regions (ChIPseq signal artefacts, erroneous regions of genome assemblies, unannotated repeats)

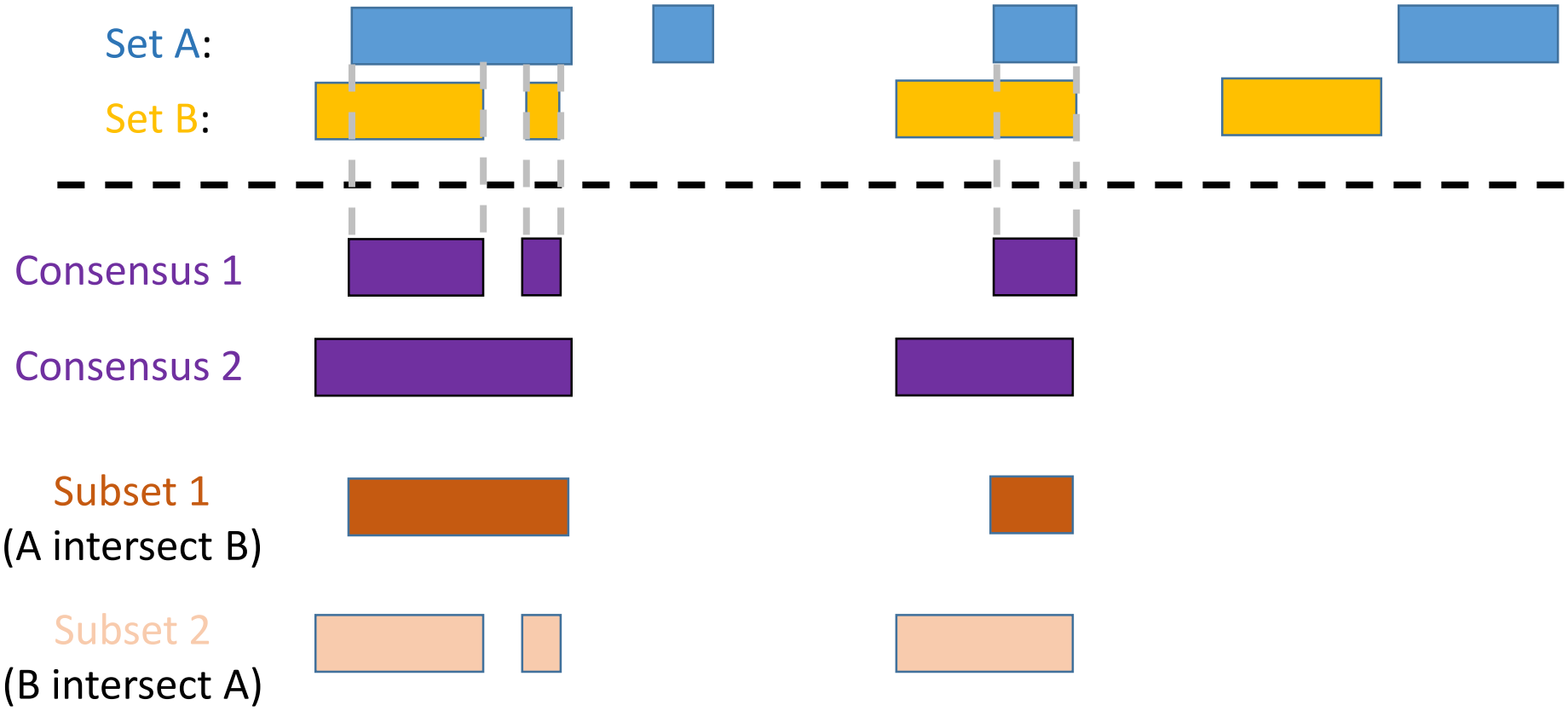
- Automatically generated from ENCODE control data (inputs, **multiple** cell types)
- see *Amemiya et al. Nature 2019*
- Blacklists on Github: https://github.com/Boyle-Lab/Blacklist/tree/master/lists
- ENCODE quality control parameter e.g. low quality samples >87% of reads BLRs
- Only valid for the "accessible" genome as input data is a proxy for "open" chromatin (d.h. DnaseSeq, FAIREseq, ATACseq, CHIPseq (+similar) …)



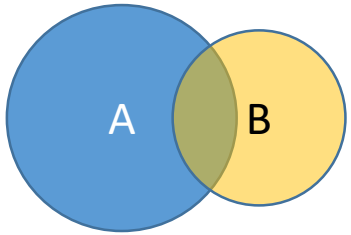
Subsets/Intersections and unions



1. **Consensus** peak set and **subsets**:



Subsets/Intersections and unions



1+2. **Consensus** peak set and **subsets**:

Set A:

Set B:

Consensus 1

for replicates, motifs

Consensus 2

for quantification

Subset 1
(A intersect B)

if B is blacklisted,
if A is of primary interest

Subset 2
(B intersect A)

if A is blacklisted,
if B is of primary interest

Subsets/Intersections and unions

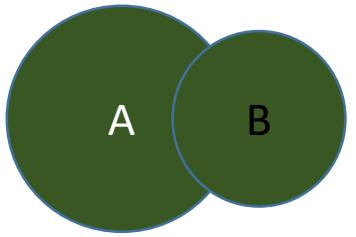
Functions:

$ *findOverlapPairs(A,B)* # GRange of overlapping GRanges paired, BEDPE
$ *intersect(A,B)* # GRanges of overlapping regions (Consensus type 1)
$ *reduce(A,B)* # merges overlapping peaks (Consensus type 2)
$ findOverlaps(A,B) # Hits object of overlapping GRanges
$ *countOverlaps(A,B)* # Vector of number of Peaks in A overlapping any sequence in B
$ *subsetByOverlaps(A,B)* #subset of A overlapping B (!sensitive to orientation)

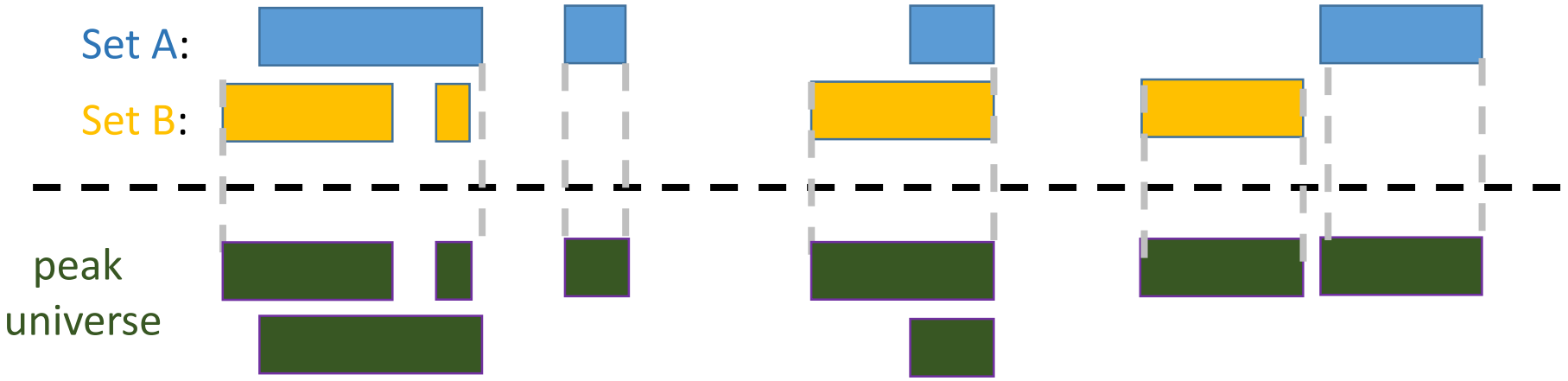
Arguments:

1. ignore.starnd TRUE for unstranded data like ChIP
2. minoverlap number of overlapping bases required
3. maxgap maximum number of bases between two ranges
4. type {start,end,equal,within,default:any}
5. select findOverlaps, {first,last,arbitrary,default:all}
6. invert {TRUE,FALSE} – keep non-(TRUE) or overlapping (FALSE) ranges

Subsets/Intersections and unions

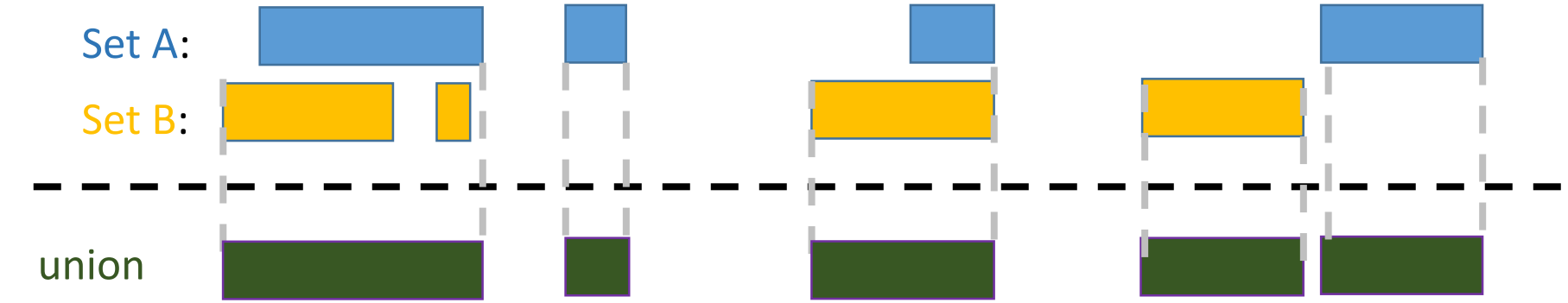


3. **Union** Type 1: merging peak region lists:



$ *append(A,B)*

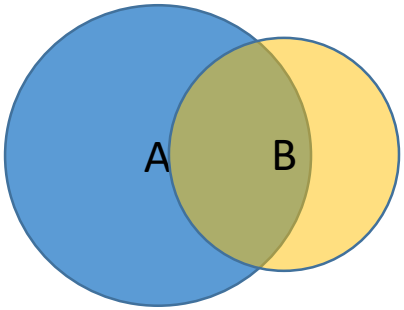
4. **Union** Type 2: merging peak regions:



$ *reduce(append(A,B))*

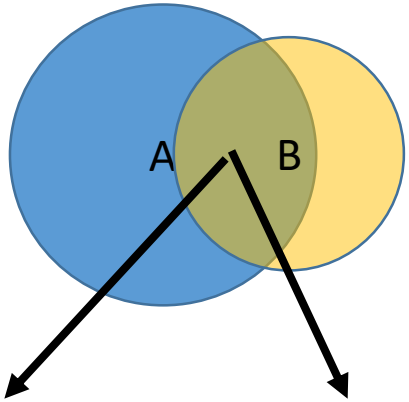
One word on reproducible peaks

1. Definition: consensus peak set



- A and B might be replicates
- The consensus peak set is the intersect of A and B
- Does not allow for statistical evaluation
- **Use case:** differential binding analysis (adds statistics)

2. Definition: irreproducible discovery rate (IDR)

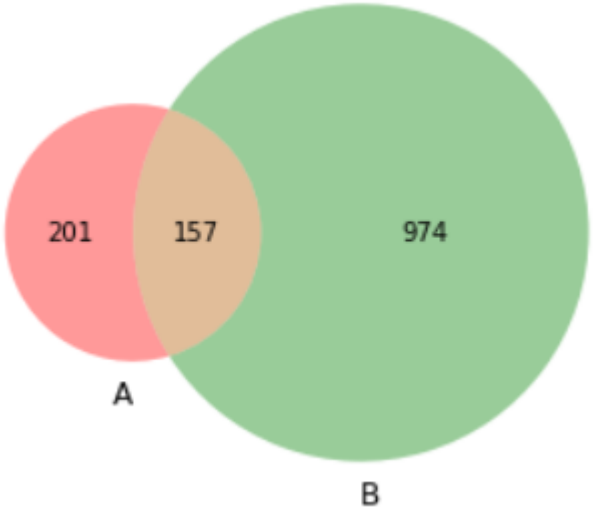


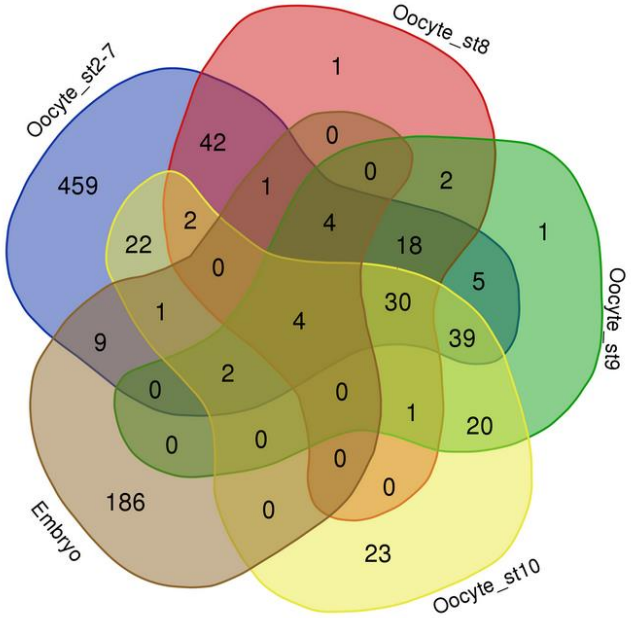
- Filtered upon reproducibility of peak calling
- Only pair-wise comparisons (standard: reproduce in 2 of n)
- Highl variable depending on background noise and peak calling paramters (use low P or FDR thresholds in peak calling)
- **Use case**: motif discovery of high confidence peaks

Visualizing intersects and subsets

1. Venn diagram

- 2-3 data sets
- >2 sets Euler diagrams are possible but they only approximate proportionality
- Switching from circles to ellipses allows to maintain proportionality in >3 data sets, but is hard to read

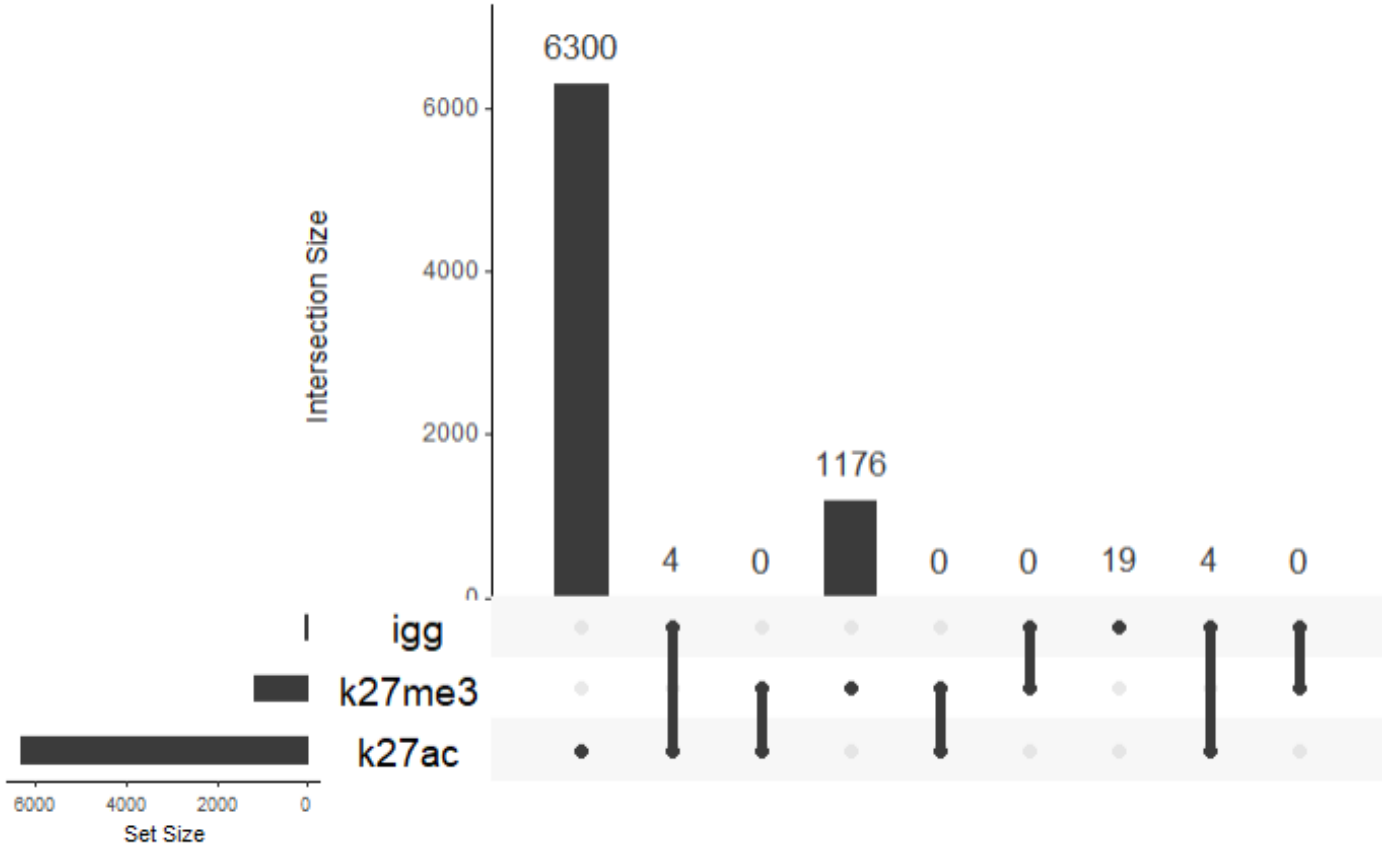




Visualizing intersects and subsets

2. Upset plot

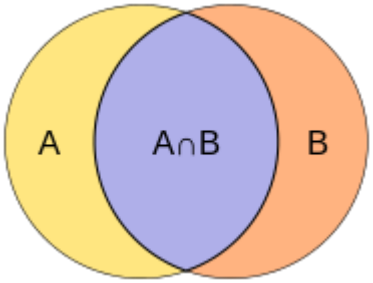
- >3 data sets, pair-wise comparison
- developed by Niels Gehlenborg and Jake Conway



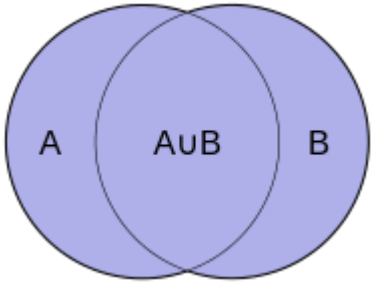
Visualizing intersects and subsets

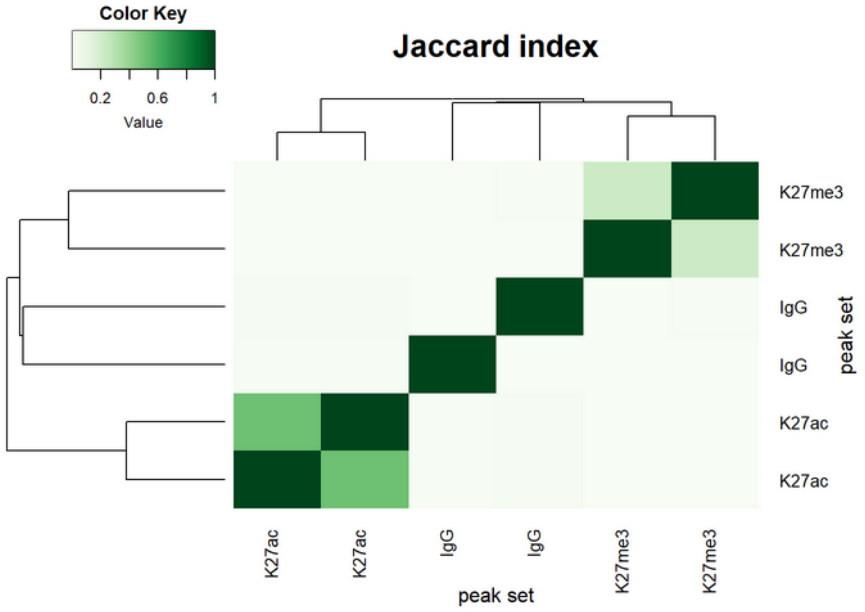
3. Jaccard index and Heatmaps

- >3 data sets, pair-wise comparison of asymmetric sets
- accounts for differences in set size by normalising to set union



$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$





Peak overlaps and subsets – Exercise (2h)

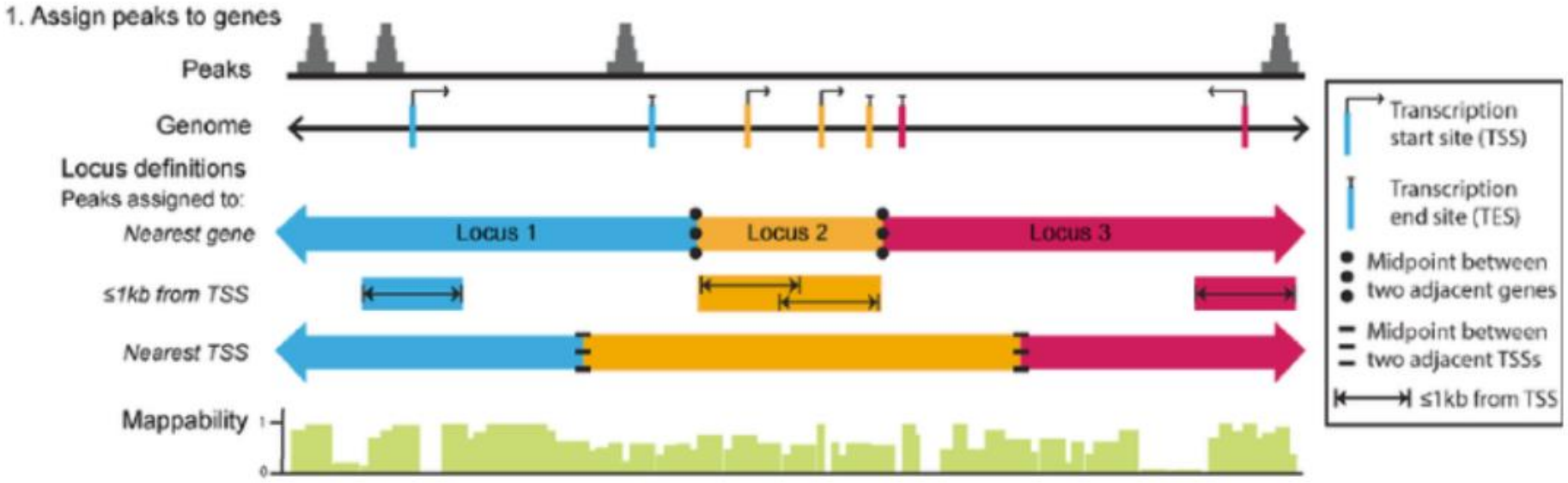
1. Load all replicates for one assay and antibody into R and convert the genomic positions into a GRangesList.
2. Compute the Jaccard index and compare the overlap of your replicates. Are there any outliers?
3. Visualise the overlaps of all replicates as an upset plot.
4. Generate a consensus type 1 peak set including peaks identified in all replicates.
5. Generate a consensus type 1 peak set including all peaks identified in at least 2 replicates.
6. Generate an IDR peak set containing all peaks identified in at least 2 replicates.
7. Share the consensus type 1 peak sets (at least 2 replicates) among each other and generate an upset plot comparing Cut&Tag with Cut&Run with ChIPseq per antibody.

Optional: Compare ATACseq with H3K27ac and H3K27me3 ChIPseq/Cut&Run&Tag using upsetR

Peak annotation

Purpose:

1. Annotate peak region to a gene
   - conventions: the closest gene in the linear genome (by TSS)



$ *annotatePeak()* #package ChIPseeker

Peak annotation

Purpose:

1. Annotate to a gene
    - conventions: the closest gene in the linear genome (by TSS)
    - adaptations:
        1. the closest expressed gene (by TSS)
        2. Correlation of binding and changes in gene expression (BETA, *Wang et al. 2013 Nature prot*.)
        3. the closest gene within a topological domain
        4. to the n closest genes, filtered by genomic distance in 2D or 3D (gene-peak mode)
        5. The n closest peaks to a gene (peak-gene mode, FindIT2)
        6. By gene ontologies (GREAT, *McLean Nature Biotechn. 2010*)

Ressources for annotation data

1. AnnotationHub (by Marc Carlson)

= Bioconductor package that provides a unified interface of retrieving genomic information from the web

Aim: unite Bioconductor annotation resources

- caches annotation information and updates on demand (version control, faster access, offline work)
- uses SQLite backend

2. Build-in annotation ressources

- UCSC: *TxDb.Hsapiens.UCSC.hg19.knownGene, : TxDb.Mmusculus.UCSC.mm10.knownGene*, …
  General: UCSC.genomeBuild.knownGene
- ENSEMBL: *EnsDb.Hsapiens.v75, EnsDb.Mmusculus.v79*,…
  General: EnsDb.Species.GenomeVersion

Ressources for annotation data

3. Own annotation files or downloads (GFF3, GTF)

- Contain gene annotation information for specific genome build
- GFF3*: „general feature format“;* 9 column, tab-delimited text file, one line per feature
  - column 1: „seqid“ → matches corresponding .fasta file
  - column 2: „source“ → Data source
  - column 3: „type“ → type of feature e.g. CDS, exon…
  - column 4: „start“ → 1-based
  - column 5: „end“ → 1-based
  - column 6: „score“ → feature score (E-value; sequence similarity features, P-values for gene predictions
  - column 7: „strand“ → {+,-,.}
  - column 8: „phase“ → {0,1,2} indicates start of next codon for feature CDS
  - column 9: „attributes“ → list of feature attributes as „tag=value“ pairs separated by ;

```
0  ##gff-version 3.1.26
1  ##sequence-region ctg123 1 1497228
2  ctg123 . gene            1000  9000  .  +  .  ID=gene00001;Name=EDEN
3  ctg123 . TF_binding_site 1000  1012  .  +  .  ID=tfbs00001;Parent=gene00001
4  ctg123 . mRNA            1050  9000  .  +  .  ID=mRNA00001;Parent=gene00001;Name=EDEN.1
```

Ressources for annotation data

3. Own annotation files or downloads (GFF3, GTF)

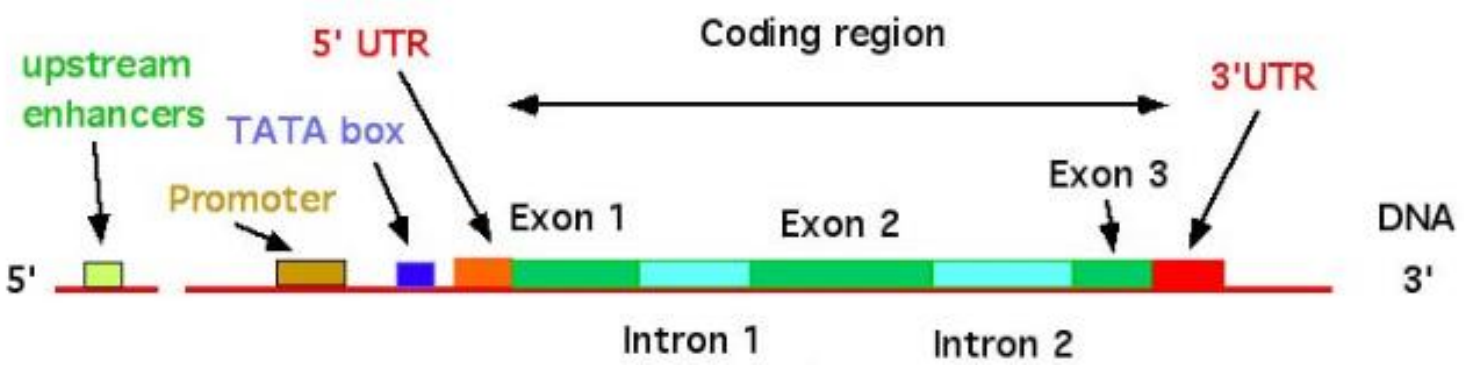
- GTF: „*general transfer format*“; 9 column, tab-delimeed text file, one line per feature
  - column 1: „seqname“ → name of chromosome/scaffold
  - column 2: „source“ → Data source
  - column 3: „feature“ → type of feature e.g. gene, variation, Similarity
  - column 4: „start“ → 1-based
  - column 5: „end“ → 1-based
  - column 6: „score“ →
  - column 7: „strand“ → {+,-,.}
  - column 8: „frame“ → {0,1,2} indicates start of next codon for feature CDS
  - column 9: „attributes“ → list of feature attributes as „tag value“ pairs separated by ;

```
1 transcribed_unprocessed_pseudogene  gene        11869 14409 . + . gene_id "ENSG00000223972"; gene_name "DDX11L1"; gene_source
1 processed_transcript                transcript  11869 14409 . + . gene_id "ENSG00000223972"; transcript_id "ENST00000456328";
```

Peak annotation

Purpose:

2. Annotate with regards to genomic features



ChIPpeakAnno: *annotatePeakinBatch()*

ChIPSeeker: *annotatePeak()*

Peak annotation – Exercise (45 min)

1. Pic a consensus peak set per antibody (IgG, H3K27ac, H3K27me3) for one of the assays and the ATACseq experiment. Annotate the peaks to the closest gene. Pic an annotation resource of your choice.
2. Compare the genomic distribution of consensus peaks between IgG, H3K27ac, H3K27me3 and ATACseq using the *plotAnnoBar()* function of the ChIPSeeker package. What do you observe?

Functional annotation of peak associated genes

Aim: assign biological meaning

Over-representation analysis (ORA)

- Comparison to a predefined set of genes
- **Aim:** Determine if a *priori* defined set of genes is enriched (over-represented) in a subset of „interesting“ genes more than expected by chance
- P-values computet using the Binomial distribution, **Hypergeometric distribution**, the Fisher exact test, the Chi-square test …
- Control for multiple testing afterwards

$$p = 1 - \sum_{i=0}^{k-1} \frac{\binom{M}{i}\binom{N-M}{n-i}}{\binom{N}{n}}$$

=> P-value as probability of a set of genes represented in a reference set by chance

n – number of „interesting“ genes
N – number of background genes
M – number of genes in annotated gene set S
k – number of „interesting“ genes annotated in S

Functional annotation of peak associated genes

Pathways:

= series of interactions among molecules (e.g. gene products)

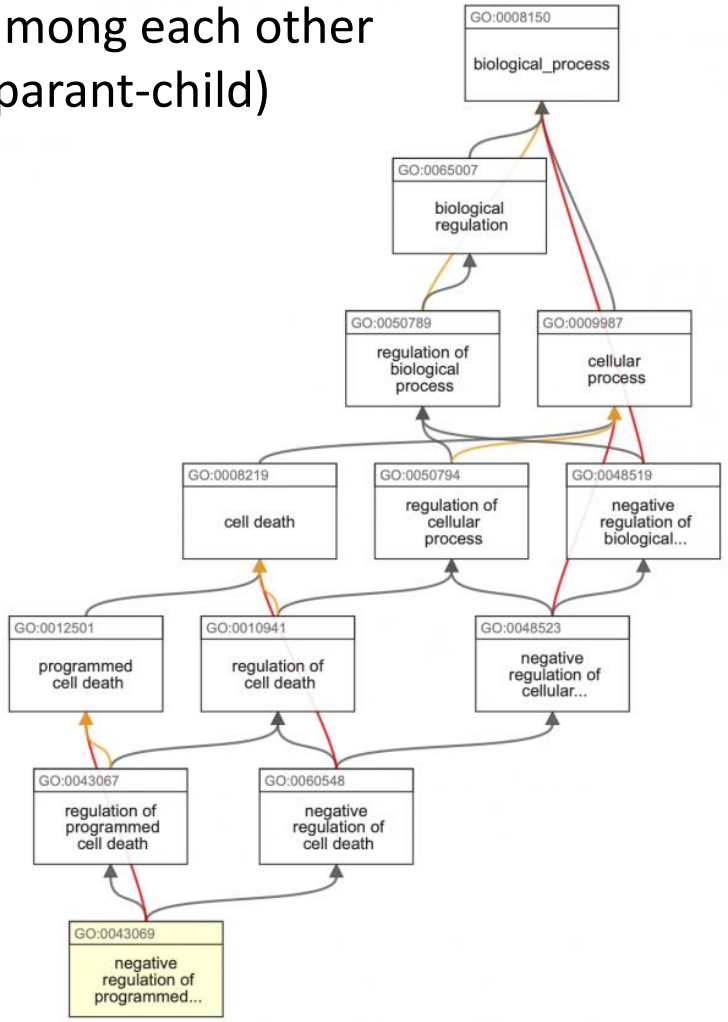
- Databases:
  - KEGG - Kyoto Encyclopedia of Genes and Genomes (https://www.genome.jp/kegg/pathway.html)
    - Manually drawn pathway maps
    - Categories: metabolism, genetic information processing, environmental information processing, cellular processes, human disease, organismal systems, drug development
  - RA – Reactome (https://reactome.org/)
  - Wikipathways (https://www.wikipathways.org/index.php/WikiPathways)
  - MSigDB – Molecular Signature Database (https://www.gsea-msigdb.org/gsea/msigdb/)
  - Pathway Commons (https://www.pathwaycommons.org/)

Functional annotation of peak associated genes

Gene Ontology (GO): http://geneontology.org/

= curated classes of gene function with relations among each other

- Directed acyclic graphs: edges have direction (parant-child)
- No cycles with in ontologies
- 3 aspects of classification:
  - MF: molecular function
  - CC: cellular component
  - BP: biological process



Fucntional enrichment – Exercise (30 min)

1. Perform GO enrichment analysis for biological processes on H3K27ac peaks overlapping accessible regions (from ATACseq). Visualise the top10 most significantly enriched ontologies.
2. Perform KEGG pathway over-representation analysis on the same peak subset as in 1, visualise the top10 pathways. Compare your results to the GO enrichment in 1.

Alternative tools to analyse peaks

1. **Bedtools** (https://bedtools.readthedocs.io/en/latest/)

= „swiss-army knife of tools for a wide-range of genomics analysis tasks”

- command line tools
- *intersect*, *merge, genomecov, shuffle, count …*

2. HOMER (homer.ucsd.edu/homer/index.html)

= Hypergeometric Optimization of Motif EnRichment

- Command line tools written in Perl and C++
- Peak annotation, heatmaps, motif discovery, coverage plots …

Further Reading

- MACS peak calling: https://hbctraining.github.io/Intro-to-ChIPseq/lessons/05_peak_calling_macs.html
- Genome annotation:

  ENSEMBL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4919035/

  RefSeq: https://www.ncbi.nlm.nih.gov/books/NBK21091/
- Differences between ENSEMBL and UCSC/RefSeq gene annotation: https://genome.ucsc.edu/FAQ/FAQgenes.html
- GenomicRanges: https://bioconductor.org/packages/devel/bioc/vignettes/GenomicRanges/inst/doc/GenomicRangesIntroduction.html
- UpSetR: https://jokergoo.github.io/ComplexHeatmap-reference/book/upset-plot.html
- Peak annotation: GREAT (doi: 10.1038/nbt.1630); BETA (10.1038/nprot.2013.150)
- Functional analysis of gene lists: Huang et al. NAR 2009 (https://doi.org/10.1093/nar/gkn923)