Week 3

- Unix Problem Set Q&A
- Review ChIP-Seq and RNA-Seq pipelines

- Loading files to the cluster using FileZilla and rsync

- On the ghpcc06 cluster, read criteria for running bowtie2, picard, samtools, etc.
- See sample bsub commands for running each module
- Look at bowtie2 output files
- Use IGV to view .bam and .bed files

Tasks:
Create lab schedule for using the ghpcc06 cluster
Explore the cluster and examine modules and tools for data analysis that are available
Upload Fastq files to ghpcc06 cluster

Week 4

- On the ghpcc06 cluster, read criteria for running macs, rsem-calculate-expression, etc.
- running rsem-calculate-expression on the cluster
  https://bioinfo.umassmed.edu/index.php?p=33
- Look at macs2 and RSEM output files
- RNA-Seq – Look at RSEM output files
- See sample bsub commands for rsem-calculate-expression

Tasks:
Create lab schedule for using the ghpcc06 cluster
Explore the cluster and examine modules and tools for data analysis that are available
Upload Fastq files to ghpcc06 cluster

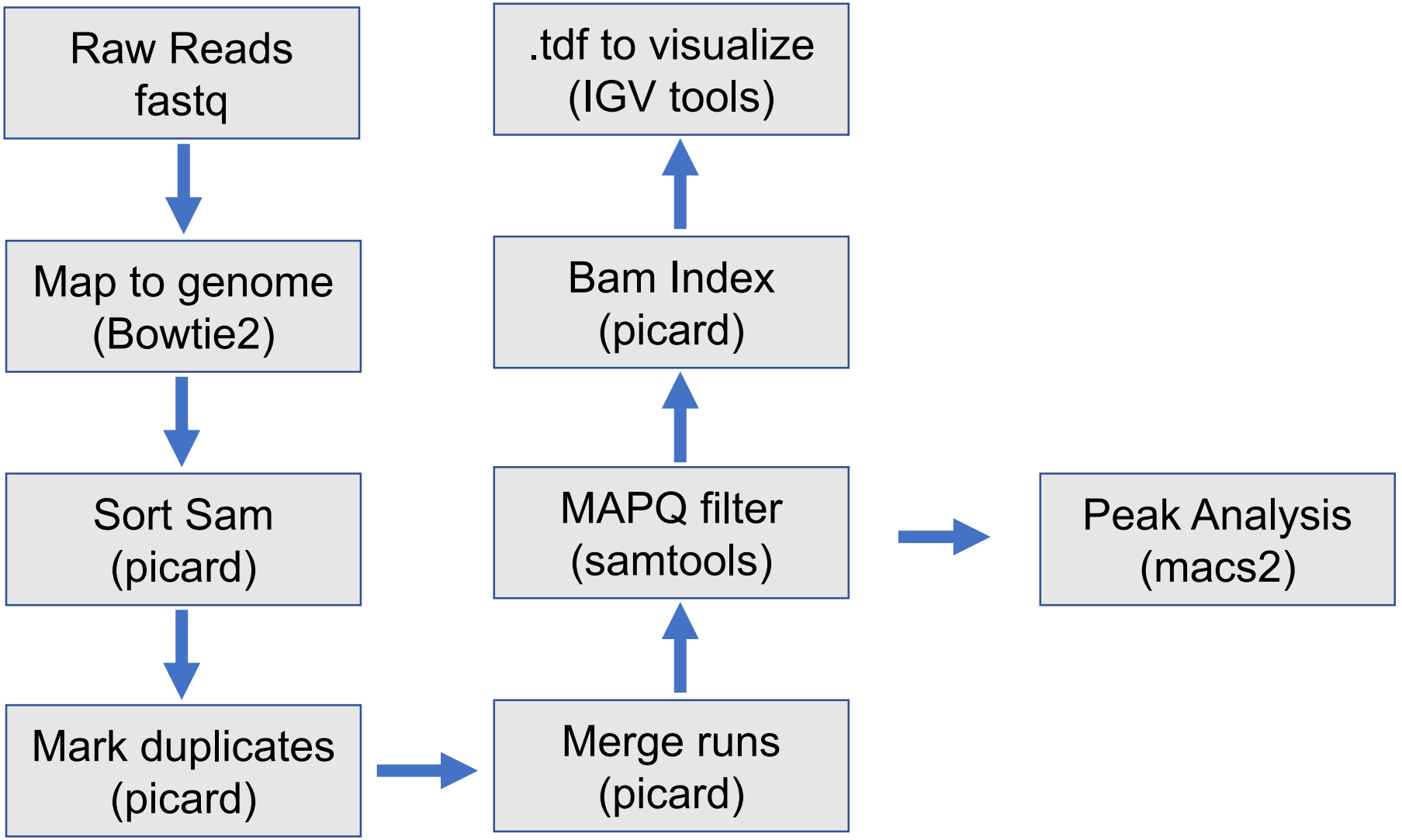
Week 5

- Overview of questions so far
- Introducing bedtools: *a powerful toolset for genome arithmetic*
  https://bedtools.readthedocs.io/en/latest/
- Running bedtools on the ghpcc06 cluster for comparing .bed files of different experiments

- Introducing NGSplot: Quick mining and visualization of NGS data by integrating genomic databases
  https://github.com/shenlab-sinai/ngsplot
- Running NGSplot on the ghpcc06 cluster for making ChIP-Seq and RNA-Seq heatmaps and extracting tag density files

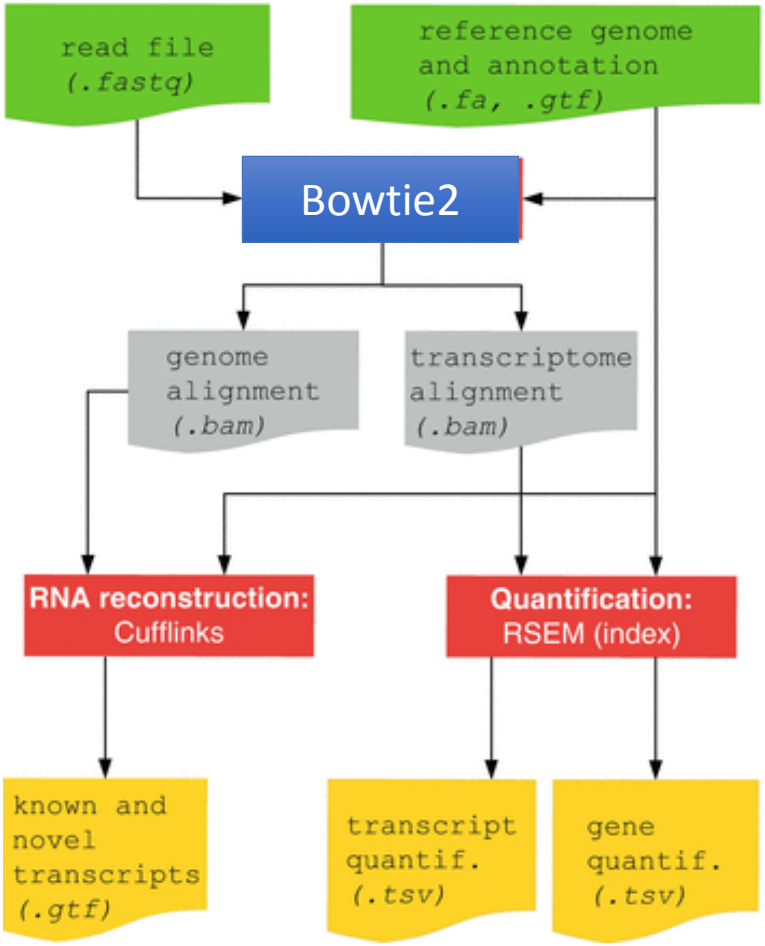
- Introducing HOMER: Software for motif discovery and next-gen sequencing analysis
  http://homer.ucsd.edu/homer/ngs/
- Running HOMER on the ghpcc06 cluster for annotating bed files and performing motif analysis

My ChIP-Sequencing Workflow



RNA-Seq Pipline

Methods and Tools



Bowtie and Bowtie2 use Burrows-Wheeler indexing for aligning reads. With bowtie2 there is no upper limit on the read length

Bowtie2 to align reads in a splice-aware manner and aids the discovery of new splice junctions

RSEM is a software package for estimating gene and isoform expression levels from RNA-Seq data. The RSEM package provides an user-friendly interface, supports threads for parallel computation of the EM algorithm, single-end and paired-end read data, quality scores, variable-length reads and RSPD estimation.

Bedtools

Bedtools is a fast, flexible toolset for genome arithmetic.

bedtools: *a powerful toolset for genome arithmetic*

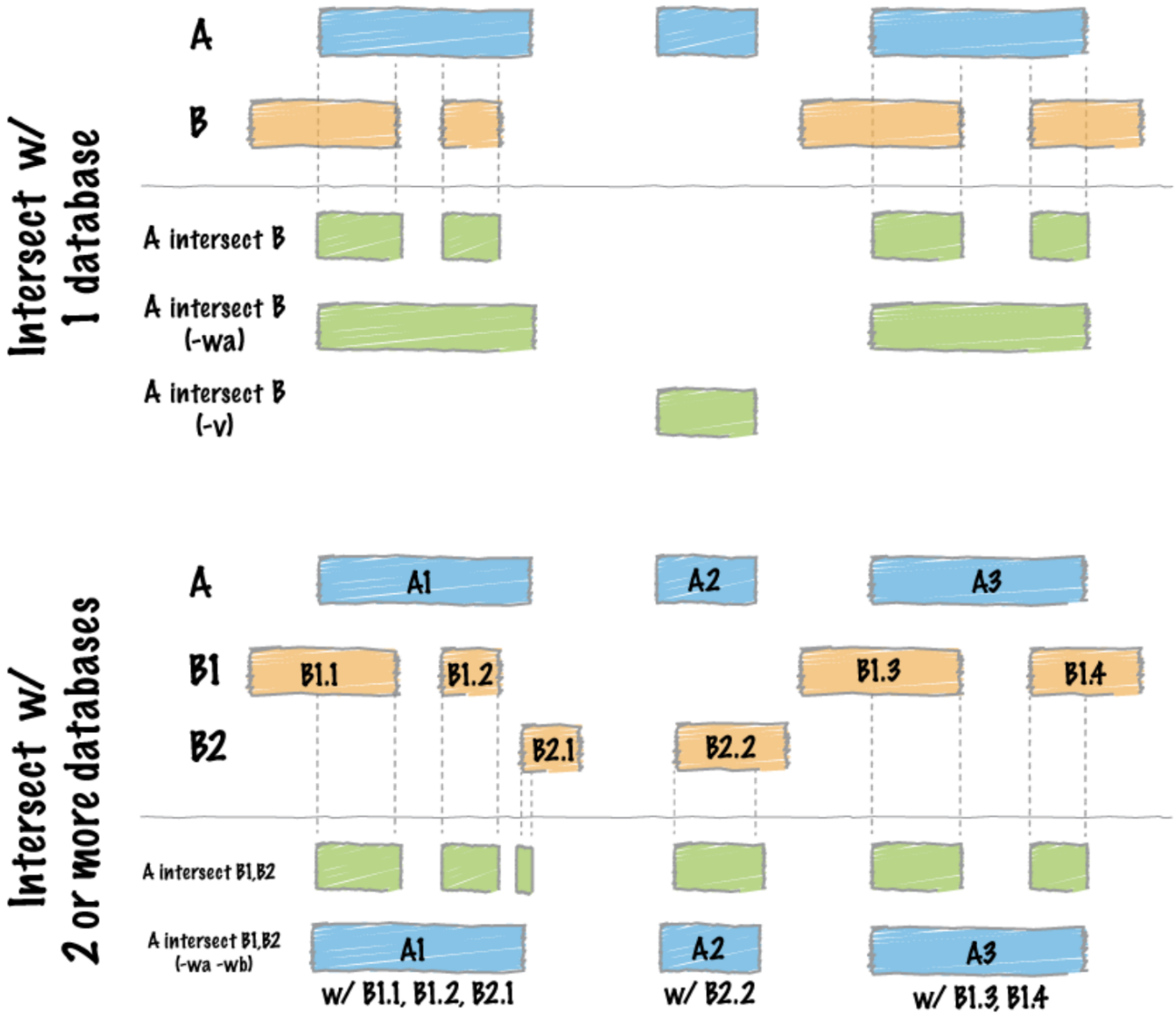
Collectively, the **bedtools** utilities are a swiss-army knife of tools for a wide-range of genomics analysis tasks. The most widely-used tools enable *genome arithmetic*: that is, set theory on the genome. For example, **bedtools** allows one to *intersect*, *merge*, *count*, *complement*, and *shuffle* genomic intervals from multiple files in widely-used genomic file formats such as BAM, BED, GFF/GTF, VCF. While each individual tool is designed to do a relatively simple task (e.g., *intersect* two interval files), quite sophisticated analyses can be conducted by combining multiple bedtools operations on the UNIX command line.

bedtools is developed in the Quinlan laboratory at the University of Utah and benefits from fantastic contributions made by scientists worldwide.

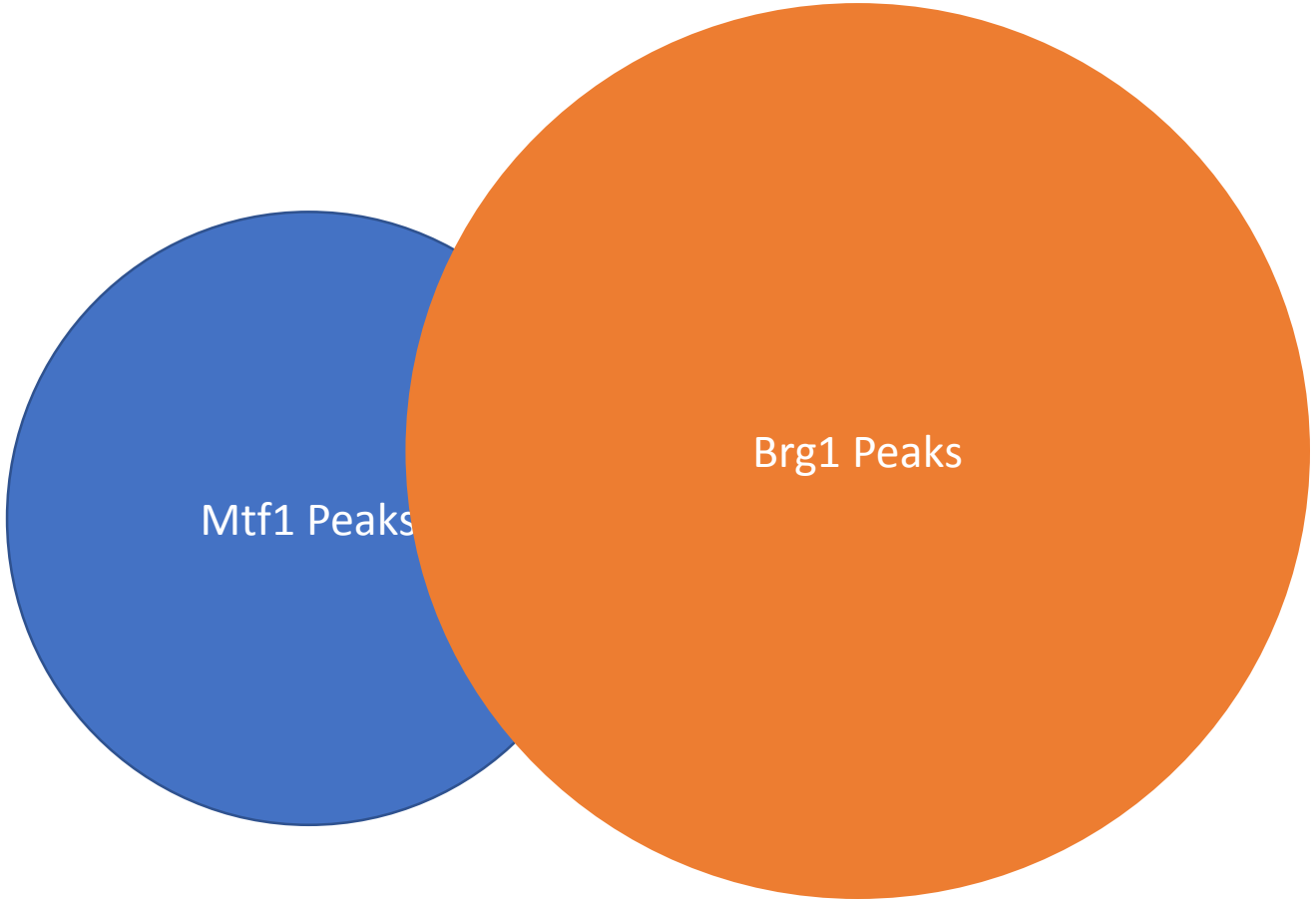
https://bedtools.readthedocs.io/en/latest/

bedtools “intersect”

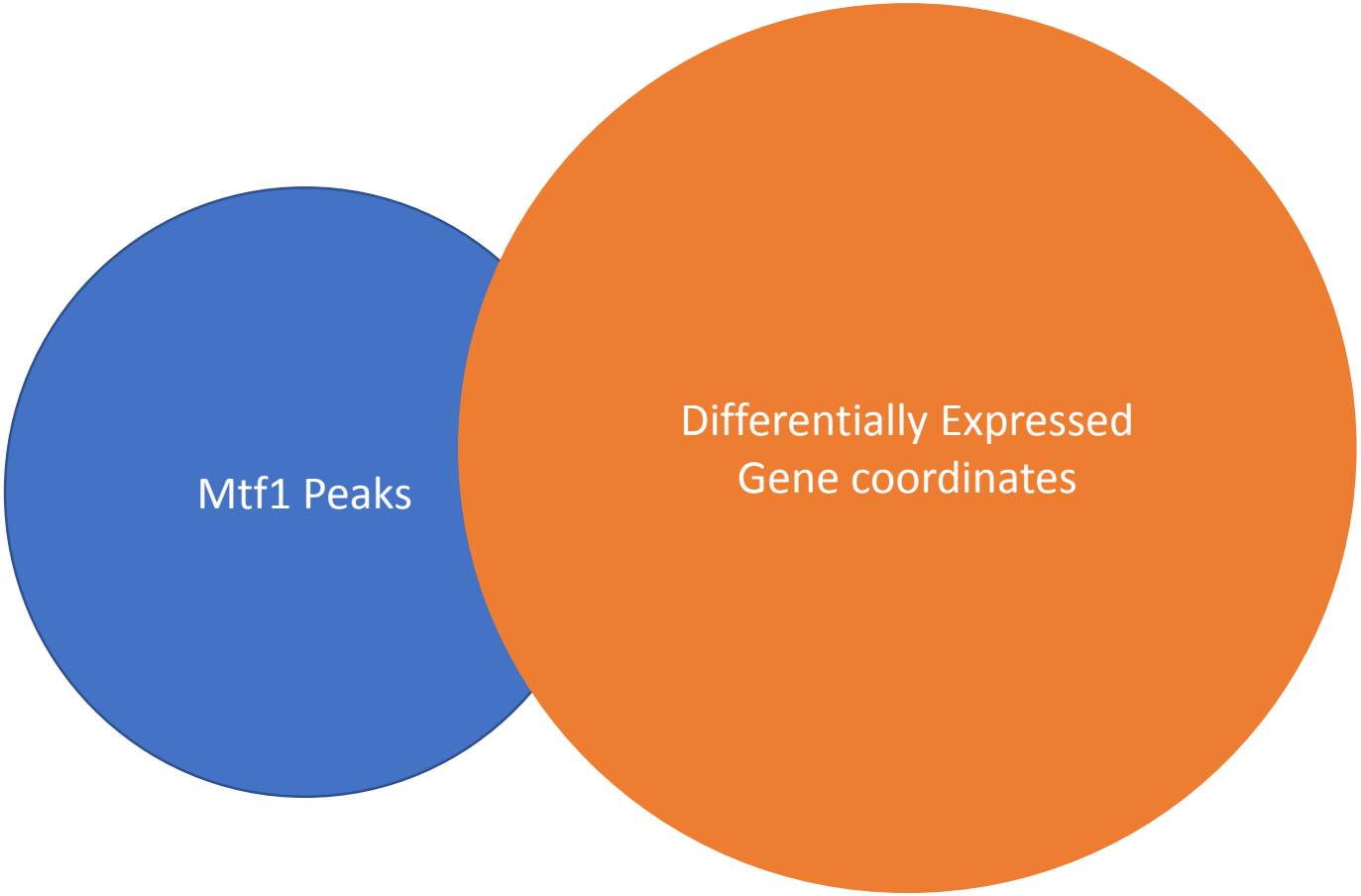
The `intersect` command is the workhorse of the `bedtools` suite. It compares two or more BED/BAM/VCF/GFF files and identifies all the regions in the genome where the features in the two files overlap (that is, share at least one base pair in common).



ChIP-Seq comparision



ChIP-Seq and RNA-Seq comparison



```
[ss45w@ghpcc06 ~]$ module load bedtools/2.27.1
bedtools 2.27.1 is located under /share/pkg/bedtools/2.27.1

[ss45w@ghpcc06 ~]$ bedtools intersect

Tool:    bedtools intersect (aka intersectBed)
Version: v2.27.1
Summary: Report overlaps between two feature files.

Usage:   bedtools intersect [OPTIONS] -a <bed/gff/vcf/bam> -b <bed/gff/vcf/bam>

        Note: -b may be followed with multiple databases and/or
        wildcard (*) character(s).
Options:
        -wa     Write the original entry in A for each overlap.

        -wb     Write the original entry in B for each overlap.
                - Useful for knowing _what_ A overlaps. Restricted by -f and -r.

        -loj    Perform a "left outer join". That is, for each feature in A
                report each overlap with B.  If no overlaps are found,
                report a NULL feature for B.

        -wo     Write the original A and B entries plus the number of base
                pairs of overlap between the two features.
                - Overlaps restricted by -f and -r.
```

Default behavior

By default, `intersect` reports the intervals that represent overlaps between your two files. To demonstrate, let's identify all of the CpG islands that overlap exons.

```
bedtools intersect -a cpg.bed -b exons.bed
chr1    29320   29370   CpG:_116
chr1    135124  135563  CpG:_30
chr1    327790  328229  CpG:_29
chr1    327790  328229  CpG:_29
chr1    327790  328229  CpG:_29
```

bedtools intersect -a D0Prmt5_n1n2_q0.05_PE_peaks.narrowPeak.bed -b D0_H3K27ac_q0.1_peaks.narrowPeak.bed > D0_3T3L1_PE0.05Prmt5_D0_H3K27ac_overlap.txt

Reporting the original feature in each file.

The `-wa` (write A) and `-wb` (write B) options allow one to see the original records from the A and B files that overlapped. As such, instead of not only showing you *where* the intersections occurred, it shows you *what* intersected.

```
bedtools intersect -a cpg.bed -b exons.bed -wa -wb \
| head -5
chr1    28735   29810   CpG:_116    chr1    29320   29370   NR_024540_exon_10_0_chr1_29321_r        -
chr1    135124  135563  CpG:_30 chr1    134772  139696  NR_039983_exon_0_0_chr1_134773_r    0   -
chr1    327790  328229  CpG:_29 chr1    324438  328581  NR_028322_exon_2_0_chr1_324439_f    0   +
chr1    327790  328229  CpG:_29 chr1    324438  328581  NR_028325_exon_2_0_chr1_324439_f    0   +
chr1    327790  328229  CpG:_29 chr1    327035  328581  NR_028327_exon_3_0_chr1_327036_f    0   +
```

Find features that DO NOT overlap

Often we want to identify those features in our A file that **do not** overlap features in the B file. The `-v` option is your friend in this case.

```
bedtools intersect -a cpg.bed -b exons.bed -v \
| head
chr1    437151  438164  CpG:_84
chr1    449273  450544  CpG:_99
chr1    533219  534114  CpG:_94
chr1    544738  546649  CpG:_171
chr1    801975  802338  CpG:_24
chr1    805198  805628  CpG:_50
chr1    839694  840619  CpG:_83
chr1    844299  845883  CpG:_153
chr1    912869  913153  CpG:_28
chr1    919726  919927  CpG:_15
```

Week 5

- Overview of questions so far
- Introducing bedtools: *a powerful toolset for genome arithmetic*
  https://bedtools.readthedocs.io/en/latest/
- Running bedtools on the ghpcc06 cluster for comparing .bed files of different experiments

- Introducing NGSplot: Quick mining and visualization of NGS data by integrating genomic databases
  https://github.com/shenlab-sinai/ngsplot
- Running NGSplot on the ghpcc06 cluster for making ChIP-Seq and RNA-Seq heatmaps and extracting tag density files

- Introducing HOMER: Software for motif discovery and next-gen sequencing analysis
  http://homer.ucsd.edu/homer/ngs/
- Running HOMER on the ghpcc06 cluster for annotating bed files and performing motif analysis

HOMER (v4.11, 10-24-2019)

Software for motif discovery and next generation sequencing analysis

---

HOMER (Hypergeometric Optimization of Motif EnRichment) is a suite of tools for Motif Discovery and next-gen sequencing analysis. It is a collection of command line programs for UNIX-style operating systems written in Perl and C++. HOMER was primarily written as a *de novo* motif discovery algorithm and is well suited for finding 8-20 bp motifs in large scale genomics data. HOMER contains many useful tools for analyzing ChIP-Seq, GRO-Seq, RNA-Seq, DNase-Seq, Hi-C and numerous other types of functional genomics sequencing data sets.

```
[ss45w@ghpcc06 ~]$ module load HOMER/4.6
perl 5.18.1 is located under /share/pkg/perl/5.18.1
HOMER 4.6 is located under /share/pkg/HOMER/4.6
```

```
[ss45w@ghpcc06 ~]$ ls /share/pkg/HOMER/4.6
0.212316585079865.tmp       COPYING                     list2
bin                         cpp                         motifs
config.txt                  data                        README.txt
configureHomer.pl           DoughnutDocumentation.pdf   update
configureHomer.pl.20160804  list1                       update.txt
[ss45w@ghpcc06 ~]$ ls /share/pkg/HOMER/4.6/data
accession  genomes  GO  knownTFs  promoters
[ss45w@ghpcc06 ~]$ ls /share/pkg/HOMER/4.6/data/genomes/
hg19  mm10
```

```
[ss45w@ghpcc06 ~]$ ls /share/pkg/HOMER/4.6/bin/
addDataHeader.pl                  fasta2tab.pl                  loadPromoters.pl
addData.pl                        fastq2fasta.pl                makeBigBedMotifTrack.pl
addGeneAnnotation.pl              filterListBy.pl               makeBigWig.pl
addInternalData.pl                findGO.pl                     makeBinaryFile.pl
addOligos.pl                      findGOtxt.pl                  makeHiCWashUfile.pl
adjustPeakFile.pl                 findHiCCompartments.pl        makeMultiWigHub.pl
adjustRedunGroupFile.pl           findHiCDomains.pl             makeTagDirectory
analyzeChIP-Seq.pl                findHiCInteractionsByChr.pl   makeUCSCfile
analyzeHiC                        findKnownMotifs.pl            map-bowtie2.pl
analyzeRepeats.pl                 findMotifsGenome.pl           map-star.pl
analyzeRNA.pl                     findMotifs.pl                 mergeData.pl
annotateInteractions.pl           findPeaks                     mergePeaks
annotatePeaks.pl                  findRedundantBLAT.pl          motif2Jaspar.pl
annotateRelativePosition.pl       findTopMotifs.pl              motif2Logo.pl
annotateTranscripts.pl            freq2group.pl                 old
assignGeneWeights.pl              genericConvertIDs.pl          parseGTF.pl
assignGenomeAnnotation            genomeOntology                pos2bed.pl
assignTSStoGene.pl                GenomeOntology.pl             preparseGenome.pl
batchAnnotatePeaksHistogram.pl    getChrLengths.pl              prepForR.pl
batchFindMotifsGenome.pl          getConservedRegions.pl        profile2seq.pl
batchFindMotifs.pl                getDifferentialBedGraph.pl    qseq2fastq.pl
batchMakeTagDirectory.pl          getDifferentialPeaks          randomizeGroupFile.pl
batchParallel.pl                  getDiffExpression.pl          randomizeMotifs.pl
bed2pos.pl                        getDistalPeaks.pl             randRemoveBackground.pl
bed2tag.pl                        getFocalPeaks.pl              removeAccVersion.pl
changeNewLine.pl                  getGenesInCategory.pl         removeBadSeq.pl
checkPeakFile.pl                  getGenomeTilingPeaks          removeOutOfBoundsReads.pl
checkTagBias.pl                   getGWASoverlap.pl             removePoorSeq.pl
chopify.pl                        getHiCcorrDiff.pl             removeRedundantPeaks.pl
chopUpBackground.pl               getMappableRegions            renamePeaks.pl
chopUpPeakFile.pl                 getPartOfPromoter.pl          resizePosFile.pl
cleanUpPeakFile.pl                getPeakTags                   revoppMotif.pl
cleanUpSequences.pl               getPos.pl                     runHiCpca.pl
cluster2bedgraph.pl               getRandomReads.pl             scanMotifGenomeWide.pl
cluster2bed.pl                    getSiteConservation.pl        scrambleFasta.pl
compareMotifs.pl                  getTopPeaks.pl                seq2profile.pl
condenseBedGraph.pl               gff2pos.pl                    SIMA.pl
cons2fasta.pl                     go2cytoscape.pl               Statistics.pm
conservationAverage.pl            groupSequences.pl             tab2fasta.pl
conservationPerLocus.pl           homer                         tag2bed.pl
convertCoordinates.pl             homer2                        tag2pos.pl
convertIDs.pl                     HomerConfig.pm                tagDir2bed.pl
convertOrganismID.pl              homerTools                    tagDir2HiCsummary.pl
duplicateCol.pl                   joinFiles.pl                  zipHomerResults.pl
eland2tags.pl                     loadGenome.pl
```

HOMER for annotating .bed files

http://homer.ucsd.edu/homer/introduction/programs.html

HOMER

Software for motif discovery and ChIP-Seq analysis

HOMER Program Index

Below is a quick introduction to the different programs included in HOMER. Running each program without any arguments will provide basic instructions and a list of command line options.

FASTA file Motif Discovery

findMotifs.pl - performs motif analysis with lists of Gene Identifiers or FASTA files (See FASTA file analysis)

homer2 - core component of motif finding (Called by everything else , See FASTA file analysis)

Gene/Promoter-based Analysis

findMotifs.pl - performs motif and gene ontology analysis with lists of Gene Identifiers, both promoter and mRNA motifs (See Gene ID Analysis Tutorial)

findGO.pl - performs only gene ontology analysis with lists of Gene Identifiers (Called by findMotifs.pl, See Gene Ontology Analysis)
loadPromoters.pl - setup custom promoter sets for specialized analysis (See Customization)

Next-Gen Sequencing/Genomic Position Analysis

findMotifsGenome.pl - performs motif analysis from genomic positions (See Finding Motifs from Peaks)

makeTagDirectory - creates a "tag directory" from high-throughput sequencing alignment files, performs quality control (See Creating a Tag Directory)
makeUCSCfile & makeBigWig.pl - create bedGraph file for visualization with the UCSC Genome Browser (See Creating UCSC file)
findPeaks - find peaks in ChIP-Seq data, regions in histone data, de novo transcripts from GRO-Seq (See Finding ChIP-Seq Peaks)
analyzeChIP-Seq.pl - automation of programs found above (See Automation of ChIP-Seq analysis)

annotatePeaks.pl - annotation of genomic positions, organization of motif and sequencing data, histograms, heatmaps, and more... (See Annotating Peaks, Quantification)
analyzeRNA.pl - quantification of RNA levels across transcripts (See RNA quantification)
analyzeRepeats.pl - quantification of RNA levels across repeats (See RNA quantification)
getDiffExpression.pl - Calculate differential enrichment of RNA-seq/ChIP-seq/ATAC-seq data (See RNA quantification)

Annotating Regions in the Genome (*annotatePeaks.pl*)

Homer contains a useful, all-in-one program for performing peak annotation called **annotatePeaks.pl**. In addition to associating peaks with nearby genes, **annotatePeaks.pl** can perform Gene Ontology Analysis, genomic feature association analysis (Genome Ontology), associate peaks with gene expression data, calculate ChIP-Seq Tag densities from different experiments, and find motif occurrences in peaks. **annotatePeaks.pl** can also be used to create histograms and heatmaps. Description of the annotation functions are covered below, while quantification of tags, motifs, histograms, etc. are covered here.

NOTE: If you're running annotatePeaks.pl on your laptop, you may want to use "**-noann**" to skip the full annotation routines, which use a bit of memory (up to 4Gb)

Basic usage:

annotatePeaks.pl <peak/BED file> <genome> > <output file>

i.e. **annotatePeaks.pl ERpeaks.txt hg18 > outputfile.txt**

The first two arguments, the <peak file> and <genome>, are required, and must be the first two arguments. Other optional command line arguments can be placed in any order after the first two. By default, **annotatePeaks.pl** prints the program output to *stdout*, which can be captured in a file by appending " > filename" to the command. With most uses of **annotatePeaks.pl**, the output is a data table that is meant to be opened with EXCEL or similar program. An example of the output can been seen below:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	PeakID	Chr	Start	End	Strand	Peak Sco	Focus Ra	Annotation	Detailed Anno	Distance to T	Nearest Pror	PromoterID	Nearest Unig	Nearest Refs	Nearest Ense	Gene Name	Gene Alias	Gene Descrip
2	chr18-1	chr18	69007968	69008268	+	593	0.939	intron (NR_03	intron (NR_03	74595	NR_034133	400655	Hs.579378	NR_034133		LOC400655	-	hypothetical
3	chr9-1	chr9	88209966	88210266	+	531.9	0.946	Intergenic	Intergenic	-50894	NM_001185	79670	Hs.597057	NM_001185	ENSG000000	ZCCHC6	DKFZp666B1	zinc finger, C
4	chr14-1	chr14	62337073	62337373	+	505.4	0.918	intron (NM_17	intron (NM_17	244485	NM_172375	27133	Hs.27043	NM_139318	ENSG000001	KCNH5	EAG2|H-EAG	potassium vc
5	chr17-1	chr17	5076243	5076543	+	492.1	0.936	intron (NR_03	intron (NR_03	2414	NM_207103	388325	Hs.462080	NM_207103	ENSG000001	C17orf87	FLJ32580|M	chromosome
6	chr17-2	chr17	47851714	47852014	+	476.2	0.824	Intergenic	Intergenic	-259488	NM_001082	56934	Hs.463466	NM_001082	ENSG000001	CA10	CA-RPX|CAR	carbonic anh
7	chr10-1	chr10	98420680	98420980	+	474.9	0.967	intron (NM_15	intron (NM_15	49439	NM_152309	118788	Hs.310456	NM_152309	ENSG000001	PIK3AP1	BCAP|RP11-	phosphoinos
8	chr9-2	chr9	81294389	81294689	+	456.3	0.957	Intergenic	Intergenic	-82159	NM_007005	7091	Hs.444213	NM_007005	ENSG000001	TLE4	BCE-1|BCE1	transducin-li
9	chr14-2	chr14	36817736	36818036	+	452.3	0.757	intron (NM_13	intron (NM_13	81017	NM_001195	145282	Hs.660396	NM_001195	ENSG000001	MIPOL1	DKFZp313M2	mirror-image
10	chr18-2	chr18	20049825	20050125	+	449.7	0.853	intron (NM_08	intron (NM_08	56219	NM_018030	114876	Hs.370725	NM_018030	ENSG000001	OSBPL1A	FLJ10217|OF	oxysterol bin
11	chr7-1	chr7	12226829	12227129	+	445.7	0.901	intron (NM_01	intron (NM_01	9606	NM_001134	54664	Hs.396358	NM_001134	ENSG000001	TMEM106B	FLJ11273|M	transmembr
12	chr14-3	chr14	88712188	88712488	+	443.1	0.844	intron (NM_0(	intron (NM_0(	240869	NM_005197	1112	Hs.621371	NM_001085	ENSG000000	FOXN3	C14orf116|C	forkhead box
13	chr18-3	chr18	62951924	62952224	+	443.1	0.947	Intergenic	Intergenic	-382689	NR_033921	643542	Hs.652901	NR_033921		LOC643542	-	hypothetical
14	chr3-1	chr3	32196769	32197069	+	443.1	0.87	Intergenic	Intergenic	-58256	NM_178868	152189	Hs.154986	NM_178868	ENSG000001	CMTM8	CKLFSF8|CKL	CKLF-like MA
15	chr11-1	chr11	110685448	110685748	+	425.8	0.907	Intergenic	Intergenic	-9849	NR_034154	399948	Hs.729225	NR_034154		C11orf92	DKFZp781P1	chromosome
16	chr4-1	chr4	81755366	81755666	+	423.2	0.908	intron (NM_15	intron (NM_15	279618	NM_152770	255119	Hs.527104	NM_152770	ENSG000001	C4orf22	MGC35043	chromosome

```
[ss45w@ghpcc06 ~]$ annotatePeaks.pl

        Usage: annotatePeaks.pl <peak file | tss> <genome version>  [additional options...]

        Available Genomes (required argument): (name,org,directory,default promoter set)
                mm10    mouse   /share/pkg/HOMER/4.6/.//data/genomes/mm10/      default
                hg19    human   /share/pkg/HOMER/4.6/.//data/genomes/hg19/      default
                        -- or --
                Custom: provide the path to genome FASTA files (directory or single file)
                If no genome is available, specify 'none'.
                If using FASTA file or none, may want to specify '-organism <...>'

        User defined annotation files (default is UCSC refGene annotation):
                annotatePeaks.pl accepts GTF (gene transfer formatted) files to annotate positions relative
                to custom annotations, such as those from de novo transcript discovery or Gencode.
                -gtf <gtf format file> (-gff and -gff3 can work for those files, but GTF is better)
                -ann <custom homer annotation file> (created by assignGenomeAnnotation, see website)

        Peak vs. tss/tts/rna mode (works with custom GTF file):
                If the first argument is "tss" (i.e. annotatePeaks.pl tss hg18 ...) then a TSS centric
                analysis will be carried out.  Tag counts and motifs will be found relative to the TSS.
                (no position file needed) ["tts" now works too - e.g. 3' end of gene]
                ["rna" specifies gene bodies, will automaticall set "-size given"]
                NOTE: The default TSS peak size is 4000 bp, i.e. +/- 2kb (change with -size option)
                -list <gene id list> (subset of genes to perform analysis [unigene, gene id, accession,
                         probe, etc.], default = all promoters)
                -cTSS <promoter position file i.e. peak file> (should be centered on TSS)

        Primary Annotation Options:
                -mask (Masked repeats, can also add 'r' to end of genome name)
                -m <motif file 1> [motif file 2] ... (list of motifs to find in peaks)
                        -mscore (reports the highest log-odds score within the peak)
                        -nmotifs (reports the number of motifs per peak)
                        -mdist (reports distance to closest motif)
                        -mfasta <filename> (reports sites in a fasta file - for building new motifs)
                        -fm <motif file 1> [motif file 2] (list of motifs to filter from above)
                        -rmrevopp <#> (only count sites found within <#> on both strands once, i.e. palindromic)
                        -matrix <prefix> (outputs a motif co-occurrence files:
                                prefix.count.matrix.txt - number of peaks with motif co-occurrence
                                prefix.ratio.matrix.txt - ratio of observed vs. expected  co-occurrence
                                prefix.logPvalue.matrix.txt - co-occurrence enrichment

         Advanced Options:
                -len <#> / -fragLength <#> (Fragment length, default=auto, might want to set to 1 for 5'RNA)
                -normLength <#> (Fragment length to normlize to for experiments with different lens, def: 100)
                -size <#> (Peak size[from center of peak], default=inferred from peak file)
                        -size #,# (i.e. -size -10,50 count tags from -10 bp to +50 bp from center)
                        -size "given" (count tags etc. using the actual regions - for variable length regions)
```

```
[ss45w@ghpcc06 Padilla]$
[ss45w@ghpcc06 Padilla]$
[ss45w@ghpcc06 Padilla]$
[ss45w@ghpcc06 Padilla]$ less PlusCu_peakFile.bed
```

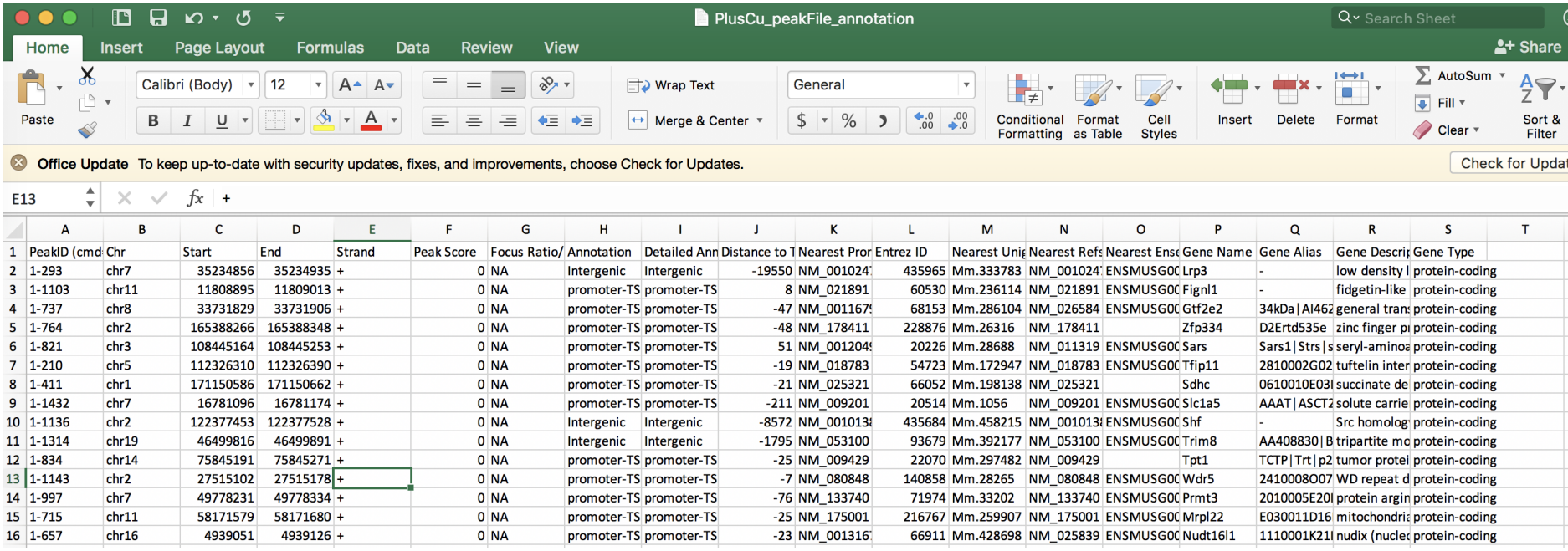
```
chr19   12005975        12006065        1-1316
chr18   67724265        67724347        Jan-77
chr15   44619086        44619172        1-1326
chr2    30364058        30364157        1-767
chr16   4624926 4625002 1-633
chr11   78497798        78497874        1-1088
chr12   16894729        16894813        1-720
chr3    103020471       103020563       1-116
chr13   65241423        65241509        1-1113
chr10   128923474       128923565       1-908
chr5    72914191        72914268        1-192
chr7    46796044        46796137        1-1426
chr5    144358391       144358486       1-1017
chr11   115933471       115933549       1-562
chr14   25457543        25457621        1-842
chrUn_GL456393  10391   10473   1-1371
chr5    34369745        34369836        1-856
chr5    109559132       109559219       1-859
chr7    101896323       101896404       1-1437
chr6    124415125       124415206       1-1076
chr11   6528786 6528872 1-1084
chrX    112370694       112370778       1-751
chr11   69632435        69632543        1-1355
chr11   4594872 4594962 1-706
chr9    104002131       104002214       1-790
chr11   98682501        98682585        1-387
chr2    181919136       181919217       1-669
chr3    36475872        36475952        Jan-37
chr7    19381874        19381960        1-1430
chr16   31948483        31948575        15-Jan
chr8    46152480        46152556        1-1235
```

```
[ss45w@ghpcc06 Padilla]$ bsub -o logT -n 1 -q short -R rusage[mem=60000] -W240 "annotatePeaks.pl PlusCu_peakFile.bed
mm10 > PlusCu_peakFile_annotation -size given"
Job <8138763> is submitted to queue <short>.
[ss45w@ghpcc06 Padilla]$ bjobs
No unfinished job found
[ss45w@ghpcc06 Padilla]$ bjobs
No unfinished job found
[ss45w@ghpcc06 Padilla]$ bjobs
JOBID      USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
8138763    ss45w   RUN   short      ghpcc06     c38b14      *ize given Jul  5 22:26
```

```
The output (if any) is above this job summary.


        Peak file = PlusCu_peakFile.bed
        Genome = mm10
        Organism = mouse
        Using actual sizes of regions
        Peak Region set to given
        Peak/BED file conversion summary:
                BED/Header formatted lines: 1452
                peakfile formatted lines: 0
                Duplicated Peak IDs: 0

        Peak File Statistics:
                Total Peaks: 1452
                Redundant Peak IDs: 0
                Peaks lacking information: 0 (need at least 5 columns per peak)
                Peaks with misformatted coordinates: 0 (should be integer)
                Peaks with misformatted strand: 0 (should be either +/- or 0/1)

        Peak file looks good!

        Reading Positions...
        -----------------------
        Finding Closest TSS...
        Annotating:.......................
                Annotation      Number of peaks Total size (bp) Log2 Enrichment
                3UTR    3.0     19679658        -1.791
                miRNA   0.0     20053   -10.488
                ncRNA   2.0     2894170 0.389
                TTS     11.0    26240586        -0.332
                pseudo  1.0     518237  1.871
                Exon    7.0     33301799        -1.328
                Intron  111.0   928533768       -2.142
                Intergenic      156.0   1679614369      -2.506
                Promoter        1126.0  28421723        6.230
                5UTR    19.0    2090448 4.106
                snoRNA  0.0     19      -10.488
                rRNA    0.0     5631    -10.488
        NOTE: If this part takes more than 2 minutes, there is a good chance
                your machine ran out of memory: consider hitting ctrl+C and rerunning
                the command with "-noann"
        To capture annotation stats in a file, use "-annStats <filename>" next time
        Annotating:.........................
```

```
Annotating:.............................
        Annotation      Number of peaks Total size (bp) Log2 Enrichment
        3UTR    3.0     19679658        -1.807
        Other   0.0     7962702 -10.504
        RC?     0.0     10979   -10.504
        RNA     0.0     114021  -10.504
        miRNA   0.0     20053   -10.504
        ncRNA   2.0     2894170 0.374
        TTS     11.0    26240586        -0.347
        LINE    6.0     543806294       -5.595
        LINE?   0.0     8168    -10.504
        srpRNA  0.0     43388   -10.504
        SINE    11.0    196891236       -3.255
        RC      0.0     65909   -10.504
        tRNA    0.0     267194  -10.504
        DNA?    0.0     142594  -10.504
        pseudo  1.0     518237  1.855
        DNA     0.0     28728583        -10.504
        Exon    7.0     33301799        -1.343
        Intron  41.0    590161527       -2.940
        Intergenic      81.0    839415489       -2.466
        Promoter        1126.0  28421723        6.215
        5UTR    19.0    2090448 4.091
        snoRNA  0.0     19      -10.504
        LTR?    0.0     193659  -10.504
        scRNA   0.0     604253  -10.504
        CpG-Island      94.0    3360987 5.713
        Low_complexity  3.0     19429679        -1.788
        LTR     13.0    313368642       -3.684
        Simple_repeat   9.0     57622492        -1.772
        snRNA   0.0     237719  -10.504
        Unknown 0.0     2421459 -10.504
        SINE?   0.0     29758   -10.504
        Satellite       24.0    4656249 3.273
        rRNA    1.0     166488  3.494
Counting Tags in Peaks from each directory...
Organism: mouse
Loading Gene Informaiton...
Outputing Annotation File...
Done annotating peaks file
```

```
------------------------------------------------------------
Sender: LSF System <lsfadmin@c38b14>
Subject: Job 8138763: <annotatePeaks.pl PlusCu_peakFile.bed mm10 > PlusCu_peakFile_annotation -size given> in cluster
 <umghpcc> Done

Job <annotatePeaks.pl PlusCu_peakFile.bed mm10 > PlusCu_peakFile_annotation -size given> was submitted from host <ghp
cc06> by user <ss45w> in cluster <umghpcc> at Sun Jul  5 22:26:57 2020
Job was executed on host(s) <c38b14>, in queue <short>, as user <ss45w> in cluster <umghpcc> at Sun Jul  5 22:26:59 2
020
</home/ss45w> was used as the home directory.
</project/umw_anthony_imbalzano/Padilla> was used as the working directory.
Started at Sun Jul  5 22:26:59 2020
Terminated at Sun Jul  5 22:27:27 2020
Results reported at Sun Jul  5 22:27:27 2020

Your job looked like:

------------------------------------------------------------
# LSBATCH: User input
annotatePeaks.pl PlusCu_peakFile.bed mm10 > PlusCu_peakFile_annotation -size given
------------------------------------------------------------

Successfully completed.

Resource usage summary:

    CPU time :                                   26.97 sec.
    Max Memory :                                 4191 MB
    Average Memory :                             3260.25 MB
    Total Requested Memory :                     60000.00 MB
    Delta Memory :                               55809.00 MB
    Max Swap :                                   -
    Max Processes :                              5
    Max Threads :                                6
    Run time :                                   28 sec.
    Turnaround time :                            30 sec.

The output (if any) is above this job summary.
```



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	PeakID (cmd	Chr	Start	End	Strand	Peak Score	Focus Ratio/	Annotation	Detailed Ann	Distance to T	Nearest Pror	Entrez ID	Nearest Unig	Nearest Refs	Nearest Ense	Gene Name	Gene Alias	Gene Descrip	Gene Type	
2	1-293	chr7	35234856	35234935	+	0	NA	Intergenic	Intergenic	-19550	NM_001024	435965	Mm.333783	NM_001024	ENSMUSG0(	Lrp3	-	low density l	protein-coding	
3	1-1103	chr11	11808895	11809013	+	0	NA	promoter-TS	promoter-TS	8	NM_021891	60530	Mm.236114	NM_021891	ENSMUSG0(	Fignl1	-	fidgetin-like	protein-coding	
4	1-737	chr8	33731829	33731906	+	0	NA	promoter-TS	promoter-TS	-47	NM_001167	68153	Mm.286104	NM_026584	ENSMUSG0(	Gtf2e2	34kDa|AI462	general trans	protein-coding	
5	1-764	chr2	165388266	165388348	+	0	NA	promoter-TS	promoter-TS	-48	NM_178411	228876	Mm.26316	NM_178411		Zfp334	D2Ertd535e	zinc finger pr	protein-coding	
6	1-821	chr3	108445164	108445253	+	0	NA	promoter-TS	promoter-TS	51	NM_001204	20226	Mm.28688	NM_011319	ENSMUSG0(	Sars	Sars1|Strs|s	seryl-aminoa	protein-coding	
7	1-210	chr5	112326310	112326390	+	0	NA	promoter-TS	promoter-TS	-19	NM_018783	54723	Mm.172947	NM_018783	ENSMUSG0(	Tfip11	2810002G02	tuftelin inter	protein-coding	
8	1-411	chr1	171150586	171150662	+	0	NA	promoter-TS	promoter-TS	-21	NM_025321	66052	Mm.198138	NM_025321		Sdhc	0610010E03	succinate de	protein-coding	
9	1-1432	chr7	16781096	16781174	+	0	NA	promoter-TS	promoter-TS	-211	NM_009201	20514	Mm.1056	NM_009201	ENSMUSG0(	Slc1a5	AAAT|ASCT2	solute carrie	protein-coding	
10	1-1136	chr2	122377453	122377528	+	0	NA	Intergenic	Intergenic	-8572	NM_001013	435684	Mm.458215	NM_001013	ENSMUSG0(	Shf	-	Src homolog	protein-coding	
11	1-1314	chr19	46499816	46499891	+	0	NA	Intergenic	Intergenic	-1795	NM_053100	93679	Mm.392177	NM_053100	ENSMUSG0(	Trim8	AA408830|B	tripartite mo	protein-coding	
12	1-834	chr14	75845191	75845271	+	0	NA	promoter-TS	promoter-TS	-25	NM_009429	22070	Mm.297482	NM_009429		Tpt1	TCTP|Trt|p2	tumor protei	protein-coding	
13	1-1143	chr2	27515102	27515178	+	0	NA	promoter-TS	promoter-TS	-7	NM_080848	140858	Mm.28265	NM_080848	ENSMUSG0(	Wdr5	2410008O07	WD repeat d	protein-coding	
14	1-997	chr7	49778231	49778334	+	0	NA	promoter-TS	promoter-TS	-76	NM_133740	71974	Mm.33202	NM_133740	ENSMUSG0(	Prmt3	2010005E20	protein argin	protein-coding	
15	1-715	chr11	58171579	58171680	+	0	NA	promoter-TS	promoter-TS	-25	NM_175001	216767	Mm.259907	NM_175001	ENSMUSG0(	Mrpl22	E030011D16	mitochondria	protein-coding	
16	1-657	chr16	4939051	4939126	+	0	NA	promoter-TS	promoter-TS	-23	NM_001316	66911	Mm.428698	NM_025839	ENSMUSG0(	Nudt16l1	1110001K21	nudix (nucle	protein-coding	

Week 5

- Overview of questions so far
- Introducing bedtools: *a powerful toolset for genome arithmetic*
  https://bedtools.readthedocs.io/en/latest/
- Running bedtools on the ghpcc06 cluster for comparing .bed files of different experiments

- Introducing NGSplot: Quick mining and visualization of NGS data by integrating genomic databases
  https://github.com/shenlab-sinai/ngsplot
- Running NGSplot on the ghpcc06 cluster for making ChIP-Seq and RNA-Seq heatmaps and extracting tag density files

- Introducing HOMER: Software for motif discovery and next-gen sequencing analysis
  http://homer.ucsd.edu/homer/ngs/
- Running HOMER on the ghpcc06 cluster for annotating bed files and performing motif analysis

http://homer.ucsd.edu/homer/motif/

HOMER

Software for motif discovery and next-gen sequencing analysis

---

HOMER Motif Analysis

HOMER contains a novel motif discovery algorithm that was designed for regulatory element analysis in genomics applications (DNA only, no protein). It is a differential motif discovery algorithm, which means that it takes two sets of sequences and tries to identify the regulatory elements that are specifically enriched in on set relative to the other. It uses ZOOPS scoring (zero or one occurrence per sequence) coupled with the hypergeometric enrichment calculations (or binomial) to determine motif enrichment. HOMER also tries its best to account for sequenced bias in the dataset. It was designed with ChIP-Seq and promoter analysis in mind, but can be applied to pretty much any nucleic acids motif finding problem.

There are several ways to perform motif analysis with HOMER. The links below introduce the various workflows for running motif analysis. In a nutshell, HOMER contains two tools, **findMotifs.pl** and **findMotifsGenome.pl**, that manage all the steps for discovering motifs in promoter and genomic regions, respectively. These scripts attempt to make it easy for the user to analyze a list of genes or genomic positions for enriched motifs. However, if you already have the sequence files that you want to analyze (i.e. FASTA files), **findMotifs.pl** (and **homer2**) can process these directly.

Analyzing lists of genes with promoter motif analysis (**findMotifs.pl**)
Analyzing genomic positions (**findMotifsGenome.pl**)
Analyzing custom FASTA files (**findMotifs.pl**, **homer2**)
Analyzing data for RNA motifs (**findMotifs.pl/findMotifsGenome.pl**)

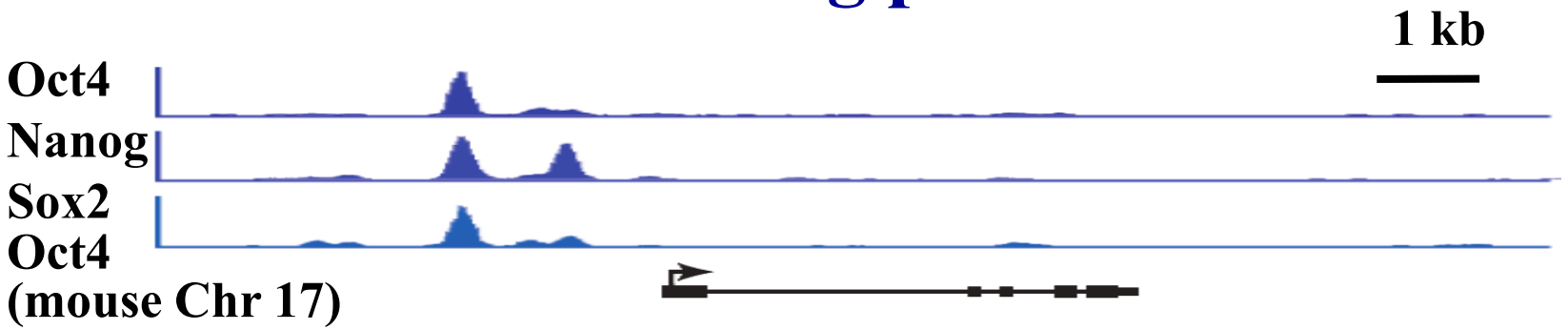
Scanning for motif across the entire genome (**scanMotifGenomeWide.pl**)

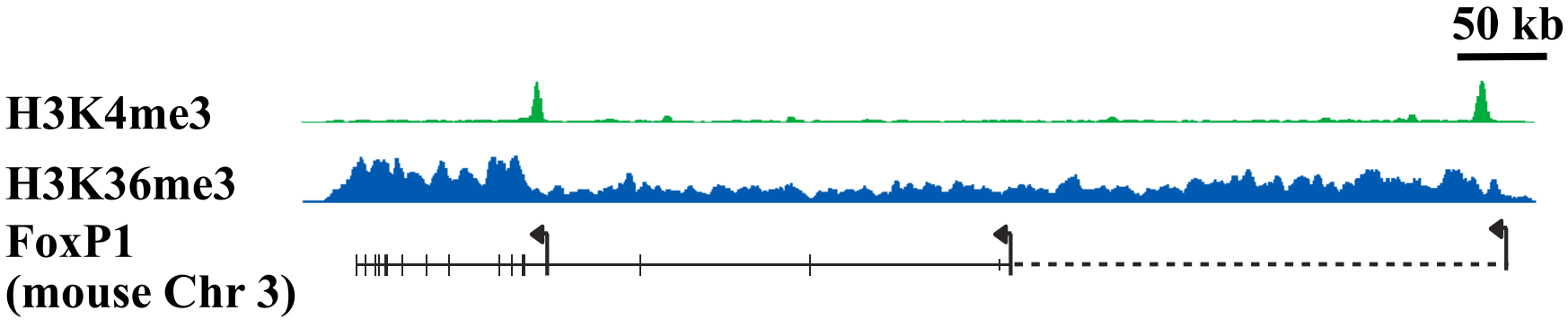
Tips for motif finding

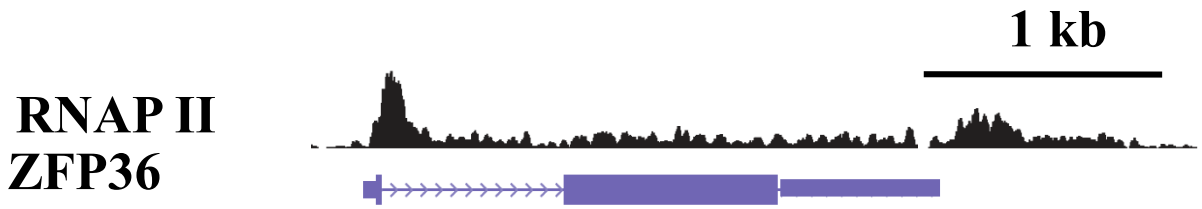
Creating custom motif files

Regardless of how you invoke HOMER, the same basic steps are executed to discover regulatory elements:

Binding profile







Kagey et al. (2010). Nature 467: 430
Mikkelsen et al. (2007). Nature 448: 553
Pepke et al. (2009). Nat. Methods 6: S22

http://homer.ucsd.edu/homer/ngs/peakMotifs.html

HOMER

Software for motif discovery and next-gen sequencing analysis

---

Finding Enriched Motifs in Genomic Regions (*findMotifsGenome.pl*)

HOMER was initially developed to automate the process of finding enriched motifs in ChIP-Seq peaks. More generally, HOMER analyzes genomic positions, not limited to only ChIP-Seq peaks, for enriched motifs. The main idea is that all the user really needs is a file containing genomic coordinates (i.e. a HOMER peak file or BED file), and HOMER will generally take care of the rest. To analyze a peak file for motifs, run the following command:

findMotifsGenome.pl <peak/BED file> <genome> <output directory> -size # [options]

i.e. **findMotifsGenome.pl ERpeaks.txt hg18 ER_MotifOutput/ -size 200 -mask**

A variety of output files will be placed in the <output directory>, including html pages showing the results. The "-mask" is optional and tells the program to use the repeat-masked sequence. (The old shorthand hg18r will also work). The **-size** parameter is now mandatory when running **findMotifsGenome.pl** to avoid confusion - plus it's always a good idea to know exactly what size the regions you are analyzing are.

The **findMotifsGenome.pl** program is a wrapper that helps set up the data for analysis using the HOMER motif discovery algorithm. By default this will perform *de novo* motif discovery as well as check the enrichment of known motifs. If you have not done so already, please look over this page describing how HOMER analyzes sequences for enriched motifs.

An important prerequisite for analyzing genomic motifs is that the appropriate genome must by configured for use with HOMER. In version v3.1, HOMER now handles custom/arbitrary genomes. Instead of intalling/configuring a genome, you can specify the path to a file or directory containing the genomic sequence in FASTA format. The genome can be in a single FASTA file, or you specify a directory where where each chromosome can be in a separate file (named chr*XXX*.fa or chr*XXX*.fa.masked). In either case, the FASTA headers must contain the chromosome names followed by white space, i.e. ">chr blahblahblah", not ">chr1-blahblahblah", or prefereably only ">chr1". (also note that homer will create a "preparsed/" directory where the genome is, so make sure you have write permissions in the genomic directory.

http://homer.ucsd.edu/homer/ngs/peakMotifs.html

Acceptable Input files

findMotifsGenome.pl accepts HOMER peak files or BED files:

HOMER peak files should have at minimum 5 columns (separated by TABs, additional columns will be ignored):

- Column1: Unique Peak ID
- Column2: chromosome
- Column3: starting position
- Column4: ending position
- Column5: Strand (+/- or 0/1, where 0="+", 1="-")

BED files should have at minimum 6 columns (separated by TABs, additional columns will be ignored)

- Column1: chromosome
- Column2: starting position
- Column3: ending position
- Column4: Unique Peak ID
- Column5: not used
- Column6: Strand (+/- or 0/1, where 0="+", 1="-")

In theory, HOMER will accept BED files with only 4 columns (+/- in the 4th column), and files without unique IDs, but this is NOT recommended. For one, if you don't have unique IDs for your regions, it's hard to go back and figure out which region contains which peak.

Mac Users: If using a EXCEL to prepare input files, make sure to save files as a "Text (Windows)" if running MacOS - saving as "Tab delimited text" in Mac produces problems for the software. Otherwise, you can run the script "**changeNewLine.pl <filename>**" to convert the Mac-formatted text file to a Windows/Dos/Unix formatted text file.

If errors occur, it is likely that the file is not in the correct format, or the first column is not actually populated with unique identifiers.

```
[ss45w@ghpcc06 ~]$ findMotifsGenome.pl

        Program will find de novo and known motifs in regions in the genome

        Usage: findMotifsGenome.pl <pos file> <genome> <output directory> [additional options]
        Example: findMotifsGenome.pl peaks.txt mm8r peakAnalysis -size 200 -len 8

        Possible Genomes:
                hg19    human
                mm10    mouse
                        -- or --
                Custom: provide the path to genome FASTA files (directory or single file)
                        Heads up: will create the directory "preparsed/" in same location.

        Basic options:
                -mask (mask repeats/lower case sequence, can also add 'r' to genome, i.e. mm9r)
                -bg <background position file> (genomic positions to be used as background, default=automatic)
                        removes background positions overlapping with target positions
                        -chopify (chop up large background regions to the avg size of target regions)
                -len <#>[,<#>,<#>...] (motif length, default=8,10,12) [NOTE: values greater 12 may cause the progra
m
                        to run out of memory - in these cases decrease the number of sequences analyzed (-N),
                        or try analyzing shorter sequence regions (i.e. -size 100)]
                -size <#> (fragment size to use for motif finding, default=200)
                        -size <#,#> (i.e. -size -100,50 will get sequences from -100 to +50 relative from center)
                        -size given (uses the exact regions you give it)
                -S <#> (Number of motifs to optimize, default: 25)
                -mis <#> (global optimization: searches for strings with # mismatches, default: 2)
                -norevopp (don't search reverse strand for motifs)
                -nomotif (don't search for de novo motif enrichment)
                -rna (output RNA motif logos and compare to RNA motif database, automatically sets -norevopp)

        Scanning sequence for motifs
                -find <motif file> (This will cause the program to only scan for motifs)
```

bsub -o logT -n 1 -q short -R rusage[mem=60000] -W240 "findMotifsGenome.pl -preparsedDir D0_Med1_q0.1_peaks.narrowPeak.bed mm10 D0_Med1_q0.1_peaks_motif -size given"

File location
/nl/umw_anthony_imbalzano/Sabriya/MandrupChIP/

```
[ss45w@ghpcc06 MandrupChIP]$ ls D0_Med1_q0.1_peaks_motif/
homerMotifs.all.motifs  homerMotifs.motifs12  homerResults       knownResults
 knownResults.txt            seq.autonorm.tsv
homerMotifs.motifs10    homerMotifs.motifs8   homerResults.html  knownResults.html
 motifFindingParameters.txt
```

Homer *de novo* Motif Results (D0_unique_motifs/)

Known Motif Enrichment Results
Gene Ontology Enrichment Results
If Homer is having trouble matching a motif to a known motif, try copy/pasting the matrix file into STAMP
More information on motif finding results: HOMER | Description of Results | Tips
Total target sequences = 33636
Total background sequences = 33464
* - possible false positive

Rank	Motif	P-value	log P-pvalue	% of Targets	% of Background	STD(Bg STD)	Best Match/Details	Motif File
1		1e-289	-6.660e+02	21.01%	13.74%	140.5bp (73.4bp)	NFIX/MA0671.1/Jaspar(0.950) More Information | Similar Motifs Found	motif file (matrix)
2		1e-265	-6.112e+02	6.58%	2.89%	133.1bp (66.2bp)	RUNX(Runt)/HPC7-Runx1-ChIP-Seq(GSE22178)/Homer(0.986) More Information | Similar Motifs Found	motif file (matrix)
3		1e-248	-5.713e+02	13.03%	7.69%	147.8bp (70.0bp)	TEAD3(TEA)/HepG2-TEAD3-ChIP-Seq(Encode)/Homer(0.928) More Information | Similar Motifs Found	motif file (matrix)
4		1e-247	-5.697e+02	16.55%	10.52%	144.1bp (70.9bp)	TWIST1/MA1123.1/Jaspar(0.987) More Information | Similar Motifs Found	motif file (matrix)
5		1e-243	-5.604e+02	12.18%	7.08%	137.6bp (67.9bp)	Fra1(bZIP)/BT549-Fra1-ChIP-Seq(GSE46166)/Homer(0.949) More Information | Similar Motifs Found	motif file (matrix)
6		1e-214	-4.949e+02	24.18%	17.40%	152.5bp (67.3bp)	AP-2gamma(AP2)/MCF7-TFAP2C-ChIP-Seq(GSE21234)/Homer(0.883) More Information | Similar Motifs Found	motif file (matrix)
7		1e-188	-4.336e+02	2.20%	0.60%	117.0bp (65.7bp)	BORIS(Zf)/K562-CTCFL-ChIP-Seq(GSE32465)/Homer(0.925) More Information | Similar Motifs Found	motif file (matrix)

Homer Known Motif Enrichment Results (D0_unique_motifs)

Homer *de novo* Motif Results
Gene Ontology Enrichment Results
Known Motif Enrichment Results (txt file)
Total Target Sequences = 33632, Total Background Sequences = 33586

Rank	Motif	Name	P-value	log P-pvalue
1		Pitx1:Ebox(Homeobox,bHLH)/Hindlimb-Pitx1-ChIP-Seq(GSE41591)/Homer	1e-714	-1.646e+03
2		NF1(CTF)/LNCAP-NF1-ChIP-Seq(Unpublished)/Homer	1e-389	-8.959e+02
3		CTCF(Zf)/CD4+-CTCF-ChIP-Seq(Barski_et_al.)/Homer	1e-210	-4.857e+02
4		AP-2gamma(AP2)/MCF7-TFAP2C-ChIP-Seq(GSE21234)/Homer	1e-210	-4.845e+02
5		AP-2alpha(AP2)/Hela-AP2alpha-ChIP-Seq(GSE31477)/Homer	1e-208	-4.794e+02
6		Fra1(bZIP)/BT549-Fra1-ChIP-Seq(GSE46166)/Homer	1e-206	-4.759e+02
7		TEAD1(TEAD)/HepG2-TEAD1-ChIP-Seq(Encode)/Homer	1e-202	-4.653e+02

http://homer.ucsd.edu/homer/introduction/basics.html

Known Motif Enrichment

First and most important: There is a subtle but IMPORTANT difference between looking for motifs de novo and looking for known motif enrichment. De novo motif discovery allows you to directly query the sequence to discover which motifs are the MOST enriched sequences in your target set. Known motif discovery will simply tell you which of the known motifs is most enriched in your target set.

This may not seem important but consider the following scenario: You have a set of random GA-rich sequences and compare them to random genomic sequences. De novo motif finding will likely return a G/A-rich matrix that doesn't look anything like a transcription factor. Known motif finding will return astonishingly high p-values for motifs like PU.1 (GAGGAAGT) and ISRE (GAAACTGAAA). Because of this de novo motif finding results are much more trustful in terms of results.

The greatest advantage to using known motifs is found when you have a limited set of target sequences. The less data that is available or the weaker the true signal, it is difficult for de novo motif finding to accurately define a signal that is significant. Known motifs have the advantage of many less degrees of freedom and in may cases find the correct motifs when the enrichment falls below the 1e-10 thresholds for reliability when considering de novo results.

A more detailed description of the motif finding procedure is available in the Motif Finding Tutorial.

Week 5

- Overview of questions so far
- Introducing bedtools: *a powerful toolset for genome arithmetic*
  https://bedtools.readthedocs.io/en/latest/
- Running bedtools on the ghpcc06 cluster for comparing .bed files of different experiments

- Introducing NGSplot: Quick mining and visualization of NGS data by integrating genomic databases
  https://github.com/shenlab-sinai/ngsplot
- Running NGSplot on the ghpcc06 cluster for making ChIP-Seq and RNA-Seq heatmaps and extracting tag density files

- Introducing HOMER: Software for motif discovery and next-gen sequencing analysis
  http://homer.ucsd.edu/homer/ngs/
- Running HOMER on the ghpcc06 cluster for annotating bed files and performing motif analysis

http://barc.wi.mit.edu/education/hot_topics/ngsplot/ngsplot_Apr2014.pdf

Why ngsplot?

- Summarize data *visually* at functional genomic regions (eg. TSSs, exons, enhancers, etc.)
- Epigenetics:
  - ➢ histone modifications or marks enriched near TSS
  - ➢ co-occurrence of histone marks
- ChIP-Seq:
  - ➢ enrichment of TF in the promoter or gene body
- Genome browsers may not capture enrichment information since they're good at viewing *slices* of the genome.

http://barc.wi.mit.edu/education/hot_topics/ngsplot/ngsplot_Apr2014.pdf

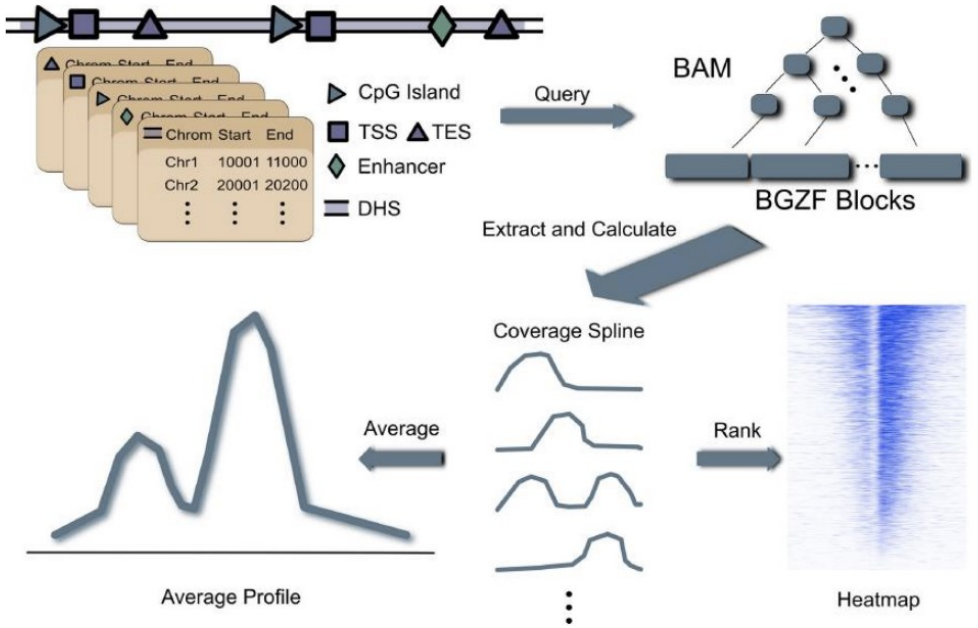
Overview: ngsplot

Step 1: Defining region(s) of interest

Input: species and regions from ngsplot database, BAM file from alignment

Step 2: Meaningful plots in the defined region(s)

Output: Average profile plot and heatmap images

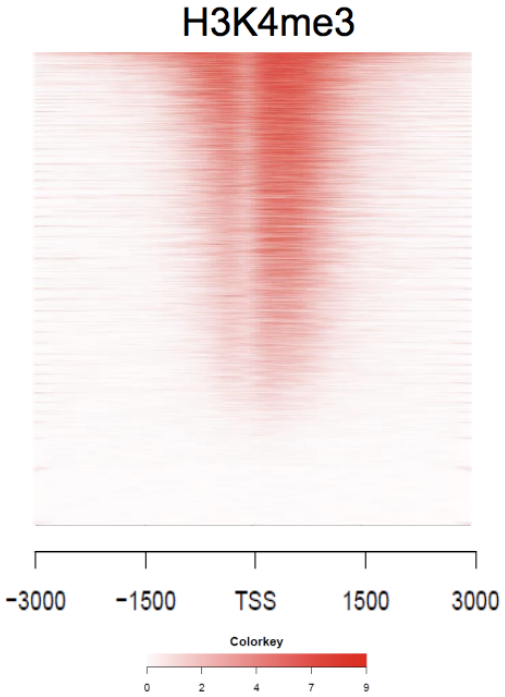


ngsplot: Implementation

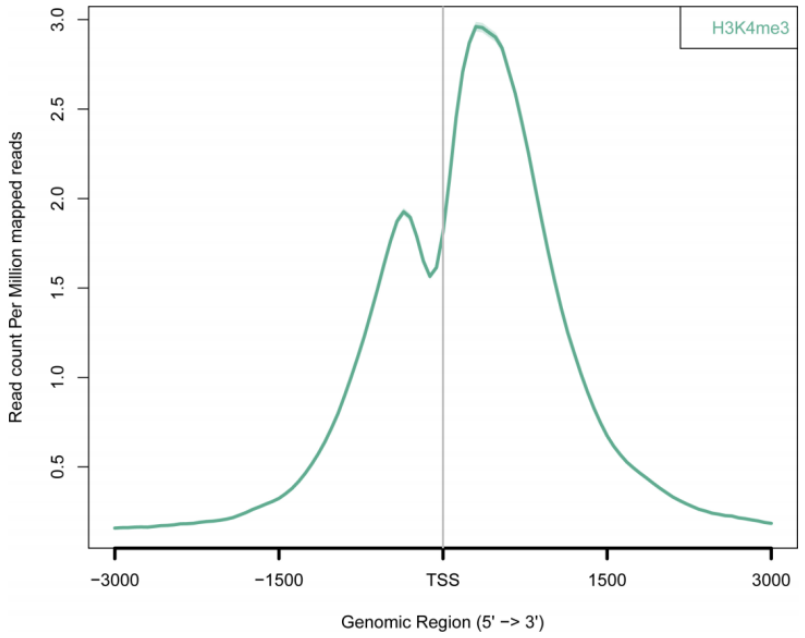
- Find genomic coordinates on regions of interest
  - use ngsplot database with predefined regions
  - custom regions in BED file
- Normalization and coverage based on aligned reads
- Generate plots: average profile and heatmaps

Hands-on

- Examining enrichment near TSS



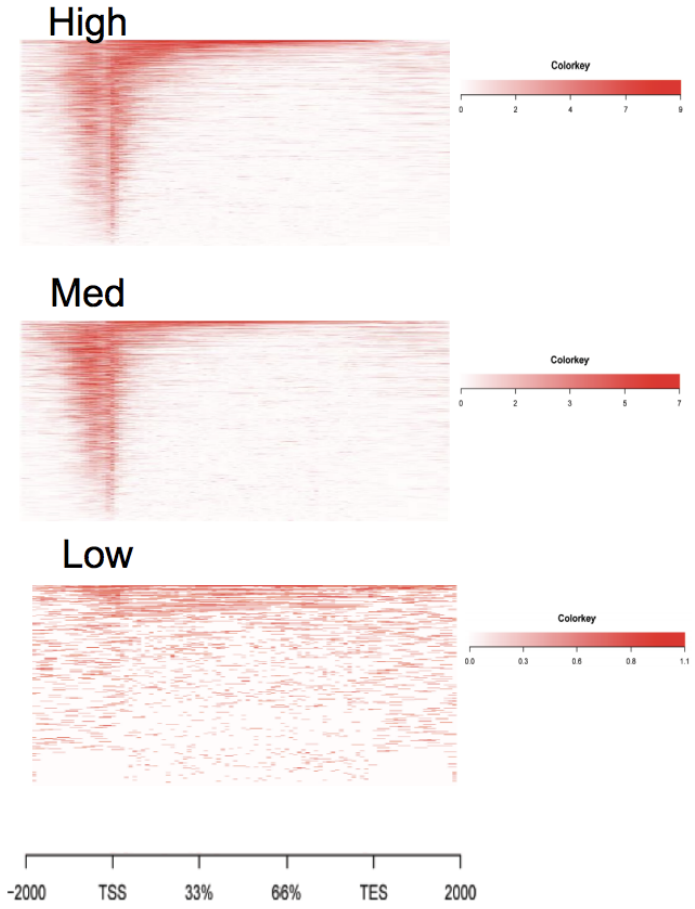
Heatmap showing H3K4me3 enrichment (by color intensity and region) near TSS, where each row is a gene.

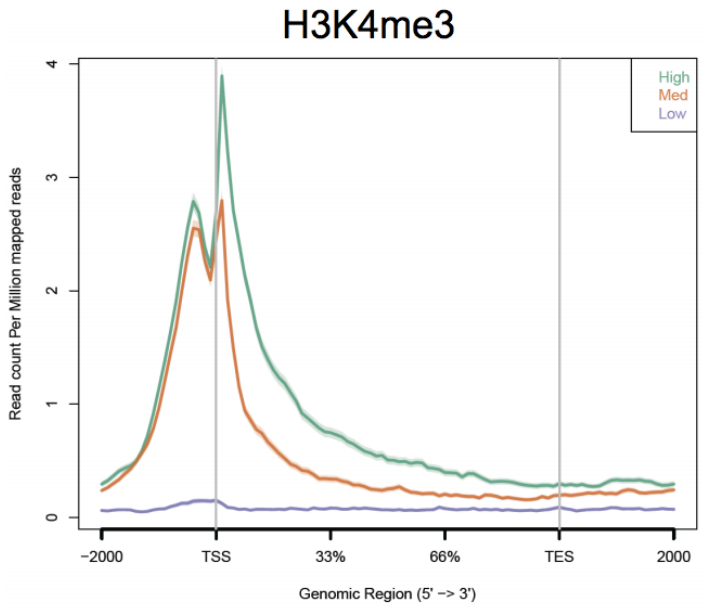


Average profile plot summarizing the heatmap (left), note: all genes/features are now collapsed. H3K4me3 enrichment can be clearly seen near the TSS. The two peaks can be also seen on the heatmap (left) by the two distinct banding pattern separated by the TSS.

Hands-on

- Multiple plots on the same graph with different subsets of genes

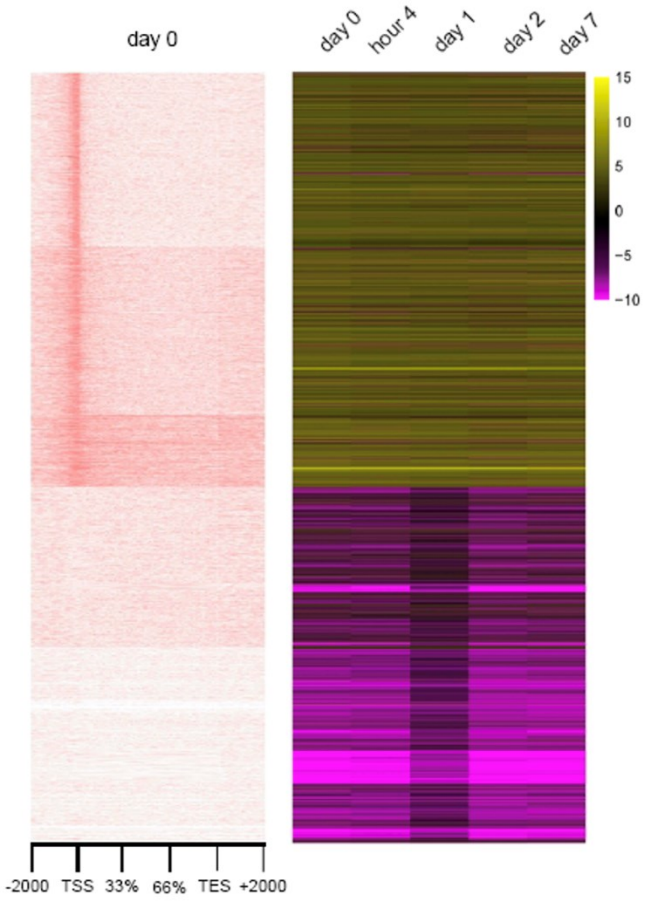




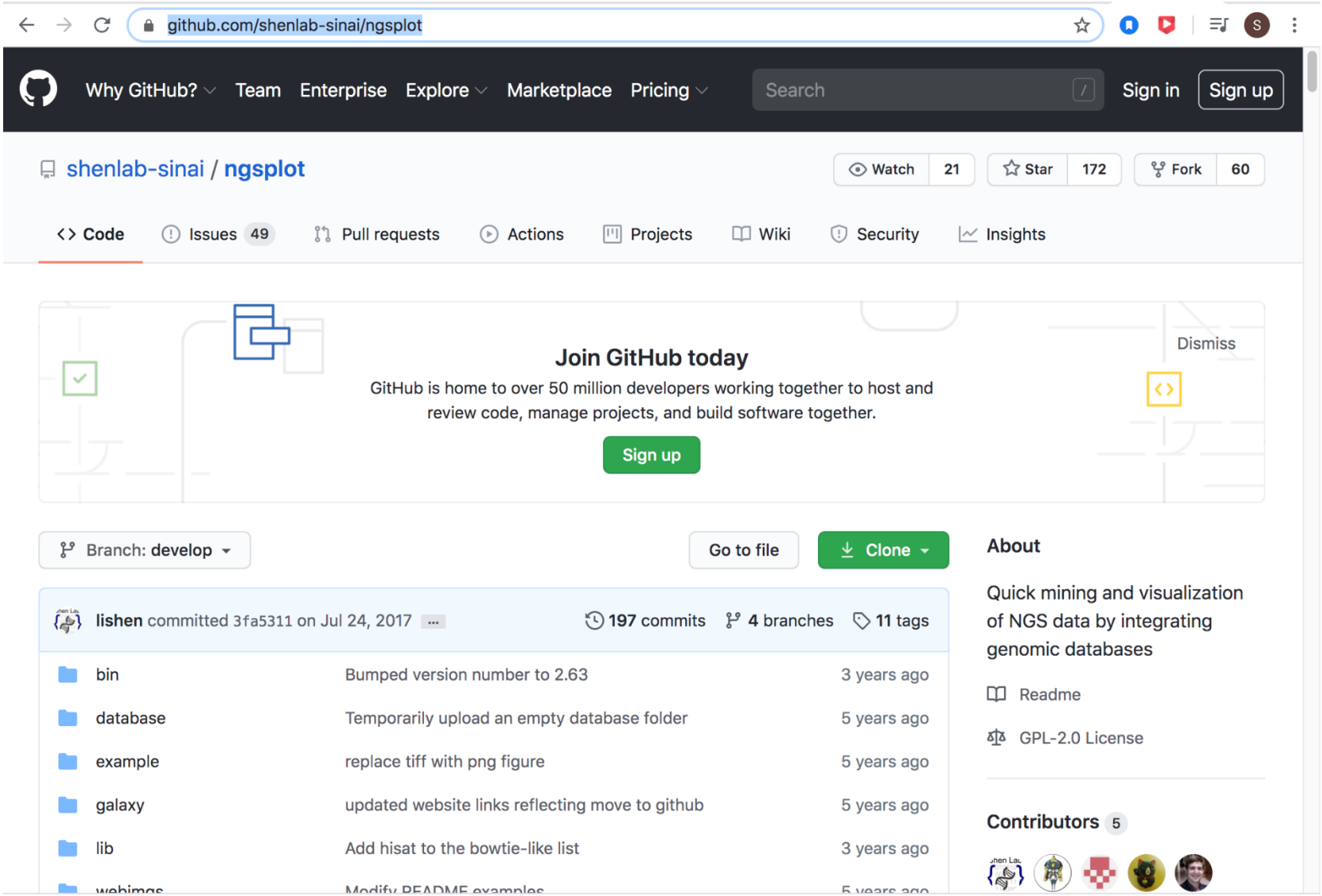
Heatmaps(left) and average profile plot with three different subsets of genes (low, medium, high) based on expression level. Genes that are highly/moderately expressed have an enrichment for H3K4me3 near the TSS that's not seen in lowly expressed genes.

ngsplot: Ranking Genes

- Genes on the heatmap can be ranked in different orders (-GO option):
  - Total (default)
  - Hierarchial clustering
  - Max
  - Product
  - Difference
  - Principal Component Analysis (PCA)
  - none



https://github.com/shenlab-sinai/ngsplot



Running NGSplot on GHPCC06.umassrc.org

```
[ss45w@ghpcc06 TereChIP]$ module load ngsplot/2.63
R 3.1.0 is located under /share/pkg/R/3.1.0
When compiling modules for this, be sure to load gcc/4.8.1
python 2.7.14 is located under /share/pkg/python/2.7.14
python/2.7.14_packages/biopython 1.70 is located under /share/pkg/python/2.7.14_packages/biopython/1.70
ngsplot 2.63 is located under /share/pkg/ngsplot/2.63.
[ss45w@ghpcc06 TereChIP]$
[ss45w@ghpcc06 TereChIP]$
[ss45w@ghpcc06 TereChIP]$
[ss45w@ghpcc06 TereChIP]$ module unload R/3.1.0
```

```
[ss45w@ghpcc06 TereChIP]$ module load R/3.6.0
R 3.6.0 is located under /share/pkg/R/3.6.0
When compiling modules for this, be sure to load gcc/8.1.0
[ss45w@ghpcc06 TereChIP]$
```

NGSplot requires several R packages

```
[ss45w@ghpcc06 TereChIP]$ Rscript -e "library(doMC)"
Loading required package: foreach
Loading required package: iterators
Loading required package: parallel
[ss45w@ghpcc06 TereChIP]$ Rscript -e "library(caTools)"
```

```
[ss45w@ghpcc06 TereChIP]$ Rscript -e "library(BSgenome)"
[ss45w@ghpcc06 TereChIP]$ Rscript -e "library(BSgenome)"
Loading required package: BiocGenerics
Loading required package: parallel

Attaching package: 'BiocGenerics'

The following objects are masked from 'package:parallel':

    clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
    clusterExport, clusterMap, parApply, parCapply, parLapply,
    parLapplyLB, parRapply, parSapply, parSapplyLB

The following objects are masked from 'package:stats':

    IQR, mad, sd, var, xtabs

The following objects are masked from 'package:base':

    anyDuplicated, append, as.data.frame, basename, cbind, colnames,
    dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
    grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
    order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
    rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
    union, unique, unsplit, which, which.max, which.min

Loading required package: S4Vectors
Loading required package: stats4

Attaching package: 'S4Vectors'

The following object is masked from 'package:base':

    expand.grid

Loading required package: IRanges
Loading required package: GenomeInfoDb
Loading required package: GenomicRanges
Loading required package: Biostrings
Loading required package: XVector

Attaching package: 'Biostrings'

The following object is masked from 'package:base':

    strsplit

Loading required package: rtracklayer
[ss45w@ghpcc06 TereChIP]$
```

```
[ss45w@ghpcc06 TereChIP]$ Rscript -e "library(Rsamtools)"
[ss45w@ghpcc06 TereChIP]$ Rscript -e "library(Rsamtools)"
Loading required package: GenomeInfoDb
Loading required package: BiocGenerics
Loading required package: parallel

Attaching package: 'BiocGenerics'

The following objects are masked from 'package:parallel':

    clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
    clusterExport, clusterMap, parApply, parCapply, parLapply,
    parLapplyLB, parRapply, parSapply, parSapplyLB

The following objects are masked from 'package:stats':

    IQR, mad, sd, var, xtabs

The following objects are masked from 'package:base':

    anyDuplicated, append, as.data.frame, basename, cbind, colnames,
    dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
    grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
    order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
    rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
    union, unique, unsplit, which, which.max, which.min

Loading required package: S4Vectors
Loading required package: stats4

Attaching package: 'S4Vectors'

The following object is masked from 'package:base':

    expand.grid

Loading required package: IRanges
Loading required package: GenomicRanges
Loading required package: Biostrings
Loading required package: XVector

Attaching package: 'Biostrings'

The following object is masked from 'package:base':

    strsplit
```

[ss45w@ghpcc06 TereChIP]$ Rscript -e "library(ShortRead)"

```
[ss45w@ghpcc06 TereChIP]$ Rscript -e "library(ShortRead)"
Loading required package: BiocGenerics
Loading required package: parallel

Attaching package: 'BiocGenerics'

The following objects are masked from 'package:parallel':

    clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
    clusterExport, clusterMap, parApply, parCapply, parLapply,
    parLapplyLB, parRapply, parSapply, parSapplyLB

The following objects are masked from 'package:stats':

    IQR, mad, sd, var, xtabs

The following objects are masked from 'package:base':

    anyDuplicated, append, as.data.frame, basename, cbind, colnames,
    dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
    grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
    order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
    rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
    union, unique, unsplit, which, which.max, which.min

Loading required package: BiocParallel
Loading required package: Biostrings
Loading required package: S4Vectors
Loading required package: stats4

Attaching package: 'S4Vectors'

The following object is masked from 'package:base':

    expand.grid

Loading required package: IRanges
Loading required package: XVector

Attaching package: 'Biostrings'

The following object is masked from 'package:base':

    strsplit

Loading required package: Rsamtools
Loading required package: GenomeInfoDb
Loading required package: GenomicRanges
Loading required package: GenomicAlignments
Loading required package: SummarizedExperiment
Loading required package: Biobase
Welcome to Bioconductor

    Vignettes contain introductory material; view with
    'browseVignettes()'. To cite Bioconductor, see
    'citation("Biobase")', and for packages 'citation("pkgname")'.

Loading required package: DelayedArray
Loading required package: matrixStats

Attaching package: 'matrixStats'

The following objects are masked from 'package:Biobase':

    anyMissing, rowMedians


Attaching package: 'DelayedArray'

The following objects are masked from 'package:matrixStats':

    colMaxs, colMins, colRanges, rowMaxs, rowMins, rowRanges

The following object is masked from 'package:Biostrings':

    type

The following objects are masked from 'package:base':

    aperm, apply, rowsum
```

```
[ss45w@ghpcc06 TereChIP]$ ngs.plot.r
Unpaired argument and value.

Visit https://github.com/shenlab-sinai/ngsplot/wiki/ProgramArguments101 for details
Version: 2.63
Usage: ngs.plot.r -G genome -R region -C [cov|config]file
                  -O name [Options]

## Mandatory parameters:
  -G   Genome name. Use ngsplotdb.py list to show available genomes.
  -R   Genomic regions to plot: tss, tes, genebody, exon, cgi, enhancer, dhs or bed
  -C   Indexed bam file or a configuration file for multiplot
  -O   Name for output: multiple files will be generated
## Optional parameters related to configuration file:
  -E   Gene list to subset regions OR bed file for custom region
  -T   Image title
## Coverage-generation parameters:
  -F   Further information provided to select database table or plottype:
         This is a string of description separated by comma.
         E.g. protein_coding,K562,rnaseq(order of descriptors does not matter)
              means coding genes in K562 cell line drawn in rnaseq mode.
  -D   Gene database: ensembl(default), refseq
  -L   Flanking region size(will override flanking factor)
  -N   Flanking region factor
  -RB  The fraction of extreme values to be trimmed on both ends
         default=0, 0.05 means 5% of extreme values will be trimmed
  -S   Randomly sample the regions for plot, must be:(0, 1]
  -P   #CPUs to use. Set 0(default) for auto detection
  -AL  Algorithm used to normalize coverage vectors: spline(default), bin
  -CS  Chunk size for loading genes in batch(default=100)
  -MQ  Mapping quality cutoff to filter reads(default=20)
  -FL  Fragment length used to calculate physical coverage(default=150)
  -SS  Strand-specific coverage calculation: both(default), same, opposite
  -IN  Shall interval be larger than flanking in plot?(0 or 1, default=automatic)
  -FI  Forbid image output if set to 1(default=0)
## Plotting-related parameters:
### Misc. parameters:
    -FS Font size(default=12)
### Avg. profiles parameters:
    -WD Image width(default=8)
    -HG Image height(default=7)
    -SE  Shall standard errors be plotted?(0 or 1)
    -MW  Moving window width to smooth avg. profiles, must be integer
           1=no(default); 3=slightly; 5=somewhat; 9=quite; 13=super.
    -H   Opacity of shaded area, suggested value:[0, 0.5]
           default=0, i.e. no shading, just lines
    -YAS Y-axis scale: auto(default) or min_val,max_val(custom scale)
    -LEG Draw legend? 1(default) or 0
    -BOX Draw box around plot? 1(default) or 0
    -VLN Draw vertical lines? 1(default) or 0
    -XYL Draw X- and Y-axis labels? 1(default) or 0
    -LWD Line width(default=3)
### Heatmap parameters:
    -GO  Gene order algorithm used in heatmaps: total(default), hc, max,
           prod, diff, km and none(according to gene list supplied)
    -LOW Low count cutoff(default=10) in rank-based normalization
    -KNC K-means or HC number of clusters(default=5)
    -MIT Maximum number of iterations(default=20) for K-means
    -NRS Number of random starts(default=30) in K-means
    -RR  Reduce ratio(default=30). The parameter controls the heatmap height
           The smaller the value, the taller the heatmap
    -SC  Color scale used to map values to colors in a heatmap
           local(default): base on each individual heatmap
           region: base on all heatmaps belong to the same region
           global: base on all heatmaps together
           min_val,max_val: custom scale using a pair of numerics
    -FC  Flooding fraction:[0, 1), default=0.02
    -CO  Color for heatmap. For bam-pair, use color-tri(neg_color:[neu_color]:pos_color)
           Hint: must use R colors, such as darkgreen, yellow and blue2
                 The neutral color is optional(default=black)
    -CD  Color distribution for heatmap(default=0.6). Must be a positive number
           Hint: lower values give more widely spaced colors at the negative end
                 In other words, they shift the neutral color to positive values
                 If set to 1, the neutral color represents 0(i.e. no bias)
```

ngs.plot.r arguments

- Mandatory arguments

Argument	Explanation
-G	Genome name (hg19, mm9,...)
-R	Genomic regions to plot (tss, tes, genebody, exon,...)
-C	Bam file or a configuration file for multiple plot
-O	Name of output

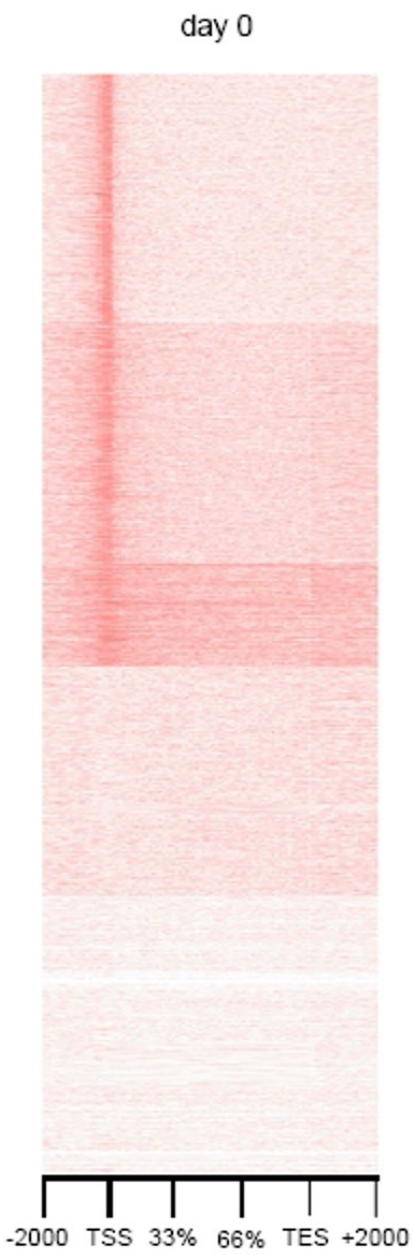
- Optional arguments (incomplete list)

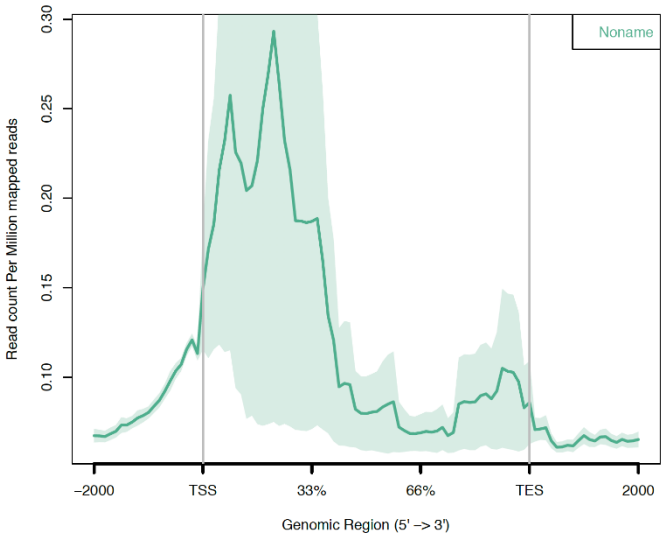
Argument	Explanation
-AL	Algorithm to normalize coverage vectors (spline or bin)
-GO	Gene order algorithm (total, hc, max,...)
-FL	Fragment length (eg. fragment size from experiment)
-D	Gene database (ensembl, refseq)
-E	Gene list to subset regions
-L	Flanking region size (in bases)
-T	Image title/name

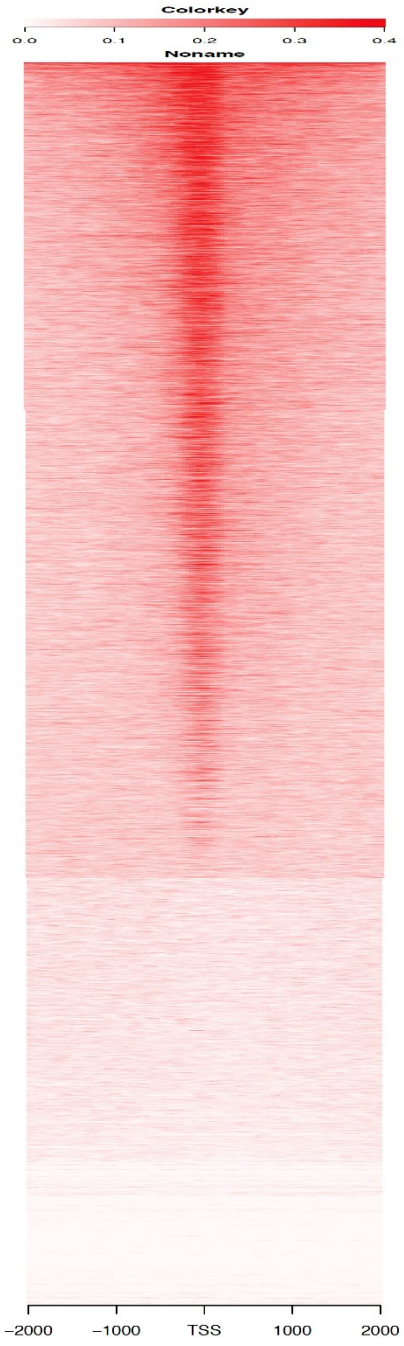
bsub -o logT -n 1 -q short -R rusage[mem=20000] -W240 "ngs.plot.r -G mm10 -R genebody -C D0_Med1_2_SRR5297599.sorted.bam -O D0_Med1_2_3T3L1_genebody.km -L 2000 -GO km"

ngs.plot.r -G mm10 -R tss -L 2000 -C /Volumes/syeds/Imbalzano_Lab/ChIP-Seq_PearsonCorrelation/MACS_D0_Prmt5merged/D0_Prmt5_n1n2merged.bam -O D0_Prmt5_TSS_2kb

NGSplot output example







NGSplot output example – tag density values (100 bins)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	gid	gname	tid	strand	1	2	3	4	5	6	7	8	9	10	11	12	13	
2	ENSMUSG00000021252	0610007P14Rik	ENSMUST00000021676	-	0.15841272	0.05280424	0.2640212	0.21230815	2.99E-48	0.05280424	0.05280424	0	0.05280424	0	0	0.15692399	0.10479262	0.052804
3	ENSMUSG00000007777	0610009B22Rik	ENSMUST00000109098	-	0.10560848	0.05280424	0	0	0.05280424	0	0.10560848	0.26402123	0.1584994	0.05280425	0.10560795	0.15841272	0	0.158412
4	ENSMUSG00000037361	0610009D07Rik	ENSMUST00000046207	+	0	0.10560848	1.00E-15	5.27E-06	0.2640212	0.10597666	0.21121692	0.21122431	0.12580245	3.89E-16	0.10560848	0.15841272	7.87E-13	5.01E-
5	ENSMUSG00000043644	0610009L18Rik	ENSMUST00000143813	+	0.15841272	0.31592279	0.26404999	0	0	1.62E-82	2.71E-25	0.21121696	0.42246324	0.15921053	3.78E-08	0	0.15841272	0.211325
6	ENSMUSG00000024442	0610009O20Rik	ENSMUST00000025314	+	0	0	0.05280424	0	0	0	4.68E-79	7.57E-22	0.21121696	0	0.05280477	0.10563712	0.05280424	2.80E-
7	ENSMUSG00000042208	0610010F05Rik	ENSMUST00000155903	-	0.05280424	3.84E-36	0.10560848	0.15841234	2.20E-07	1.79E-64	9.71E-45	0.05280424	0.05280424	0.21121013	0.05280477	0.2640212	0.15841272	0.05387
8	ENSMUSG00000020831	0610010K14Rik	ENSMUST00000102569	-	0.10560848	0.05280425	0.15841272	1.00E-11	0.2640212	0.20828769	0.05280424	5.34E-31	0.05291355	0.23521123	0.2640212	0.15841268	0.10560848	0.052804
9	ENSMUSG00000025731	0610011F06Rik	ENSMUST00000163356	+	0.10560848	0.10571847	9.53E-40	2.71E-08	0.09690309	0.05280424	0.05280424	0.05288104	0.05280427	0.10560848	0.10560848	0.05280424	0.10560848	0.158412
10	ENSMUSG00000055312	0610012H03Rik	ENSMUST00000068813	+	0.10560848	0.05280424	0.05280527	0.05484043	0.10560848	0.05274979	0.05280424	0.05280424	0.05280424	0	0	0	0	
11	ENSMUSG00000058706	0610030E20Rik	ENSMUST00000077783	+	0.58084665	0.26402698	0.15841272	0	0.05280424	7.06E-24	0.21121696	0.21120914	0.15841272	0.16396875	0.05281157	0.1056448	0.10520055	0.101043
12	ENSMUSG00000001418	0610031J06Rik	ENSMUST00000177005	+	0	0.21121696	0.10559788	0.00619564	1.18E-19	0.05290289	0.10560848	0.05280368	0.05280424	2.53E-09	1.73E-66	8.05E-85	1.15E-27	0.052804
13	ENSMUSG00000028608	0610037L13Rik	ENSMUST00000135454	+	0.15841272	0.05280423	0	0.05280424	0.05280424	0.05280424	0.10560848	4.03E-08	0.10560848	1.82E-10	1.24E-67	3.42E-75	4.88E-18	0.211216
14	ENSMUSG00000060512	0610040J01Rik	ENSMUST00000081747	+	0	0	0	0	0	0	0	0.05280424	0	0	0	8.21E-84	1.17E-26	0.105608
15	ENSMUSG00000051748	1100001G20Rik	ENSMUST00000070832	+	0.05280424	1.75E-19	0.05280424	0.15841272	0.10560847	0.00151693	0.05280424	0.10560848	0.10256364	3.29E-58	2.25E-115	0	0	0.052804
16	ENSMUSG00000062691	1110001A16Rik	ENSMUST00000180077	+	0.05280424	0	0.05280424	0.05323627	0	0.36962969	0.58848965	0.31682544	0.21122196	0.26402487	0.10560848	0.05280424	0.21121906	0.211216
17	ENSMUSG00000019689	1110001J03Rik	ENSMUST00000147651	+	0.10560848	0.15841272	0.26442216	0.05280484	0.10560848	0.10843875	0.15841272	0.06996661	0.05280424	0.10560848	0.05280371	9.36E-13	0.05280424	0.211216
18	ENSMUSG00000090066	1110002E22Rik	ENSMUST00000163080	+	0.05280424	0.10560848	0.10560848	0.05280424	0.05280379	0.10560848	0.15841272	0	0.0471225	0.05280424	0.05280424	0.15841272	0.05280634	0.052881
19	ENSMUSG00000071456	1110002L01Rik	ENSMUST00000168012	-	0	0	0.05280424	0.1586442	0.15841272	0.15841272	0	0	0.05280424	0.10560848	0.15841272	0.15841272	0.05318288	0.211216
20	ENSMUSG00000022972	1110004E09Rik	ENSMUST00000023694	-	0.05280424	0.21121696	0.10560848	0.16221229	0.05280424	0.10560848	0.15841263	0.15841287	0.21121696	0.31682544	0.10560856	0.15830584	0.10552181	0.017162
21	ENSMUSG00000030663	1110004F10Rik	ENSMUST00000032899	+	0	0.21121696	0.20952008	0.10560848	0.05280424	6.85E-22	0.15851826	0.1640773	0.21121696	0.15841272	5.01E-31	0.10560668	0.19101677	0.158412
22	ENSMUSG00000037960	1110007C09Rik	ENSMUST00000048946	-	0	0	0	0.05280424	0.05280424	0	0.05280424	0.05251763	3.21E-08	0.05280232	0.10560848	0	0.05280424	0.052804
23	ENSMUSG00000027637	1110008F13Rik	ENSMUST00000029165	+	0	0.05280424	0.21121676	0.10399649	0.15841272	0.05280431	0.15841272	0.10960049	0.05280424	0	0.15841272	0.26404672	0.36980303	0.633244
24	ENSMUSG00000029600	1110008J03Rik	ENSMUST00000111884	-	0	0.21121696	0.2112222	0.2640212	0.05280424	9.72E-43	0	0.10560848	0.0482988	0.15841272	0.10560848	0.31187595	0.21876933	0.052804
25	ENSMUSG00000021023	1110008L16Rik	ENSMUST00000184980	+	0.05280424	0	0.05280424	0	0	0	0	0.09994391	0.10560847	0	0	0.10560863	0.05280424	0.052804

This information can be refined and used to generate Peak density heatmaps for specific lists of genes (e.g. Mtf1 targets sites!)

NGSplot kmeans clustering gene order

```
[ss45w@ghpcc06 TereChIP]$ ExtractGName.R
Usage: ExtractGname.R file
Extract gene names and cluster info for ngsplot

File can either be the zip file containing the RData file or the RData
file directly.
Output varies depending on input. If input data file has no cluster
information, only one gene_name.txt file produced for each region.
If cluster information present, an additional C*.txt file
produced for each cluster for each region.
```

Week 5

- Overview of questions so far
- Introducing bedtools: *a powerful toolset for genome arithmetic*
  https://bedtools.readthedocs.io/en/latest/
- Running bedtools on the ghpcc06 cluster for comparing .bed files of different experiments

- Introducing NGSplot: Quick mining and visualization of NGS data by integrating genomic databases
  https://github.com/shenlab-sinai/ngsplot
- Running NGSplot on the ghpcc06 cluster for making ChIP-Seq and RNA-Seq heatmaps and extracting tag density files

- Introducing HOMER: Software for motif discovery and next-gen sequencing analysis
  http://homer.ucsd.edu/homer/ngs/
- Running HOMER on the ghpcc06 cluster for annotating bed files and performing motif analysis