Gene Annotation & ChIP-Seq Data Analysis 2

Lauren Vanderlinden

BIOS 6660

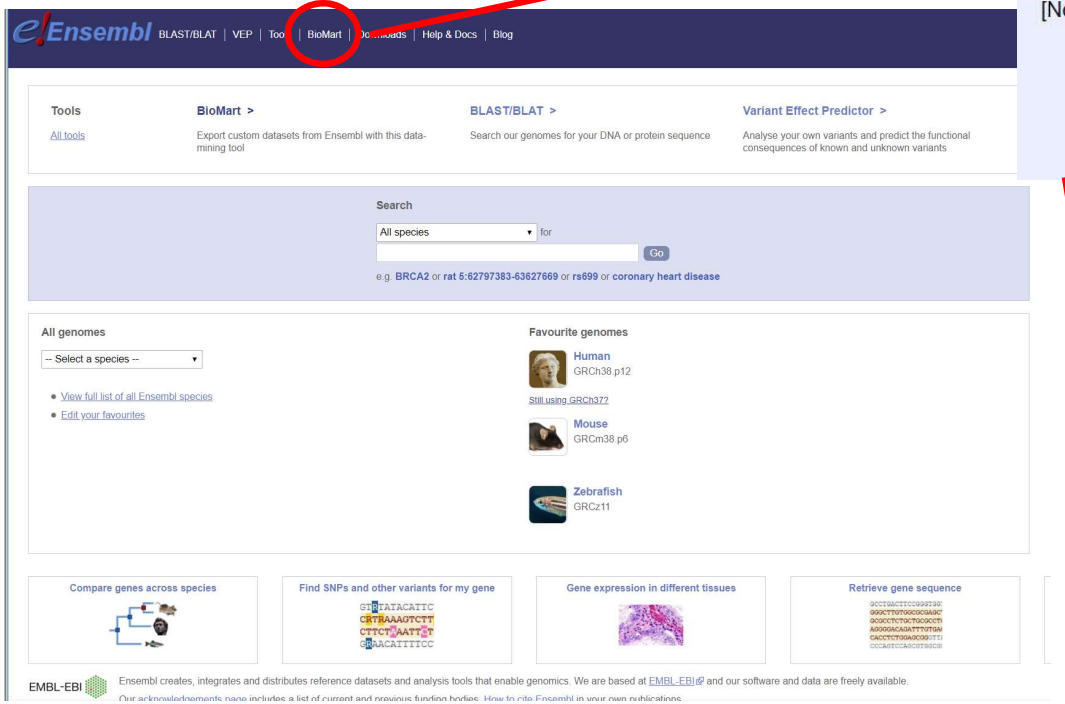
Spring 2019

Gene Annotation

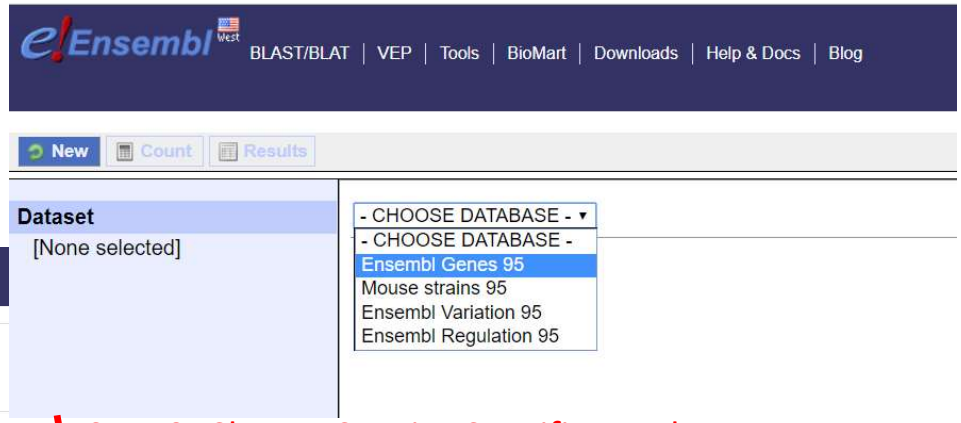
- Many forms of identification
  - Ensembl ID:
    - ENSG# (human gene)
    - ENST# (human transcript)
    - ENSMUSG#/ENSMUST# (mouse gene/transcript)
    - ENSRNOG#/ENSRNOT# (rat gene/transcript)
  - FlyBase ID:
    - FBgn# (fruitfly gene)
    - FBtr# (fruitfly transcript)
  - ZFIN ID:
    - ZDB-GENE-# (zebrafish gene)
    - Alpha numeric combo (transcript)
- RefSeq ID:
  - NM_# mRNA (not species specific)
  - XM_# predicted mRNA
  - NR_# non-coding RNA
- Gene Symbol:
  - Gnb1
  - You really want to include this as this is interpretable
  - Gnb1 symbol for Guanine Nucleotide Binding Protein Subunit Beta-1
- Gene Name:
  - Full on gene name like guanine nucleotide binding protein subunit beta-1

Ensembl BioMart

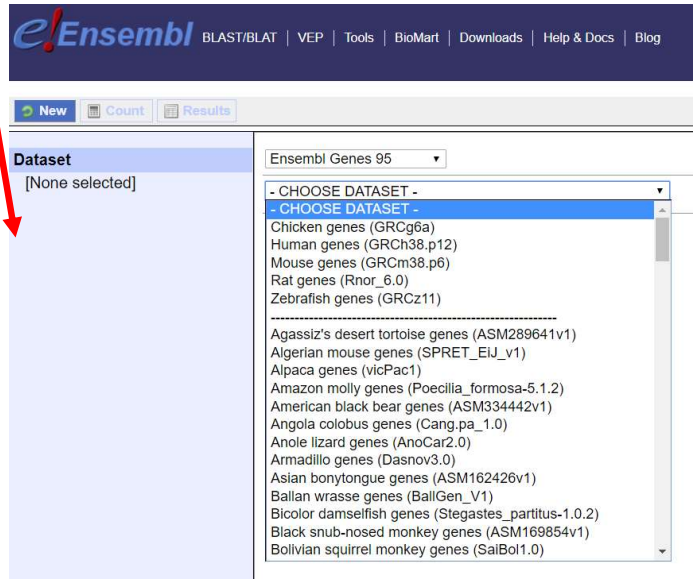
https://www.ensembl.org/



Step 1: Choose Gene Database



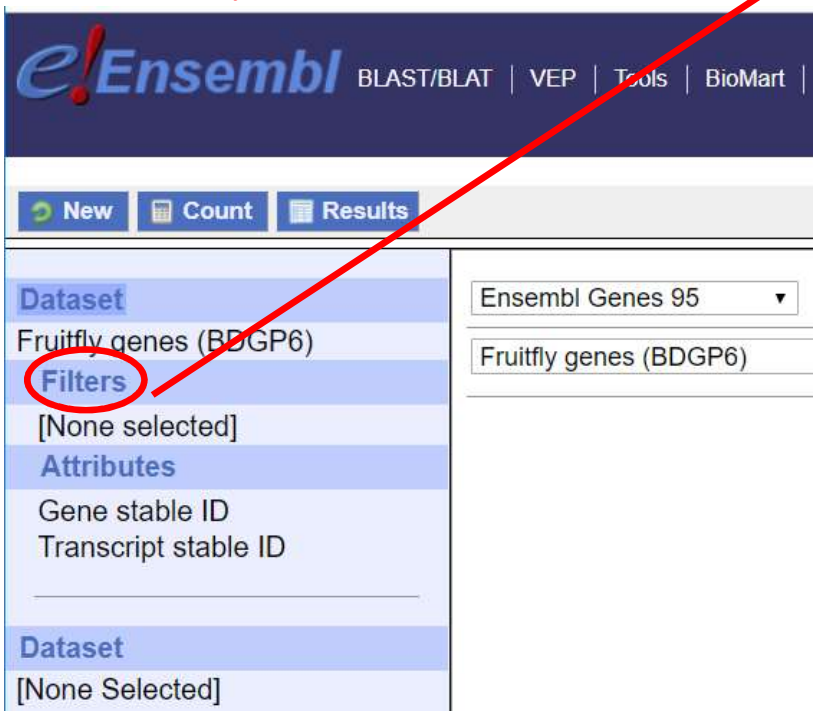
Step 2: Choose Species Specific Database

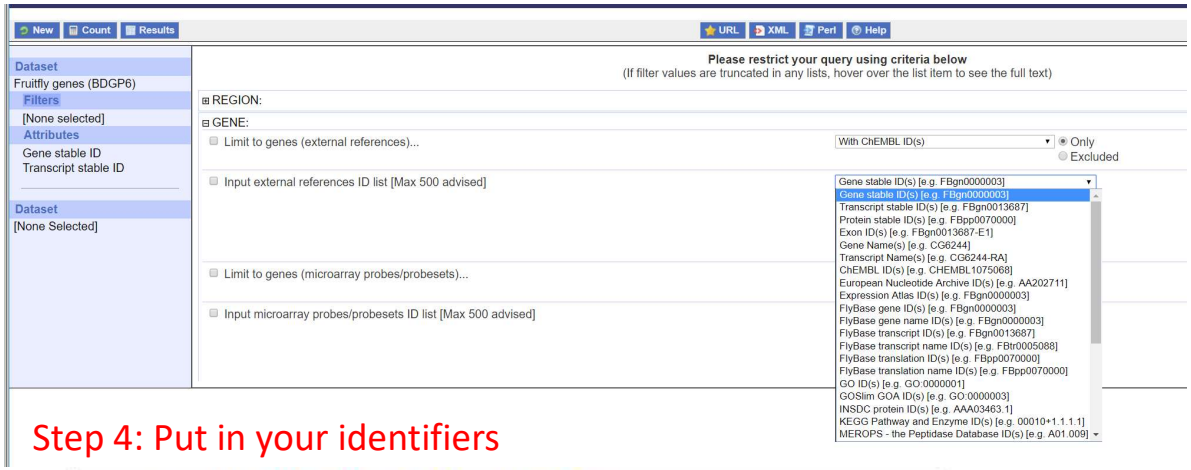


BioMart: Filters Page

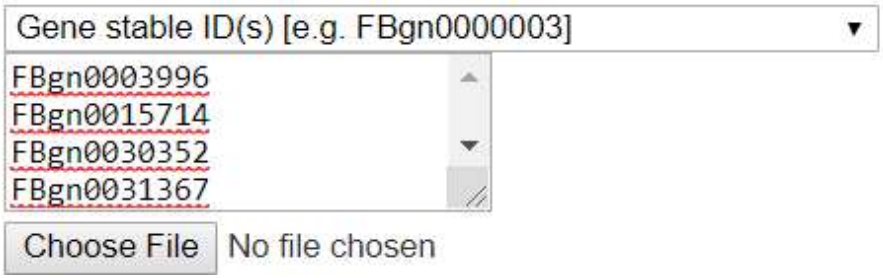
I chose fruitfly

Step 3: Select the Filters Tab



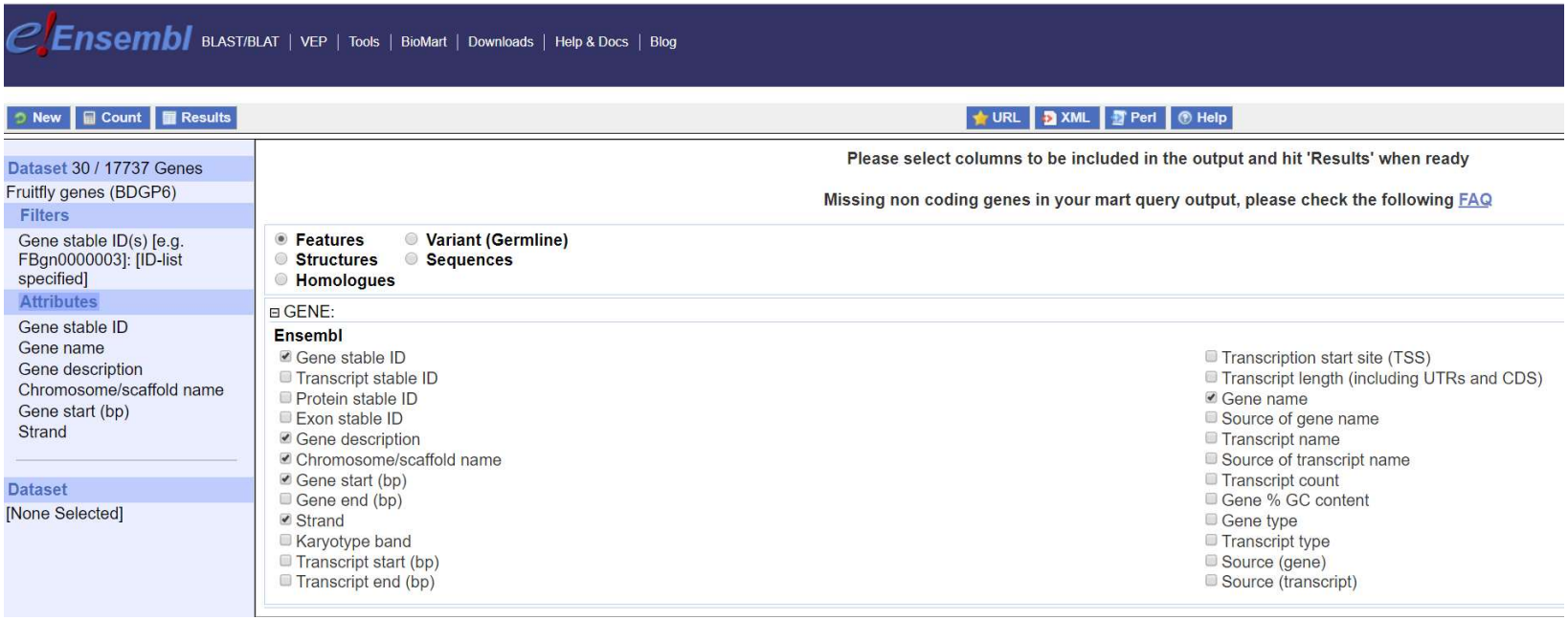


Step 4: Put in your identifiers

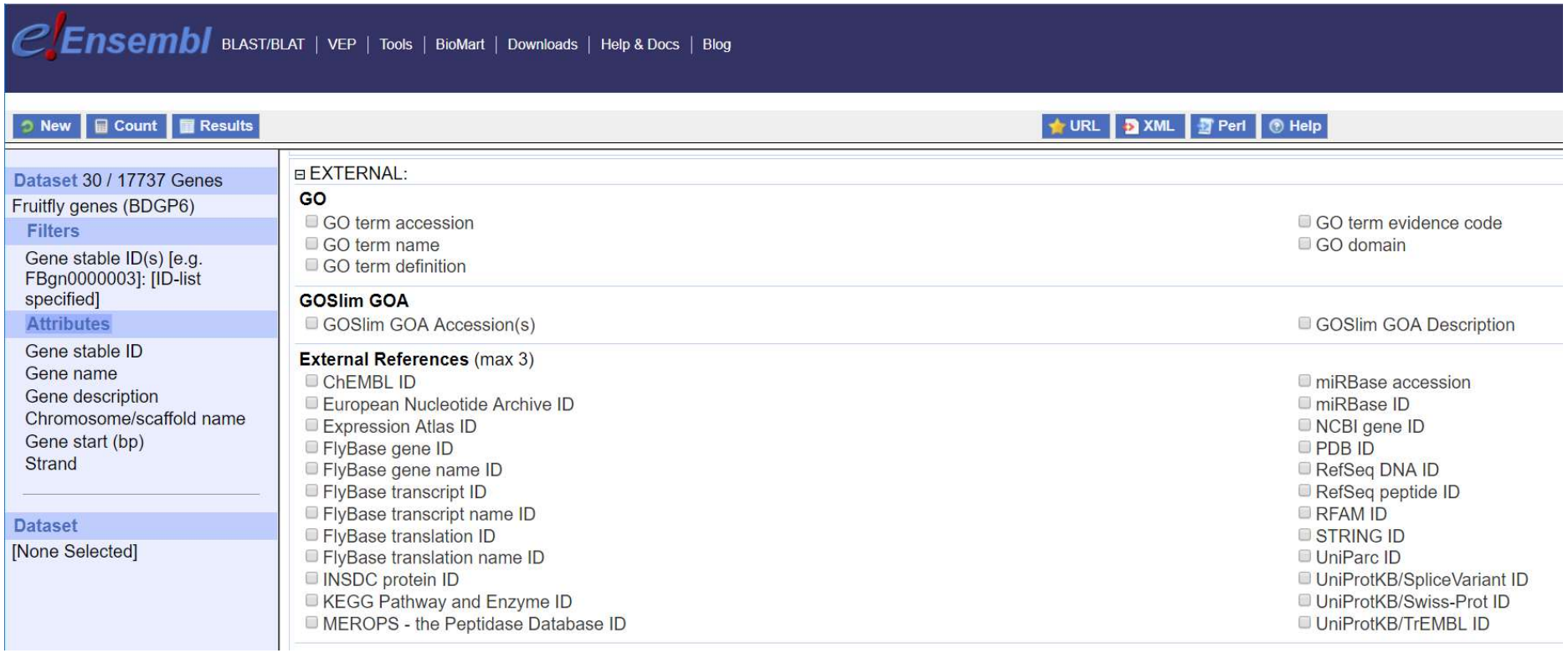


BioMart: Attributes Page

Step 5: Select Attributes You Want

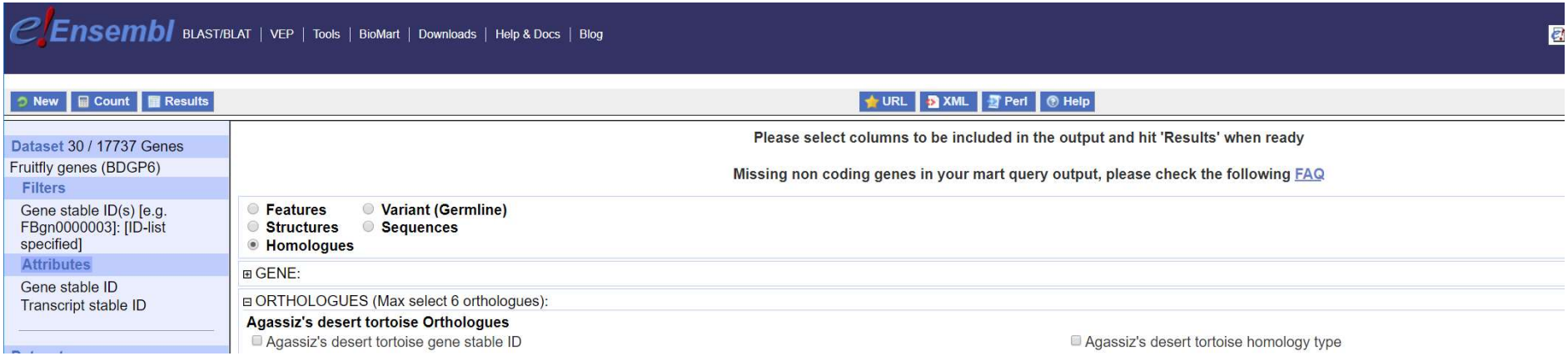


BioMart: Attributes Page Continued



BioMart: Attributes Across Species

Select Homologues on top… Can't select attributes across the Features/Structures/Homologues/Variants/Sequences



Keep scrolling till you get human (or whatever species you want)

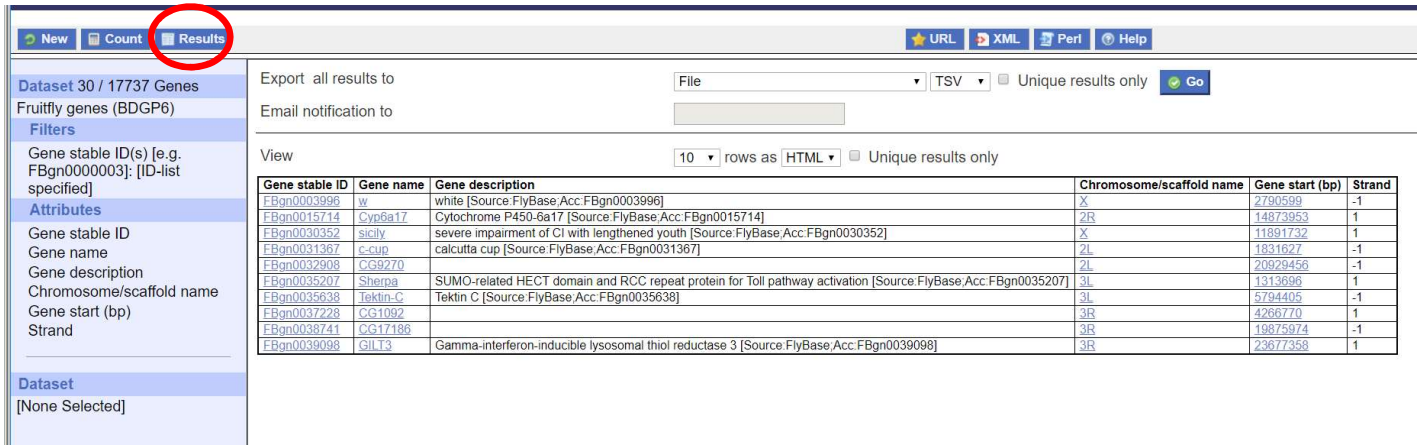
Human Orthologues

- Human gene stable ID
- Human gene name
- Human protein or transcript stable ID
- Human chromosome/scaffold name
- Human chromosome/scaffold start (bp)
- Human chromosome/scaffold end (bp)
- Query protein or transcript ID
- Last common ancestor with Human
- Human homology type
- %id. target Human gene identical to query gene
- %id. query gene identical to target Human gene
- Human Gene-order conservation score
- Human Whole-genome alignment coverage
- dN with Human
- dS with Human
- Human orthology confidence [0 low, 1 high]

BioMart: Feature Results

Selecting count lets you check how many results you have. If you say uploaded 50 IDs and it says here you only have 30, then it's not recognizing 20 of the IDs you loaded.

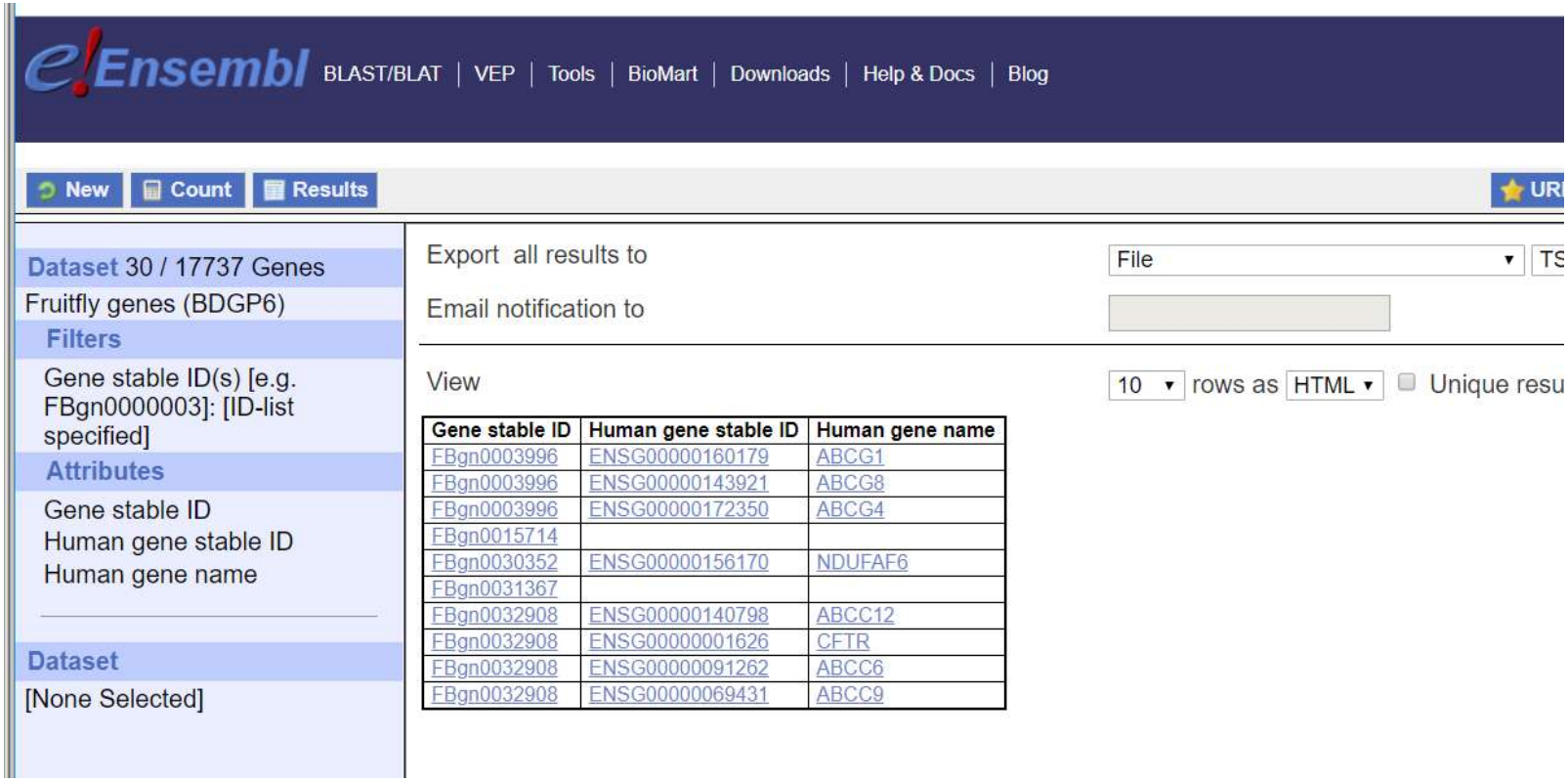
Can export all results as a txt file or view in website



Gene stable ID	Gene name	Gene description	Chromosome/scaffold name	Gene start (bp)	Strand
FBgn0003996	w	white [Source:FlyBase;Acc:FBgn0003996]	X	2790599	-1
FBgn0015714	Cyp6a17	Cytochrome P450-6a17 [Source:FlyBase;Acc:FBgn0015714]	2R	14873953	1
FBgn0030352	sicily	severe impairment of CI with lengthened youth [Source:FlyBase;Acc:FBgn0030352]	X	11891732	1
FBgn0031367	c-cup	calcutta cup [Source:FlyBase;Acc:FBgn0031367]	2L	1831627	-1
FBgn0032908	CG9270		2L	20929456	-1
FBgn0035207	Sherpa	SUMO-related HECT domain and RCC repeat protein for Toll pathway activation [Source:FlyBase;Acc:FBgn0035207]	3L	1313696	1
FBgn0035638	Tektin-C	Tektin C [Source:FlyBase;Acc:FBgn0035638]	3L	5794405	-1
FBgn0037228	CG1092		3R	4266770	1
FBgn0038741	CG17186		3R	19875974	-1
FBgn0039098	GILT3	Gamma-interferon-inducible lysosomal thiol reductase 3 [Source:FlyBase;Acc:FBgn0039098]	3R	23677358	1

In attributes ALWAYS select the identifier you uploaded as well so you can link it back to your statistical results. This example it is "Gene stable ID"

BioMart: Homologue Results



Gene stable ID	Human gene stable ID	Human gene name
FBgn0003996	ENSG00000160179	ABCG1
FBgn0003996	ENSG00000143921	ABCG8
FBgn0003996	ENSG00000172350	ABCG4
FBgn0015714		
FBgn0030352	ENSG00000156170	NDUFAF6
FBgn0031367		
FBgn0032908	ENSG00000140798	ABCC12
FBgn0032908	ENSG00000001626	CFTR
FBgn0032908	ENSG00000091262	ABCC6
FBgn0032908	ENSG00000069431	ABCC9

R/biomaRt

Allows you to access Ensembl's BioMart through R

```
library("biomaRt")
listMarts()
```

```
##               biomart               version
## 1 ENSEMBL_MART_ENSEMBL      Ensembl Genes 94
## 2   ENSEMBL_MART_MOUSE      Mouse strains 94
## 3     ENSEMBL_MART_SNP  Ensembl Variation 94
## 4 ENSEMBL_MART_FUNCGEN Ensembl Regulation 94
```

```
ensembl=useMart("ensembl")
```

```
listDatasets(ensembl)
```

```
##                              dataset
        description
## 1         acalliptera_gene_ensembl                                Eastern happy
genes (fAstCal1.2)
## 2       acarolinensis_gene_ensembl                                  Anole lizar
d genes (AnoCar2.0)
## 3        acitrinellus_gene_ensembl                                  Midas cichl
id genes (Midas_v5)
## 4        amelanoleuca_gene_ensembl                                         Pa
nda genes (ailMel1)
## 5          amexicanus_gene_ensembl                     Cave fish genes (Asty
anax_mexicanus-2.0)
## 6          anancymaae_gene_ensembl                       Ma's night monk
ey genes (Anan_2.0)
## 7          aocellaris_gene_ensembl                     Clown anemonefis
h genes (AmpOce1.0)
```

R/biomaRt

```
filters = listFilters(ensembl)
filters[1:5,]
```

```
##              name              description
## 1 chromosome_name Chromosome/scaffold name
## 2           start                    Start
## 3             end                      End
## 4      band_start               Band Start
## 5        band_end                 Band End
```

```
attributes = listAttributes(ensembl)
attributes[1:5,]
```

```
##                           name                 description         page
## 1              ensembl_gene_id               Gene stable ID feature_page
## 2       ensembl_gene_id_version       Gene stable ID version feature_page
## 3         ensembl_transcript_id         Transcript stable ID feature_page
## 4 ensembl_transcript_id_version Transcript stable ID version feature_page
## 5            ensembl_peptide_id            Protein stable ID feature_page
```

R/biomaRt

```
affyids=c("202763_at","209310_s_at","207500_at")
getBM(attributes=c('affy_hg_u133_plus_2', 'entrezgene'),
      filters = 'affy_hg_u133_plus_2',
      values = affyids,
      mart = ensembl)
```

```
##   affy_hg_u133_plus_2 entrezgene
## 1           202763_at        836
## 2         209310_s_at        837
## 3           207500_at        838
```

OrgDB Objects in R

Packages found under OrgDb:

Rank based on number of downloads: lower numbers are more frequently downloaded.

Show All entries Search table:

Package	Maintainer	Title	Rank
org.Hs.eg.db	Bioconductor Package Maintainer	Genome wide annotation for Human	3
org.Mm.eg.db	Bioconductor Package Maintainer	Genome wide annotation for Mouse	4
org.Rn.eg.db	Bioconductor Package Maintainer	Genome wide annotation for Rat	13
org.Sc.sgd.db	Bioconductor Package Maintainer	Genome wide annotation for Yeast	24
org.Dm.eg.db	Bioconductor Package Maintainer	Genome wide annotation for Fly	26
org.Dr.eg.db	Bioconductor Package Maintainer	Genome wide annotation for Zebrafish	35
org.At.tair.db	Bioconductor Package Maintainer	Genome wide annotation for Arabidopsis	36
org.Bt.eg.db	Bioconductor Package Maintainer	Genome wide annotation for Bovine	45
org.Ce.eg.db	Bioconductor Package Maintainer	Genome wide annotation for Worm	46
org.Gg.eg.db	Bioconductor Package Maintainer	Genome wide annotation for Chicken	48
org.Cf.eg.db	Bioconductor Package Maintainer	Genome wide annotation for Canine	55

Stand alone R packages

Type of S4 object

Meta data this includes is more database annotation links:

- Symbols
- Various IDs
- Names
- KEGG ID
- GO ID

TxDB Objects in R

- Databases it pulls from
  - BioMart (Ensembl)
  - UCSC Genome Bioinformatics
- R/GenomicFeatures
- Backed by SQLite database
- Manages genomic locations (mapping traits)
  - Transcript level information
  - Exons
  - Protein coding sequences
  - Related gene identifier
- Come as their own R packages

```
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene
```

A type of S4 object

Can't really browse this like a dataset

```
txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene
txdb
```

```
## TxDb object:
## # Db type: TxDb
## # Supporting package: GenomicFeatures
## # Data source: UCSC
## # Genome: hg19
## # Organism: Homo sapiens
## # Taxonomy ID: 9606
## # UCSC Table: knownGene
## # Resource URL: http://genome.ucsc.edu/
## # Type of Gene ID: Entrez Gene ID
## # Full dataset: yes
## # miRBase build ID: GRCh37
## # transcript_nrow: 82960
## # exon_nrow: 289969
## # cds_nrow: 237533
## # Db created by: GenomicFeatures package from Bioconductor
## # Creation time: 2015-10-07 18:11:28 +0000 (Wed, 07 Oct 2015)
## # GenomicFeatures version at creation time: 1.21.30
## # RSQLite version at creation time: 1.0.0
## # DBSCHEMAVERSION: 1.1
```

Made into a very large matrix

```
txdb.toBrowse = select(txdb, columns=columns(txdb), keys=keys(txdb), keytype=c("GENEID"))
dim(txdb.toBrowse)
```

```
## [1] 718909     21
```

```
head(txdb.toBrowse)
```

```
##   GENEID  CDSID CDSNAME CDSCHROM CDSSTRAND CDSSTART   CDSEND EXONID
## 1      1 206062    <NA>    chr19         - 58864770 58864803 250818
## 2      1 206061    <NA>    chr19         - 58864658 58864693 250817
## 3      1 206060    <NA>    chr19         - 58864294 58864563 250816
## 4      1 206059    <NA>    chr19         - 58863649 58863921 250815
## 5      1 206058    <NA>    chr19         - 58862757 58863053 250814
## 6      1 206057    <NA>    chr19         - 58861736 58862017 250813
##   EXONNAME EXONCHROM EXONSTRAND EXONSTART  EXONEND  TXID EXONRANK
## 1     <NA>     chr19          -  58864770 58864865 70455        1
## 2     <NA>     chr19          -  58864658 58864693 70455        2
## 3     <NA>     chr19          -  58864294 58864563 70455        3
## 4     <NA>     chr19          -  58863649 58863921 70455        4
## 5     <NA>     chr19          -  58862757 58863053 70455        5
## 6     <NA>     chr19          -  58861736 58862017 70455        6
##       TXNAME TXTYPE TXCHROM TXSTRAND  TXSTART    TXEND
## 1 uc002qsd.4   <NA>   chr19        - 58858172 58864865
## 2 uc002qsd.4   <NA>   chr19        - 58858172 58864865
## 3 uc002qsd.4   <NA>   chr19        - 58858172 58864865
## 4 uc002qsd.4   <NA>   chr19        - 58858172 58864865
## 5 uc002qsd.4   <NA>   chr19        - 58858172 58864865
## 6 uc002qsd.4   <NA>   chr19        - 58858172 58864865
```

https://bioconductor.org/packages/devel/bioc/vignettes/GenomicFeatures/inst/doc/GenomicFeatures.pdf

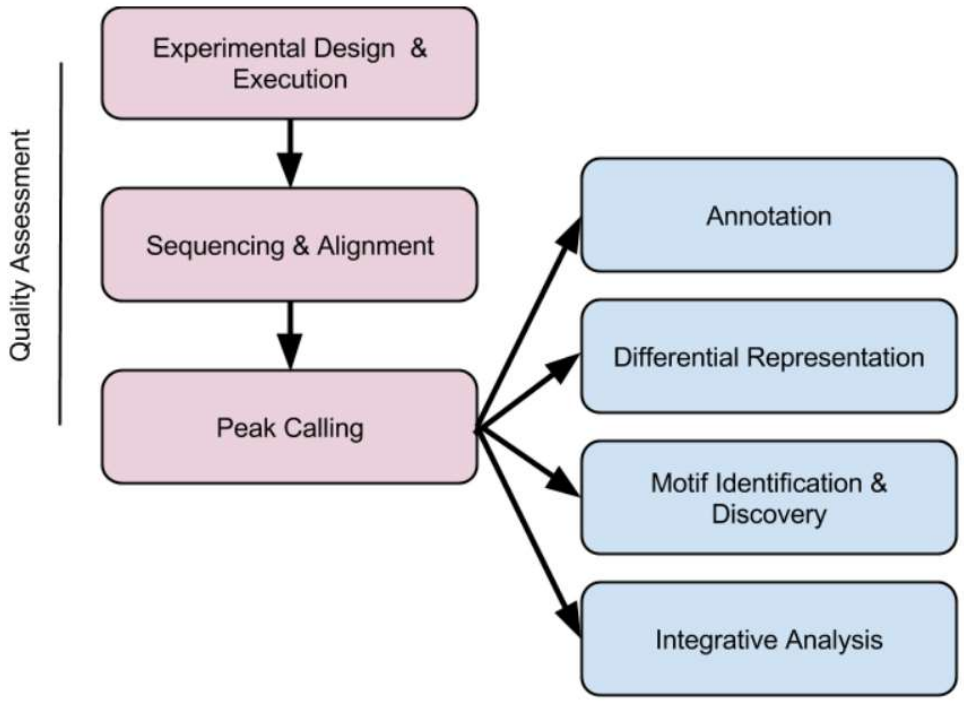
Various Annotation Objects in R

R Class	Description
`OrgDb`	Gene-based information for Homo sapiens; useful for mapping between gene IDs, Names, Symbols, GO and KEGG identifiers, etc.
`TxDb`	Transcriptome ranges for the known gene track of Homo sapiens, e.g., introns, exons, UTR regions.
`OrganismDb`	Collection of multiple annotations for a common organism and genome build.
`BSgenome`	Full genome sequence for Homo sapiens.
`AnnotationHub`	Provides a convenient interface to annotations from many different sources; objects are returned as fully parsed Bioconductor data objects or as the name of a file on disk.

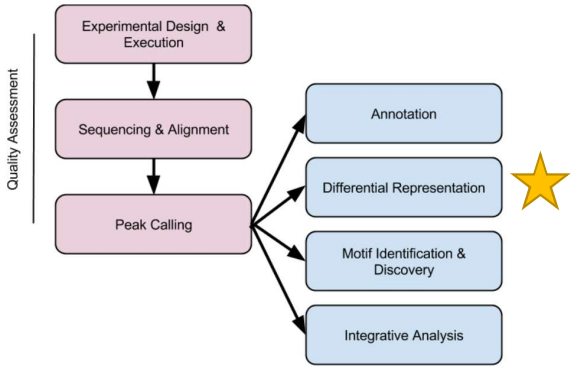
Great blog about getting Ensembl annotations in R:
https://blog.liang2.tw/posts/2016/05/biocondutor-ensembl-reference/

ChIP-Seq Overview From Last Time

- Use chromatin immunoprecipitation to extract DNA fragments which bind to protein of interest
- Make peak calls
- Annotate peak calls



Differential Binding Analysis



- Compare conditions to find differences in binding
  - Treated vs placebo
  - Time points
  - Tissue comparison
- You have inputs/controls for each group (condition you are comparing)
  - Group 1 vs Group 2
  - Group1 vs Control 1 & Group 2 vs Control 2 ← MACS2

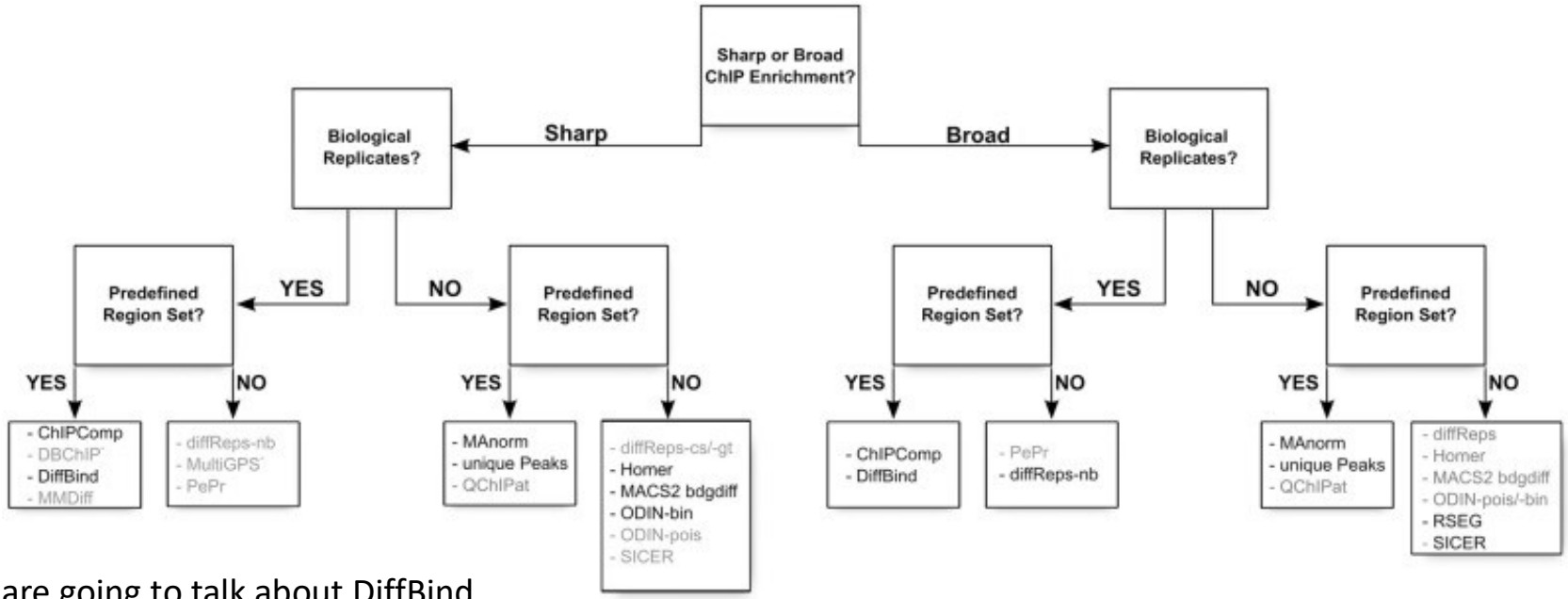
Differential Binding Methods

- Overlapping Peaks
  - Generating peaks separately then compare
  - Fold-change criteria
  - DiffBind, DBChIP (both R packages), MACS2
  - Use RNA-Seq methods
    - DESeq, edgeR, voom
- Other Methods
  - Hidden Markov Models (ChIPDiff)
  - Mixture models on normalized values (DIME)
  - Non-parameteric methods (MMDiff)
  - Count based analysis (MAnorm)

Michael Love (DESeq2 developer) doesn't recommend using RNA-Seq tools for differential binding

Shown that models that use all reads, rather than identify peaks than compare, perform better

Differential Binding Analysis Decision Tree



We are going to talk about DiffBind

Source: Steinhauser et al 2016

R/DiffBind

- First need to have a sample sheet
- Just a csv file where you have with meta data, aligned data and peak output data locations

```
library(DiffBind)
samples <- read.csv(file.path(system.file("extra", package="DiffBind"),"tamoxifen.csv"))
samples
```

```
##    SampleID Tissue Factor  Condition  Treatment Replicate
## 1    BT4741  BT474     ER  Resistant Full-Media         1
## 2    BT4742  BT474     ER  Resistant Full-Media         2
## 3     MCF71   MCF7     ER Responsive Full-Media         1
## 4     MCF72   MCF7     ER Responsive Full-Media         2
## 5     MCF73   MCF7     ER Responsive Full-Media         3
## 6     T47D1   T47D     ER Responsive Full-Media         1
## 7     T47D2   T47D     ER Responsive Full-Media         2
## 8    MCF7r1   MCF7     ER  Resistant Full-Media         1
## 9    MCF7r2   MCF7     ER  Resistant Full-Media         2
## 10    ZR751   ZR75     ER Responsive Full-Media         1
## 11    ZR752   ZR75     ER Responsive Full-Media         2
##                      bamReads ControlID                 bamControl
## 1  reads/Chr18_BT474_ER_1.bam    BT474c reads/Chr18_BT474_input.bam
## 2  reads/Chr18_BT474_ER_2.bam    BT474c reads/Chr18_BT474_input.bam
## 3   reads/Chr18_MCF7_ER_1.bam     MCF7c  reads/Chr18_MCF7_input.bam
## 4   reads/Chr18_MCF7_ER_2.bam     MCF7c  reads/Chr18_MCF7_input.bam
## 5   reads/Chr18_MCF7_ER_3.bam     MCF7c  reads/Chr18_MCF7_input.bam
## 6   reads/Chr18_T47D_ER_1.bam     T47Dc  reads/Chr18_T47D_input.bam
## 7   reads/Chr18_T47D_ER_2.bam     T47Dc  reads/Chr18_T47D_input.bam
## 8   reads/Chr18_TAMR_ER_1.bam     TAMRc  reads/Chr18_TAMR_input.bam
## 9   reads/Chr18_TAMR_ER_2.bam     TAMRc  reads/Chr18_TAMR_input.bam
## 10  reads/Chr18_ZR75_ER_1.bam     ZR75c  reads/Chr18_ZR75_input.bam
## 11  reads/Chr18_ZR75_ER_2.bam     ZR75c  reads/Chr18_ZR75_input.bam
##                       Peaks PeakCaller
## 1  peaks/BT474_ER_1.bed.gz        bed
## 2  peaks/BT474_ER_2.bed.gz        bed
## 3   peaks/MCF7_ER_1.bed.gz        bed
## 4   peaks/MCF7_ER_2.bed.gz        bed
## 5   peaks/MCF7_ER_3.bed.gz        bed
## 6   peaks/T47D_ER_1.bed.gz        bed
## 7   peaks/T47D_ER_2.bed.gz        bed
## 8   peaks/TAMR_ER_1.bed.gz        bed
## 9   peaks/TAMR_ER_2.bed.gz        bed
## 10  peaks/ZR75_ER_1.bed.gz        bed
## 11  peaks/ZR75_ER_2.bed.gz        bed
```

R/DiffBind

1. Create DBA object
- Reading in peaksets
- Finding common peaks
- S4 type of object
- Has meta data

```
#construct a DBA object
tamoxifen <- dba(sampleSheet="C:/Users/vanderll/Documents/R/win-library/3.5/DiffBind/extra/tamoxifen.csv")
```

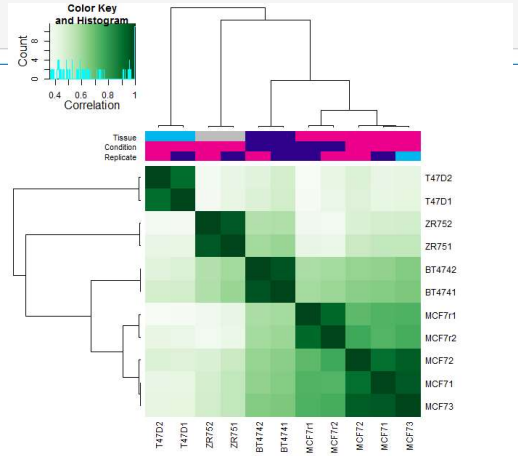
```
tamoxifen
```

Total # Peaks

```
## 11 Samples, 2845 sites in matrix (3795 total):
##        ID Tissue Factor  Condition  Treatment Replicate Caller Intervals
## 1  BT4741  BT474     ER  Resistant Full-Media         1    bed      1080
## 2  BT4742  BT474     ER  Resistant Full-Media         2    bed      1122
## 3   MCF71   MCF7     ER Responsive Full-Media         1    bed      1556
## 4   MCF72   MCF7     ER Responsive Full-Media         2    bed      1046
## 5   MCF73   MCF7     ER Responsive Full-Media         3    bed      1339
## 6   T47D1   T47D     ER Responsive Full-Media         1    bed       527
## 7   T47D2   T47D     ER Responsive Full-Media         2    bed       373
## 8  MCF7r1   MCF7     ER  Resistant Full-Media         1    bed      1438
## 9  MCF7r2   MCF7     ER  Resistant Full-Media         2    bed       930
## 10  ZR751   ZR75     ER Responsive Full-Media         1    bed      2346
## 11  ZR752   ZR75     ER Responsive Full-Media         2    bed      2345
```

Peaks in Each Sample

```
plot(tamoxifen)
```



Peaks Where At Least 2 Samples Overlap

R/DiffBind

2. Counting Reads

- Looking at the peaks where at lease 2 samples overlap
- Look at counts for all peaks (even if not initially identified in sample)

```
tamoxifen <- dba.count(tamoxifen, summits=250)
tamoxifen
```

```
## 11 Samples, 2845 sites in matrix:
##        ID Tissue Factor  Condition  Treatment Replicate Caller Intervals
## 1  BT4741  BT474     ER  Resistant Full-Media         1 counts      2845
## 2  BT4742  BT474     ER  Resistant Full-Media         2 counts      2845
## 3   MCF71   MCF7     ER Responsive Full-Media         1 counts      2845
## 4   MCF72   MCF7     ER Responsive Full-Media         2 counts      2845
## 5   MCF73   MCF7     ER Responsive Full-Media         3 counts      2845
## 6   T47D1   T47D     ER Responsive Full-Media         1 counts      2845
## 7   T47D2   T47D     ER Responsive Full-Media         2 counts      2845
## 8  MCF7r1   MCF7     ER  Resistant Full-Media         1 counts      2845
## 9  MCF7r2   MCF7     ER  Resistant Full-Media         2 counts      2845
## 10  ZR751   ZR75     ER Responsive Full-Media         1 counts      2845
## 11  ZR752   ZR75     ER Responsive Full-Media         2 counts      2845
##    FRiP
## 1  0.16
## 2  0.15
## 3  0.27
## 4  0.17
## 5  0.23
## 6  0.10
## 7  0.06
## 8  0.20
## 9  0.13
## 10 0.32
## 11 0.22
```

All samples quantitated on all peaks

FRiP = Fraction of Reads in Peaks
Proportion of reads that overlap consensus dataset, indicates which samples show more enrichment overall

R/DiffBind

3. Set up model you want to test
4. Perform analysis

- Example is showing we have 629 sites that have differential binding at an FDR < 0.05

```
#testing resistant vs responsive
tamoxifen <- dba.contrast(tamoxifen, categories=DBA_CONDITION)
tamoxifen <- dba.analyze(tamoxifen)
tamoxifen
```

```
## 11 Samples, 2845 sites in matrix:
##        ID Tissue Factor  Condition  Treatment Replicate Caller Intervals
## 1  BT4741  BT474     ER  Resistant Full-Media         1 counts      2845
## 2  BT4742  BT474     ER  Resistant Full-Media         2 counts      2845
## 3   MCF71   MCF7     ER Responsive Full-Media         1 counts      2845
## 4   MCF72   MCF7     ER Responsive Full-Media         2 counts      2845
## 5   MCF73   MCF7     ER Responsive Full-Media         3 counts      2845
## 6   T47D1   T47D     ER Responsive Full-Media         1 counts      2845
## 7   T47D2   T47D     ER Responsive Full-Media         2 counts      2845
## 8  MCF7r1   MCF7     ER  Resistant Full-Media         1 counts      2845
## 9  MCF7r2   MCF7     ER  Resistant Full-Media         2 counts      2845
## 10  ZR751   ZR75     ER Responsive Full-Media         1 counts      2845
## 11  ZR752   ZR75     ER Responsive Full-Media         2 counts      2845
##    FRiP
## 1  0.16
## 2  0.15
## 3  0.27
## 4  0.17
## 5  0.23
## 6  0.10
## 7  0.06
## 8  0.20
## 9  0.13
## 10 0.32
## 11 0.22
##
## 1 Contrast:
##      Group1 Members1     Group2 Members2 DB.DESeq2
## 1 Resistant        4 Responsive        7       629
```

Differential Binding Results at Bottom

R/DiffBind

5. Get Differential Bound Sites

- The results are in a form of a GRanges object
- I would transform to a matrix so you can export as a csv

```
tamoxifen.results <- dba.report(tamoxifen)
tamoxifen.results
```

```
## GRanges object with 629 ranges and 6 metadata columns:
##        seqnames            ranges strand |      Conc Conc_Resistant
##           <Rle>         <IRanges>  <Rle> | <numeric>      <numeric>
##   2452    chr18 64490686-64491186      * |      6.36           1.39
##   1291    chr18 34597713-34598213      * |      5.33           0.22
##    976    chr18 26860997-26861497      * |       7.3           3.13
##   2338    chr18 60892900-60893400      * |      7.13           1.84
##   2077    chr18 55569087-55569587      * |      5.52           1.89
##    ...      ...               ...    ... .       ...            ...
##    551    chr18 14465945-14466445      * |      6.02           4.38
##   2659    chr18 71909888-71910388      * |      5.58           3.73
##   2541    chr18 68007206-68007706      * |      3.61           2.41
##   1967    chr18 52609747-52610247      * |      3.87           2.39
##   2383    chr18 61927095-61927595      * |      1.72          -0.22
##        Conc_Responsive      Fold   p-value       FDR
##              <numeric> <numeric> <numeric> <numeric>
##   2452              7     -5.61  3.57e-10  1.02e-06
##   1291           5.97     -5.75   1.1e-09  1.57e-06
##    976           7.92     -4.79   1.1e-08  1.05e-05
##   2338           7.77     -5.93  1.68e-08  1.17e-05
##   2077           6.13     -4.23  2.36e-08  1.17e-05
##    ...            ...       ...       ...       ...
##    551           6.49     -2.11    0.0108     0.049
##   2659           6.07     -2.34    0.0108    0.0491
##   2541           4.01      -1.6    0.0109    0.0495
##   1967           4.32     -1.94    0.0109    0.0495
##   2383           2.22     -2.44     0.011    0.0498
##   -------
##   seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

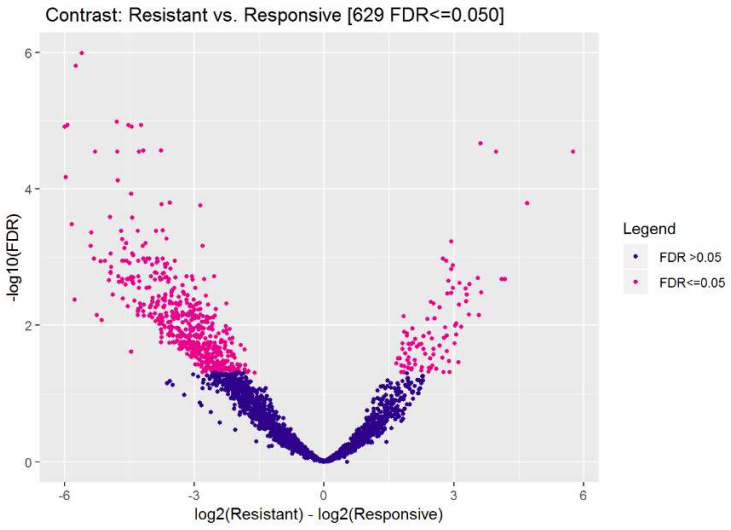
```
forExport = as.data.frame(tamoxifen.results)
dim(forExport)
```

```
## [1] 629  11
```

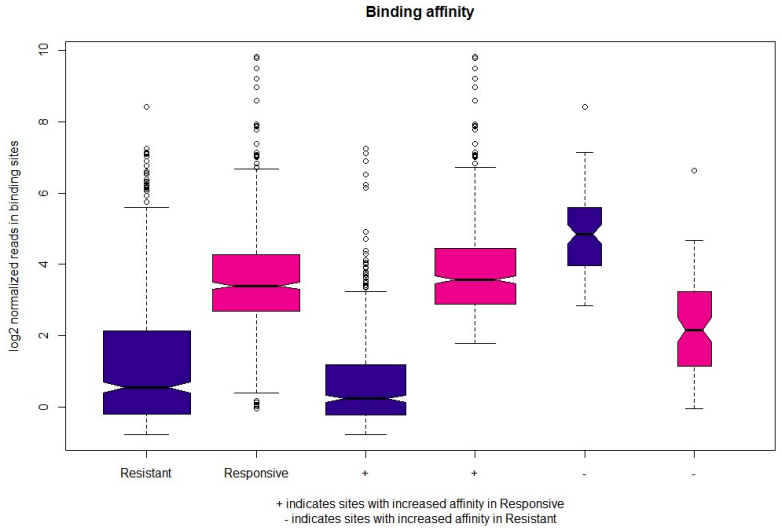
R/BindDiff

6. Visualize Results

```
dba.plotVolcano(tamoxifen)
```

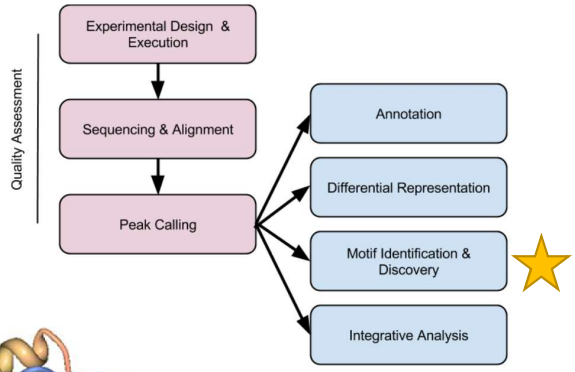


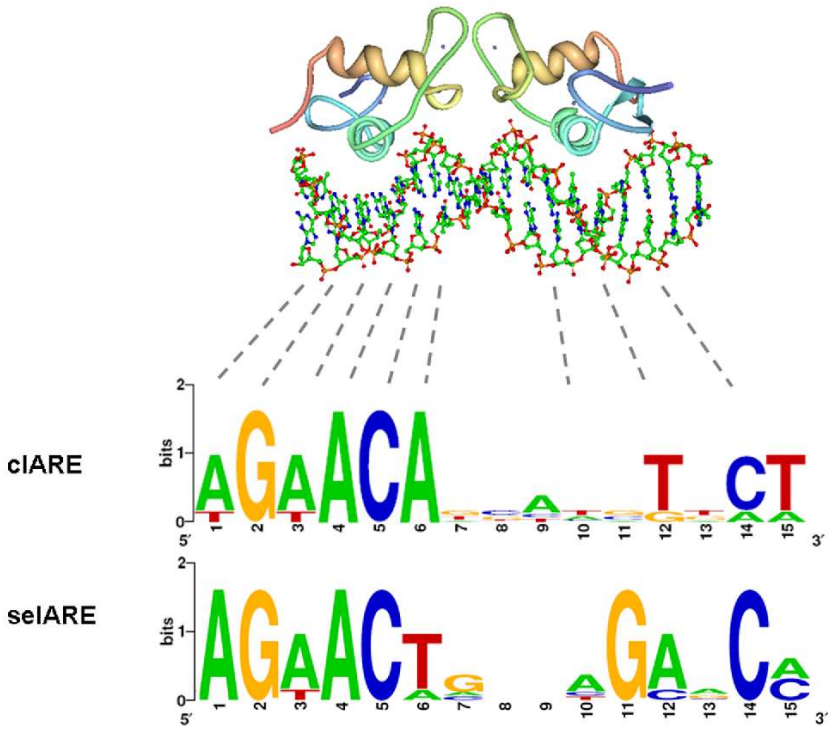

```
dba.plotBox(tamoxifen)
```



Motif Discovery

- For transcription factors they bind to specific motifs
- Want to identify what motifs are enriched based on the peaks you identified
- In the example on right, the cIARE TF has a high affinity for sequences AGAACA and the seIARE looks to have a similar binding motif except T at the end instead of A
- Calculates the positional distribution of oligonucleotides in a set of sequences, and detects those which significantly discard from a homogeneous distribution





Source: Claessens et al, 2008

FASTA file format

- All motif discovery programs need a fasta file
- This helps you go from a bed format (chr and location of peaks) and extracts out those sequences in that location
- Bedtools
- R/hoardeR

```
getFastaFromBed(bed, species=NULL, assembly = NULL, fastaFolder=NULL,
                verbose=TRUE, export=NULL, fileName=NULL)
```

bedtools

Table of Contents

getfasta
- Usage and option summary
- Default behavior
- `-name` Using the BED "nam
- `-tab` Creating a tab-delimit
- `-bedOut` Creating a tab-deli
- `-s` Forcing the extracted se
- `-split` Extracting BED "blo

Previous topic
genomecov

Next topic
groupby

This Page
Show Source

Quick search
Go

getfasta

`bedtools getfasta` extracts sequences from a FASTA file for each of the intervals defined in a BED/GFF/VCF file.

Tip

1. The headers in the input FASTA file must *exactly* match the chromosome column in the BED file.
2. You can use the UNIX `fold` command to set the line width of the FASTA output. For example, `fold -w 60` will make each line of the FASTA file have at most 60 nucleotides for easy viewing.
3. BED files containing a single region require a newline character at the end of the line, otherwise a blank output file is produced.

See also

maskfasta

Usage and option summary

Usage

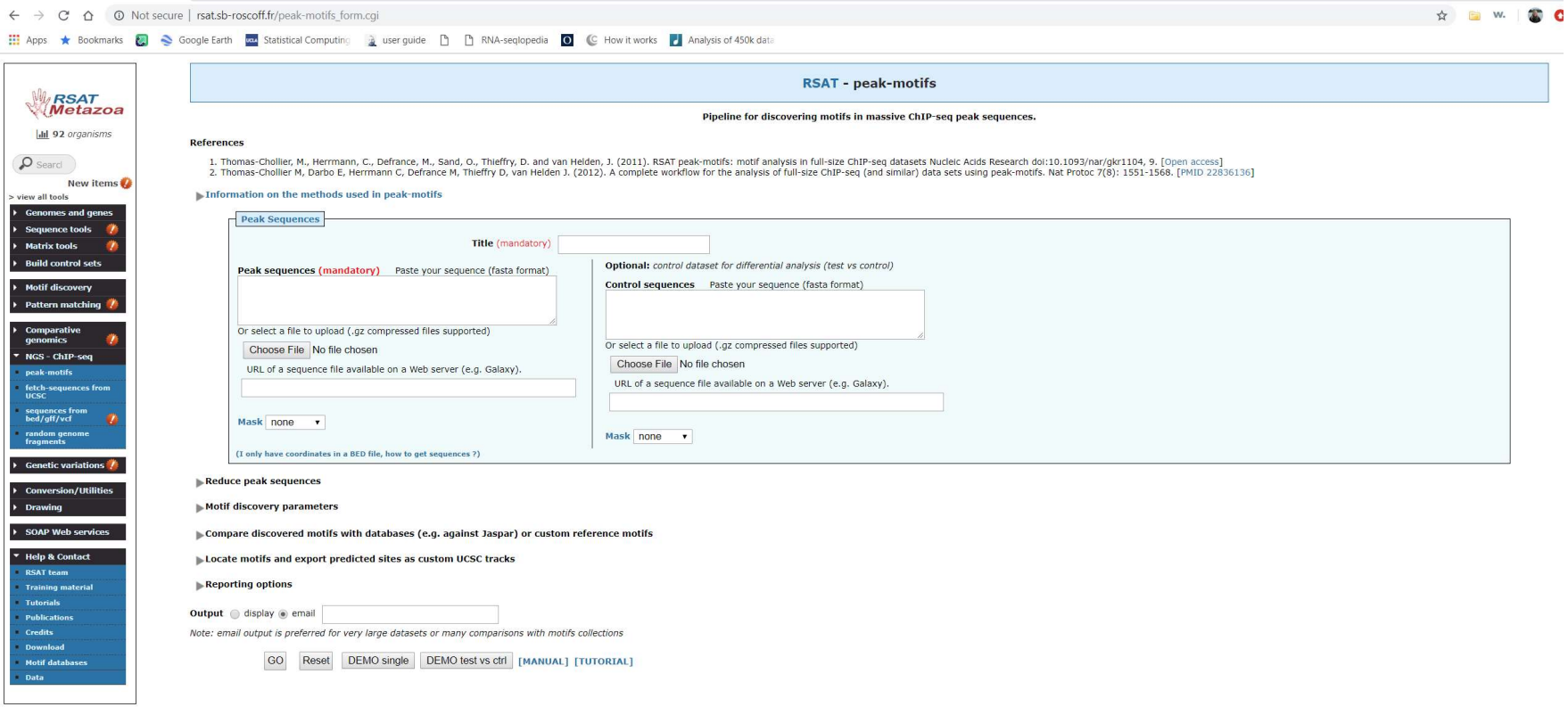
```
$ bedtools getfasta [OPTIONS] -fi <input FASTA> -bed <BED/GFF/VCF>
```

(or):

```
$ getFastaFromBed [OPTIONS] -fi <input FASTA> -bed <BED/GFF/VCF>
```

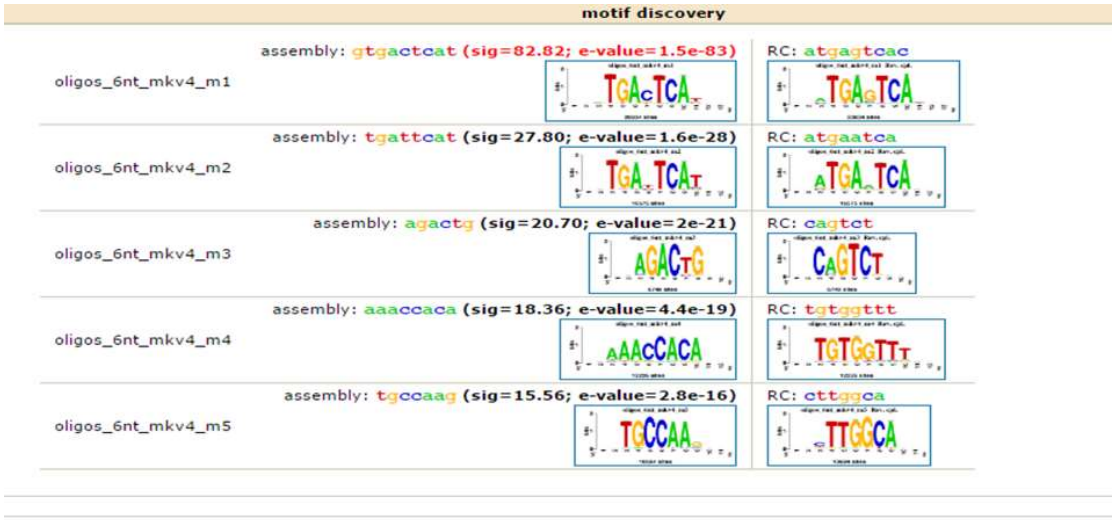
RSAT – Regulatory Sequence Analysis Tools

http://rsat.sb-roscoff.fr/peak-motifs_form.cgi



RSAT options & output

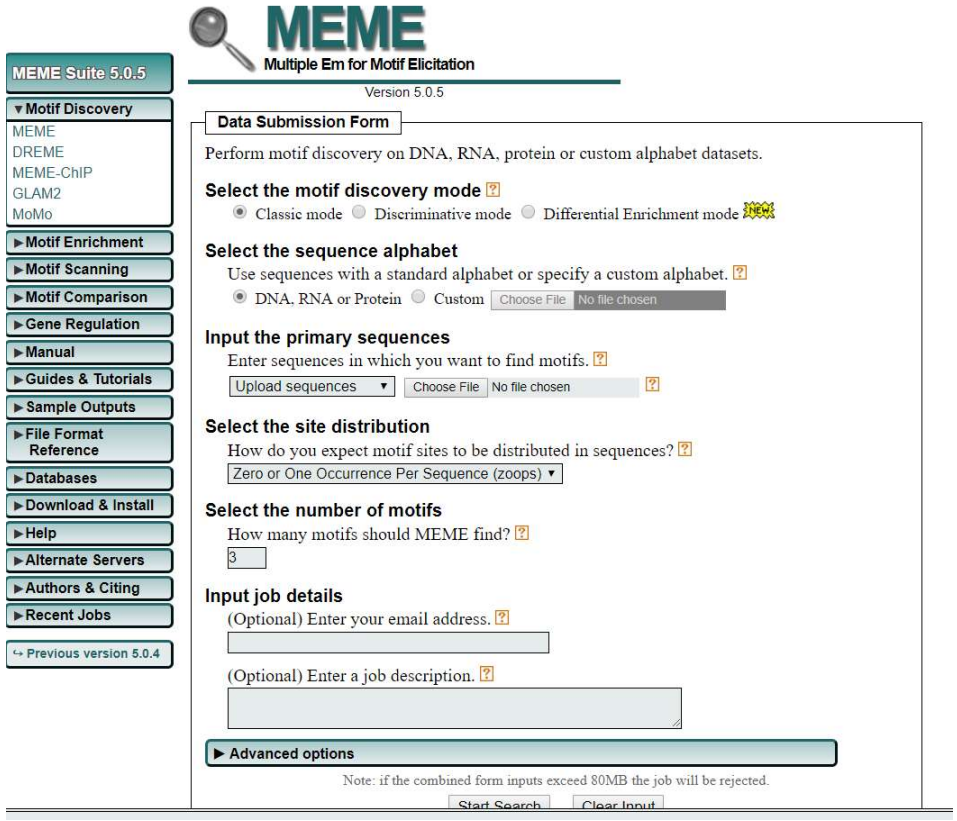
- ***oligo-analysis***: detection of over-represented oligonucleotides (words)
- ***dyad-analysis***: detection of over-represented spaced pairs of oligonucleotides
- ***position-analysis***: detection of words having a positional bias in sequences aligned on some reference position



MEME-ChIP

http://meme-suite.org/doc/meme-chip.html

MEME	Discovers novel, ungapped motifs (recurring fixed-length)
DREME	Discovers short, ungapped motifs (recurring fixed-length) that are relatively enriched compared to shuffled sequence
MEME-ChIP	Comprehensive motif analysis on large sets of sequences.
GLAM2	Discovers novel gapped motifs (recurring, variable length) in DNA or protein sequences
MoMo	Discovers motifs with different post translation modifications (amino acid motifs)



HOMER

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:

R/BCRANK

- R package to identify peaks
- Really easy to do
- Not used often

- http://bioconductor.org/packages/devel/bioc/vignettes/systemPipeR/inst/doc/systemPipeChIPseq.html#6_peak_calling_with_macs2

```
> set.seed(0)
> BCRANKout <- bcrank(fastaFile, restarts=25, use.P1=TRUE, use.P2=TRUE)
```

Since it takes some time to run the algorithm, results can instead be loaded from a p
a larger USF1 data set containing the top 5211 regions:

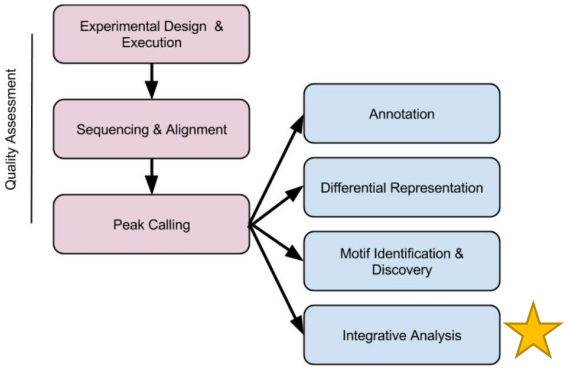
```
> data(BCRANKout)
```

2.3 BCRANK output

An object of type BCRANKresult is returned:

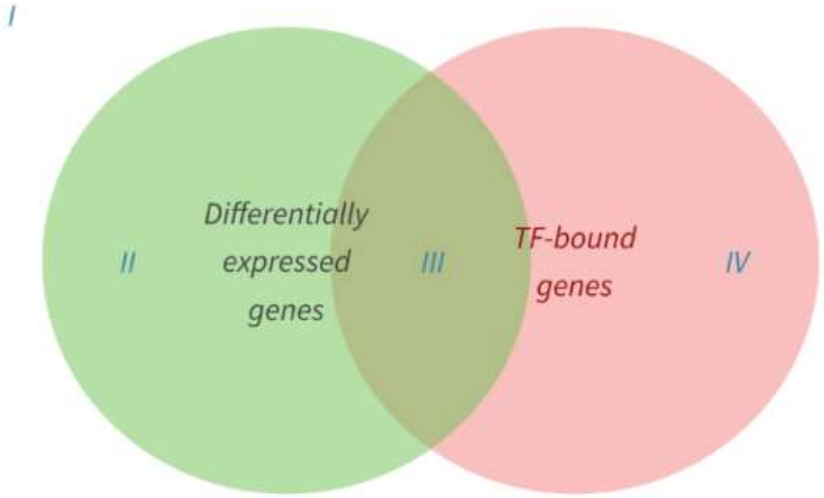
```
> BCRANKout

An object of class "BCRANKresult"

Top 25 DNA motifs predicted by BCRANK:

     Consensus     Score
1     GTCACGTG 316.34270
2     CACGTGAC 304.59499
3       CGCGGA 147.04100
4       GCGAST 135.22207
5    AHATAATAA 128.92440
6     GCGGNGCG 121.87198
7    TNCDGGGCG 119.76454
8    GCGGGGVNG 119.65945
9    CCGCGNTBY 118.77481
10   GDGCGGHGH 115.19679
11   TNCGCGNDG 112.95972
12   CGGGNGMGC 111.93052
```

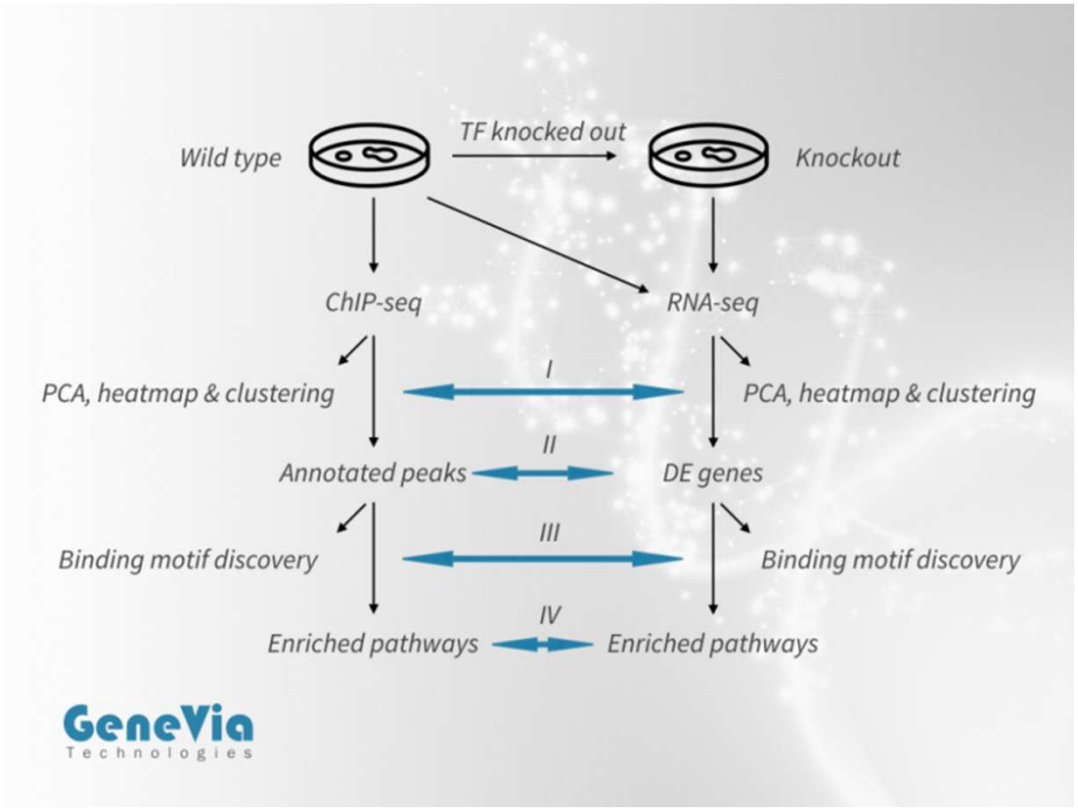
Integrative Analysis



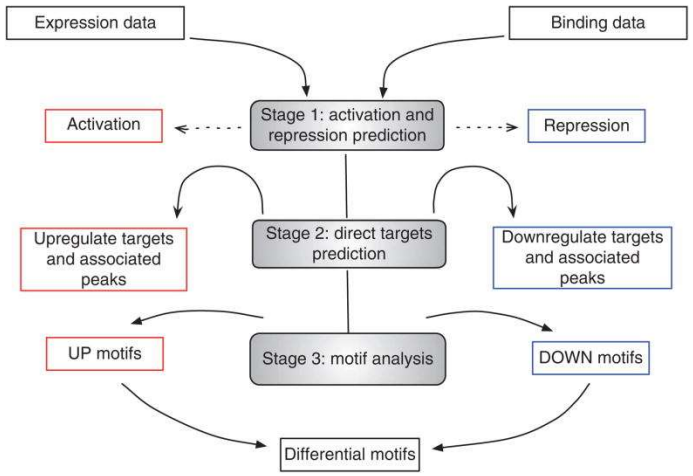
- Performing ChIP-seq, you would be interested in the regulation of gene transcription
- Most often paired with gene expression data
- Do the gene that my transcription factor target actually result in differential expression?

ChIP-Seq RNA-Seq integration

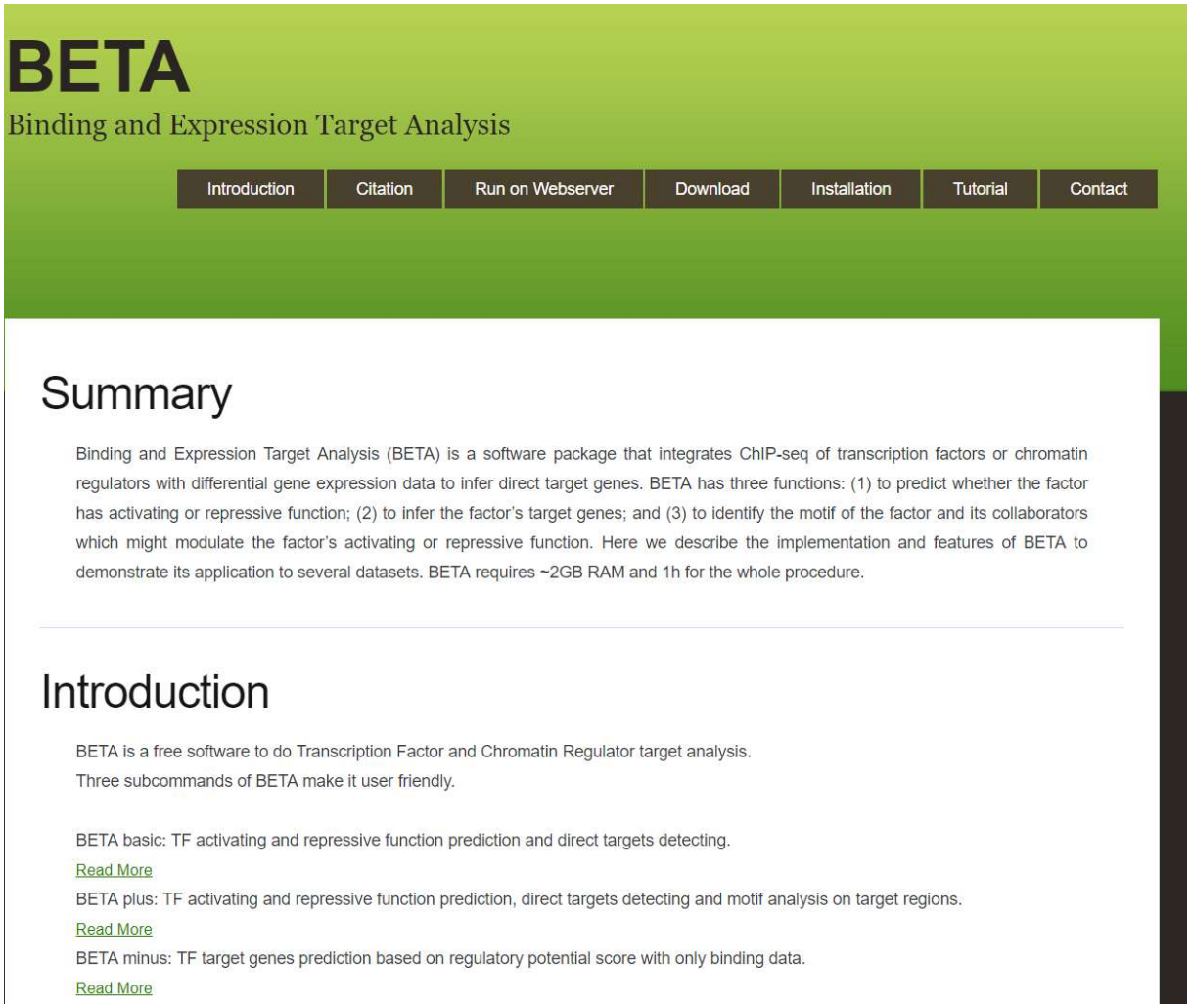




BETA



Source: Wang et al, 2013



Enrichment Analysis

- Can look at nearest gene (within a certain distance to TSS) and use programs we have talked about previously
- R/Chip-Enrich
  - Histones: broadenrich()
  - Transcription Factors: chipenrich()
- ChIP-Enrich is designed for use with 1,000s or 10,000s of narrow peaks which results in fewer gene loci containing a peak overall
- peak ~ GO + s(log10_length)
  - GO is a binary vector indicating whether a gene is in the gene set being tested,
  - peak is a binary vector indicating the presence of a peak in a gene
  - s(log10_length) is a binomial cubic smoothing spline which adjusts for the relationship between the presence of a peak and locus length

ChIP-Enrich Code

- Peaks is either a BED file or a data.frame with the bed format

```
chipenrich(peaks, out_name = "chipenrich", out_path = getwd(), genome = "hg19",
genesets = c("GOBP","GOCC","GOMF"),
locusdef = "nearest_tss", method = "chipenrich",
fisher_alt = "two.sided", use_mappability = F, mappa_file = NULL, read_length = 36,
qc_plots = T, max_geneset_size = 2000, num_peak_threshold = 1, n_cores=1)
```

Results = chipenrich()
Results$results

```
##    Geneset.Type Geneset.ID Description      P.value          FDR
## 1 user-supplied GO:0034660  GO:0034660 5.435777e-05 0.0004529814
## 2 user-supplied GO:0007346  GO:0007346 8.592104e-05 0.0005370065
## 3 user-supplied GO:0031400  GO:0031400 1.164884e-03 0.0058244176
## 4 user-supplied GO:0009314  GO:0009314 2.166075e-02 0.0676898435
## 5 user-supplied GO:0051129  GO:0051129 1.196240e-01 0.2718727018
```

References

- Steinhauser S, Kurzawa N, Eils R, Herrmann C. *A comprehensive comparison of tools for differential ChIP-seq analysis.* Brief Bioinform. **2016** Nov;17(6):953-966.
- Claessens F, et al. (2008) Diverse roles of androgen receptor (AR) domains in ARmediated signaling. Nucl Recept Signal 6:e008
- Wang, S., Sun, H., Ma, J., Zang, C., Wang, C., Wang, J., ... & Liu, X. S. (2013). Target analysis by integration of transcriptome and ChIP-seq data with BETA. ***Nature protocols***, 8(12), 2502-2515.