

# 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 gene name like guanine nucleotide binding protein subunit beta-1

# Ensembl BioMart

<https://www.ensembl.org/>

The screenshot shows the main Ensembl website with a focus on the BioMart section. A red arrow highlights the 'BioMart' link in the top navigation bar. Below it, the BioMart interface is shown with a search bar, a dropdown menu for species selection, and several functional links for genome comparison, SNP finding, gene expression, and sequence retrieval.

## Step 1: Choose Gene Database

This screenshot shows the 'Dataset' selection step. A dropdown menu titled 'CHOOSE DATABASE' is open, listing four options: Ensembl Genes 95, Mouse strains 95, Ensembl Variation 95, and Ensembl Regulation 95. The 'Ensembl Genes 95' option is highlighted with a blue background.

## Step 2: Choose Species Specific Database

This screenshot shows the 'Dataset' selection step for Ensembl Genes 95. A dropdown menu titled 'CHOOSE DATASET' is open, listing five options: Chicken genes (GRCg6a), Human genes (GRCh38.p12), Mouse genes (GRCm38.p6), Rat genes (Rnor\_6.0), and Zebrafish genes (GRCz11). The 'Chicken genes (GRCg6a)' option is highlighted with a blue background.

# BioMart: Filters Page

I chose fruitfly

The screenshot shows the Ensembl BioMart homepage. At the top, there are links for BLAST/BLAT, VEP, Tools, and BioMart. Below these are buttons for New, Count, and Results. On the left, a sidebar has sections for Dataset (selected 'Fruitfly genes (BDGP6)'), Filters (circled in red), Attributes (Gene stable ID, Transcript stable ID), and another Dataset section (None Selected). The main area shows 'Ensembl Genes 95' selected and 'Fruitfly genes (BDGP6)' listed below it.

Step 3: Select the Filters Tab

The screenshot shows the BioMart Filters page. A red arrow points from the 'Filters' tab on the Ensembl page to this screen. The page has tabs for New, Count, and Results at the top right. It includes a note to restrict queries using criteria below. The left sidebar shows 'Dataset' (Fruitfly genes (BDGP6)), 'Filters' (selected), and 'Attributes' (Gene stable ID, Transcript stable ID). The main area has sections for REGION, GENE, and PROBESETS. A dropdown menu for 'Gene stable ID(s)' lists various identifiers. Below is a text input field for 'Gene stable ID(s) [e.g. FBgn0000003]' containing 'FBgn0003996', 'FBgn0015714', 'FBgn0030352', and 'FBgn0031367'. There is also a 'Choose File' button with 'No file chosen'.

Step 4: Put in your identifiers

# BioMart: Attributes Page

## Step 5: Select Attributes You Want

Please select columns to be included in the output and hit 'Results' when ready

Missing non coding genes in your mart query output, please check the following [FAQ](#)

Features    Variant (Germline)  
 Structures    Sequences  
 Homologues

GENE:

**Ensembl**

Gene stable ID  
 Transcript stable ID  
 Protein stable ID  
 Exon stable ID  
 Gene description  
 Chromosome/scaffold name  
 Gene start (bp)  
 Gene end (bp)  
 Strand  
 Karyotype band  
 Transcript start (bp)  
 Transcript end (bp)

Transcription start site (TSS)  
 Transcript length (including UTRs and CDS)  
 Gene name  
 Source of gene name  
 Transcript name  
 Source of transcript name  
 Transcript count  
 Gene % GC content  
 Gene type  
 Transcript type  
 Source (gene)  
 Source (transcript)

# BioMart: Attributes Page Continued

The screenshot shows the Ensembl BioMart interface. The top navigation bar includes links for BLAST/BLAT, VEP, Tools, BioMart, Downloads, Help & Docs, and Blog. Below the navigation is a toolbar with New, Count, Results, URL, XML, Perl, and Help buttons. On the left, a sidebar lists the dataset (30 / 17737 Genes), filters (Fruitfly genes (BDGP6)), gene stable ID(s) (e.g., FBgn0000003), and attributes (Gene stable ID, Gene name, Gene description, Chromosome/scaffold name, Gene start (bp), Strand). The main content area displays external datasets like GO, GOSlim GOA, and various external references such as ChEMBL ID, European Nucleotide Archive ID, and FlyBase IDs. A legend on the right maps symbols to specific database identifiers.

Symbol	Description
<input type="checkbox"/>	GO term accession
<input type="checkbox"/>	GO term name
<input type="checkbox"/>	GO term definition
<input type="checkbox"/>	GO term evidence code
<input type="checkbox"/>	GO domain
<input type="checkbox"/>	GOSlim GOA Description
<input type="checkbox"/>	miRBase accession
<input type="checkbox"/>	miRBase ID
<input type="checkbox"/>	NCBI gene ID
<input type="checkbox"/>	PDB ID
<input type="checkbox"/>	RefSeq DNA ID
<input type="checkbox"/>	RefSeq peptide ID
<input type="checkbox"/>	RFAM ID
<input type="checkbox"/>	STRING ID
<input type="checkbox"/>	UniParc ID
<input type="checkbox"/>	UniProtKB/SpliceVariant ID
<input type="checkbox"/>	UniProtKB/Swiss-Prot ID
<input type="checkbox"/>	UniProtKB/TrEMBL ID

# BioMart: Attributes Across Species

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

The screenshot shows the Ensembl BioMart interface. On the left, there's a sidebar with 'Dataset 30 / 17737 Genes' (Fruitfly genes (BDGP6)), 'Filters' (Gene stable ID(s) [e.g. FBgn0000003]: [ID-list specified]), and 'Attributes' (Gene stable ID, Transcript stable ID). The main area has tabs for 'New', 'Count', and 'Results'. It displays a message: 'Please select columns to be included in the output and hit 'Results' when ready'. Below this, it says 'Missing non coding genes in your mart query output, please check the following [FAQ](#)'. There are several radio button groups: 'Features', 'Variant (Germline)', 'Structures', 'Sequences', and 'Homologues' (which is selected). Under 'Attributes', there are checkboxes for 'GENE', 'ORTHOLOGUES (Max select 6 orthologues): Agassiz's desert tortoise Orthologues', and 'Agassiz's desert tortoise homology type'. A note at the bottom right says 'Agassiz's desert tortoise gene stable ID'.

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

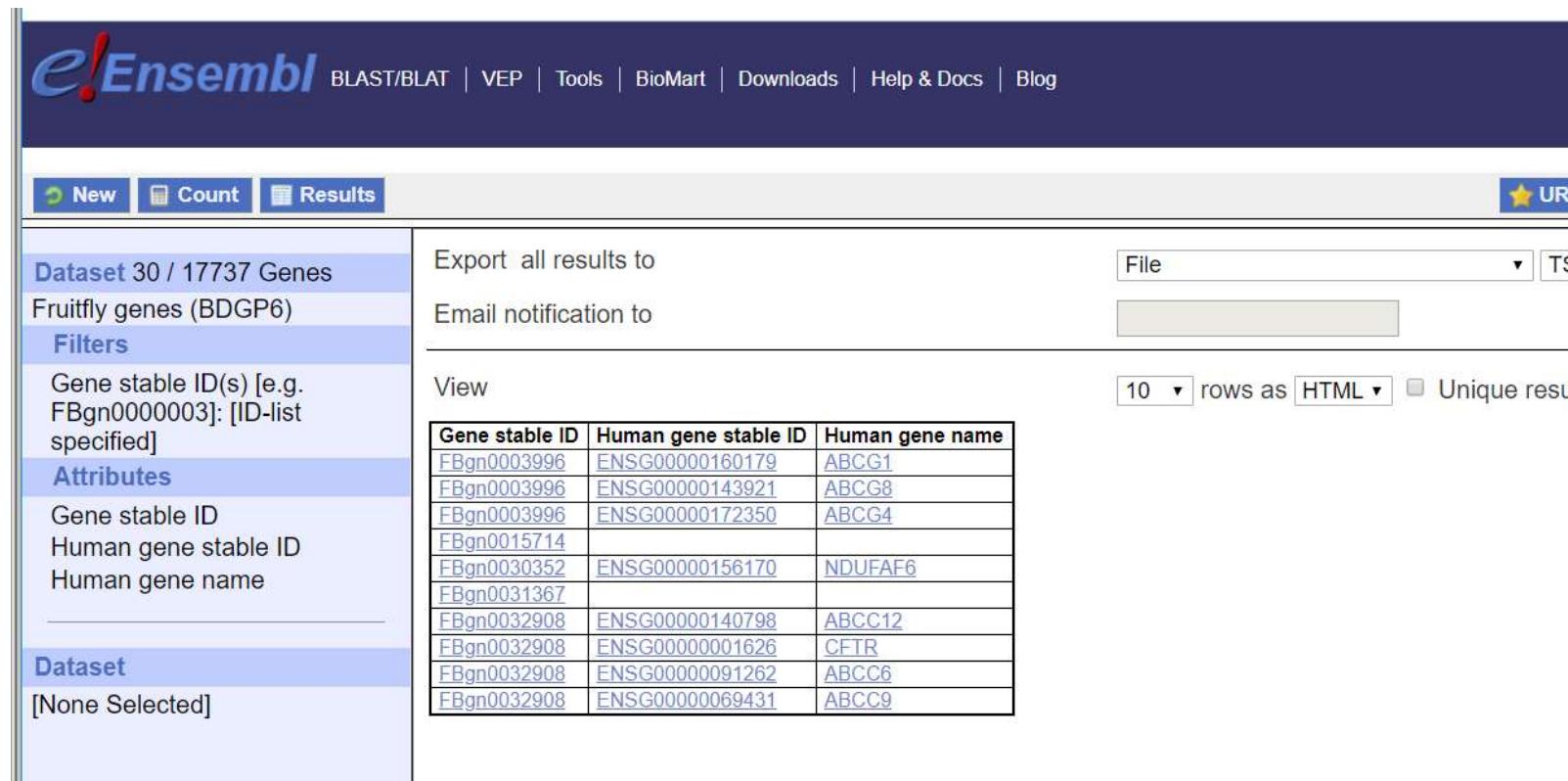
The screenshot shows the Ensembl BioMart homepage. At the top, there's a navigation bar with 'e!Ensembl' logo, 'BLAST/BLAT', and other links. Below it is a search bar with 'New', 'Count' (which is circled in red), and 'Results'. A large blue header box displays 'Dataset 30 / 17737 Genes'. On the left, there's a sidebar with 'Attributes' (Gene stable ID, Gene name, Gene description, Chromosome/scaffold name, Gene start (bp), Strand) and a 'Dataset' section ('[None Selected]').

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.

The screenshot shows the BioMart results page for Dataset 30. At the top, there are buttons for 'New', 'Count' (circled in red), and 'Results'. Below that is an 'Export' section with options for 'File' (TSV), 'Email notification', and 'View' (rows as HTML). A table lists gene data with columns: Gene stable ID, Gene name, Gene description, Chromosome/scaffold name, Gene start (bp), and Strand. The table contains 10 rows of data, such as FBgn0003996 (w, white), FBgn0015714 (Cyp6a17, Cytochrome P450-6a17), and FBgn0030352 (sicily, severe impairment of CI with lengthened youth).

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



The screenshot shows the Ensembl BioMart interface. The top navigation bar includes links for BLAST/BLAT, VEP, Tools, BioMart, Downloads, Help & Docs, and Blog. Below the navigation is a toolbar with New, Count, and Results buttons, and a UR (Uniform Resource) link. On the left, a sidebar displays the dataset information: "Dataset 30 / 17737 Genes" and "Fruitfly genes (BDGP6)". It also contains sections for Filters (Gene stable ID(s)), Attributes (Gene stable ID, Human gene stable ID, Human gene name), and Dataset (None Selected). The main content area shows results for homologous genes. At the top of the results table are options to export all results to a file or email notifications. Below these are settings for viewing: "10 rows as HTML" and a checkbox for "Unique results". The results table has columns for Gene stable ID, Human gene stable ID, and Human gene name. The data is as follows:

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	acalyptrata_gene_ensembl	Eastern happy
	genes (fAstcall1.2)	
## 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 (ailMe1)	
## 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.

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

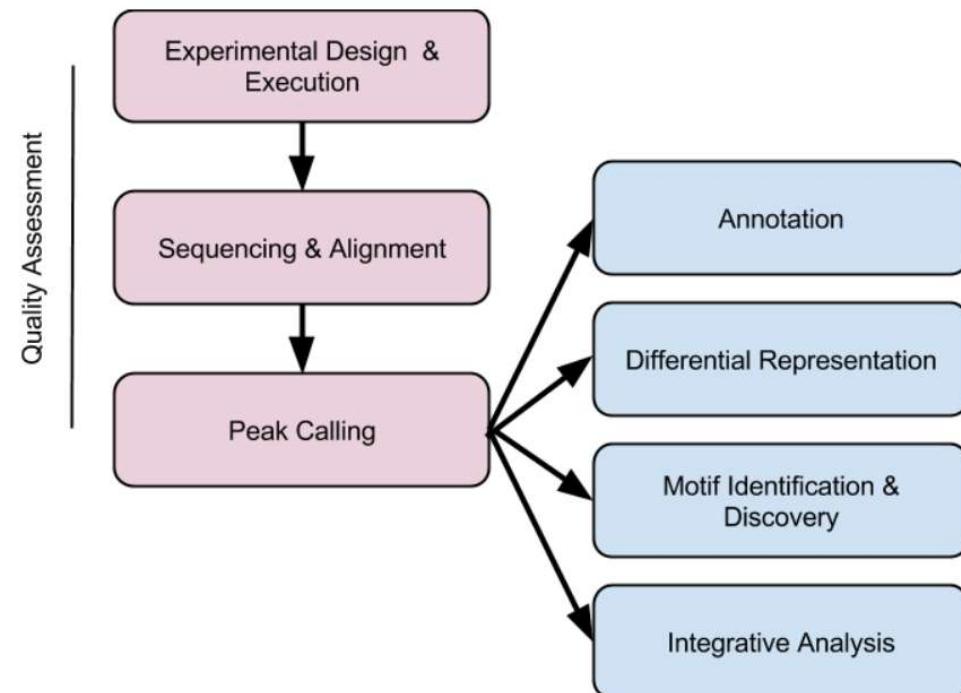
R Class	Description
OrgDb	Gene-based information for Homo sapiens; useful for mapping between gene IDs, Names, Symbols, GO and KEGG identifiers, etc.
TxDb	Transcriptome ranges for the known gene track of Homo sapiens, e.g., introns, exons, UTR regions.
OrganismDb	Collection of multiple annotations for 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/bioconductor-ensembl-reference/>

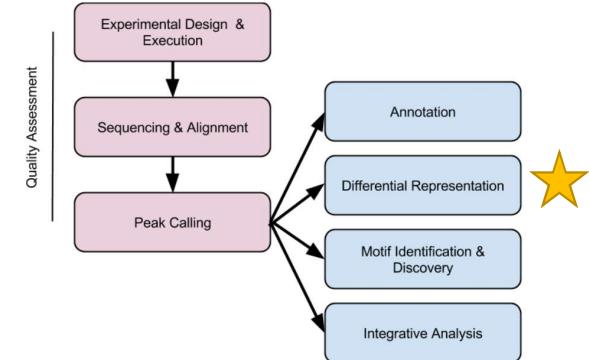
# ChIP-Seq Overview From Last Time

- 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

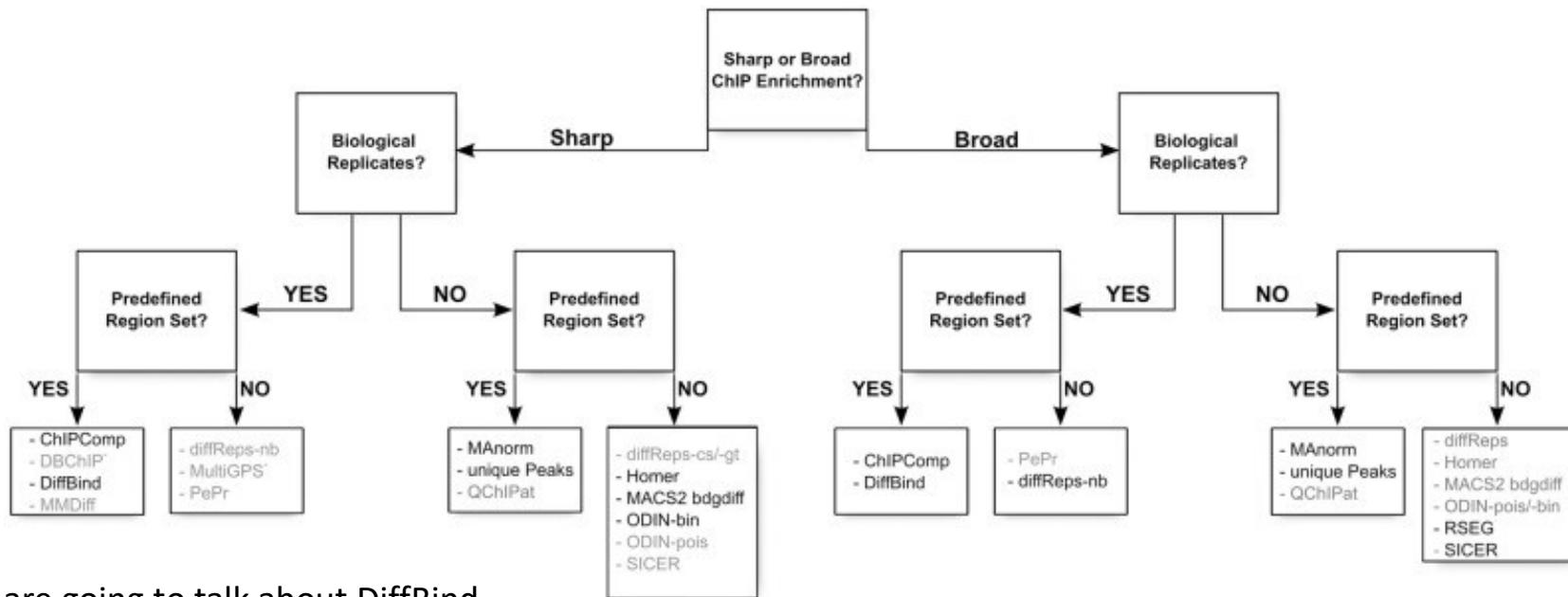
- Other Methods

- 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/vanderl1/Documents/R/win-library/3.5/DiffBind/extr/tamoxifen.csv")
```

```
tamoxifen
```

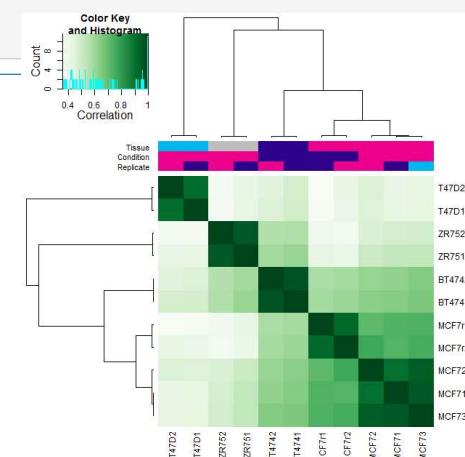
Total # Peaks

```
## 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 least 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

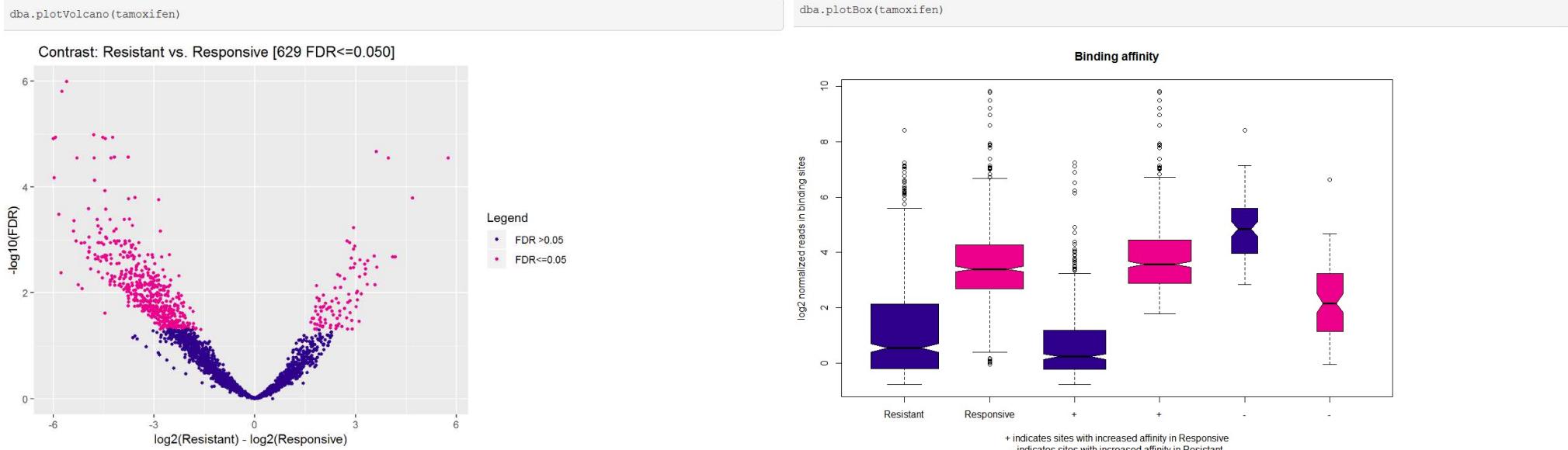
## GRanges object with 629 ranges and 6 metadata columns:
##   seqnames      ranges strand |  Conc_Conc_Responsive
##   <Rle>      <IRanges> <Rle> | <numeric>    <numeric>
## 2452 chr18 64490686-64491186 * | 6.36 1.39
## 1291 chr18 34597713-34598213 * | 5.33 0.22
## 976  chr18 26860997-26861497 * | 7.3  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_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
```

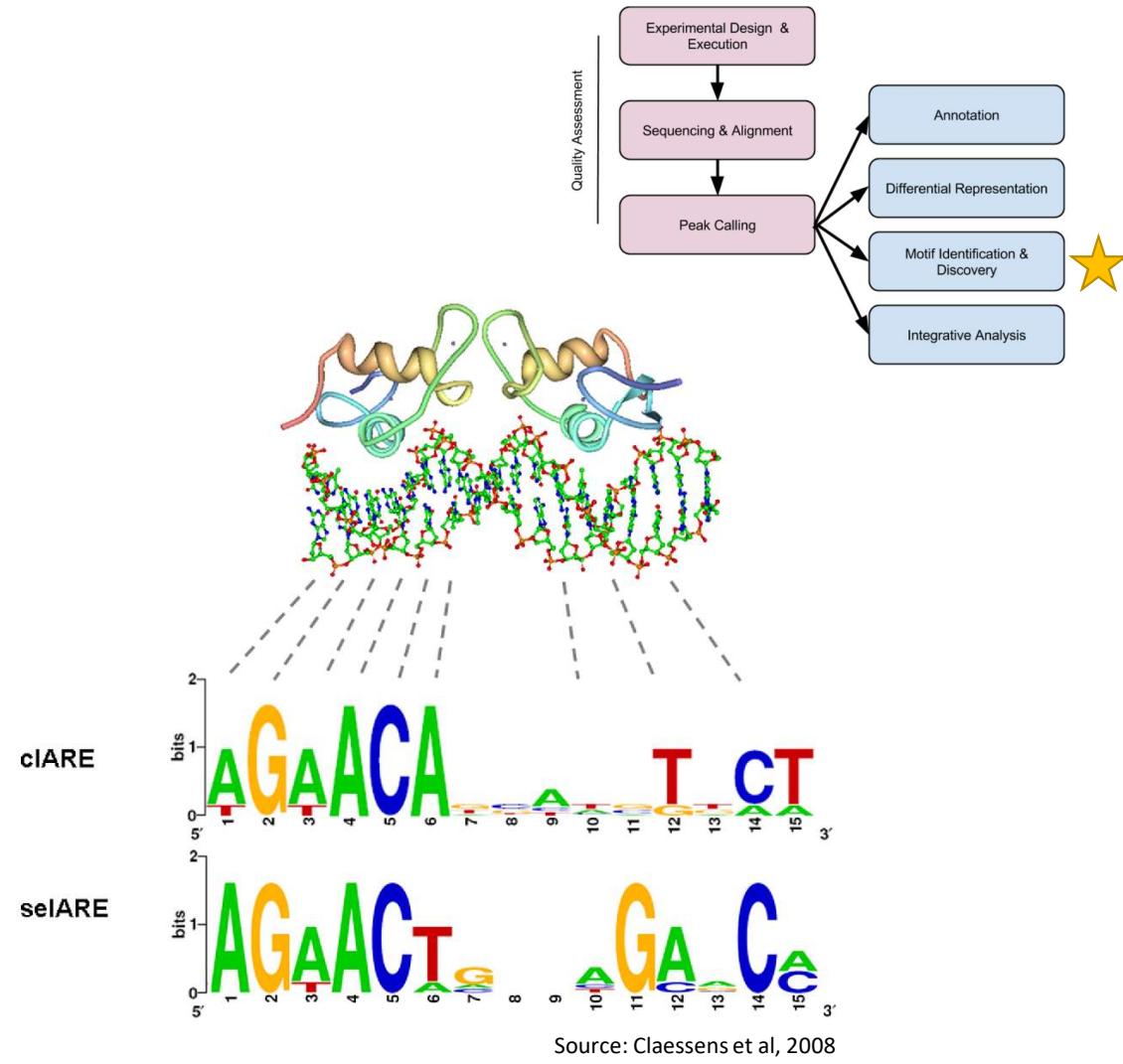
# R/BindDiff

## 6. Visualize Results



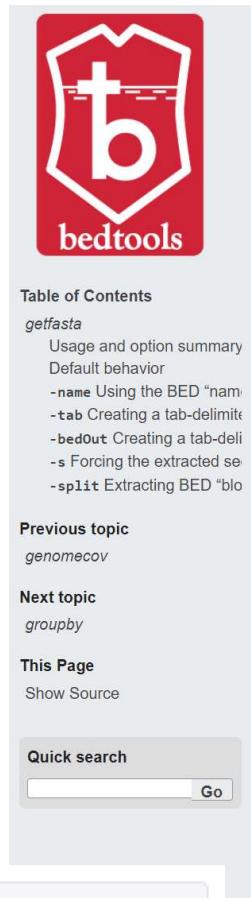
# Motif Discovery

- 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 AGAACCA and the selARE 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 deviate from a homogeneous distribution



# 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



## getfasta

**FASTA** ACAGACTGGTATGAAGGTGCCACAATTCAAGAAAGAAAAAGAGC  
**BED** The diagram shows a horizontal sequence of DNA bases (ACAGACTGGTATGAAGGTGCCACAATTCAAGAAAGAAAAAGAGC) with three vertical yellow boxes indicating extraction regions. The first region covers 'GACT', the second 'TGAAAGGT', and the third 'AAAAAAG'.

**getfasta** GACT TGAAAGGT

AAAAAAG

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>
```

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

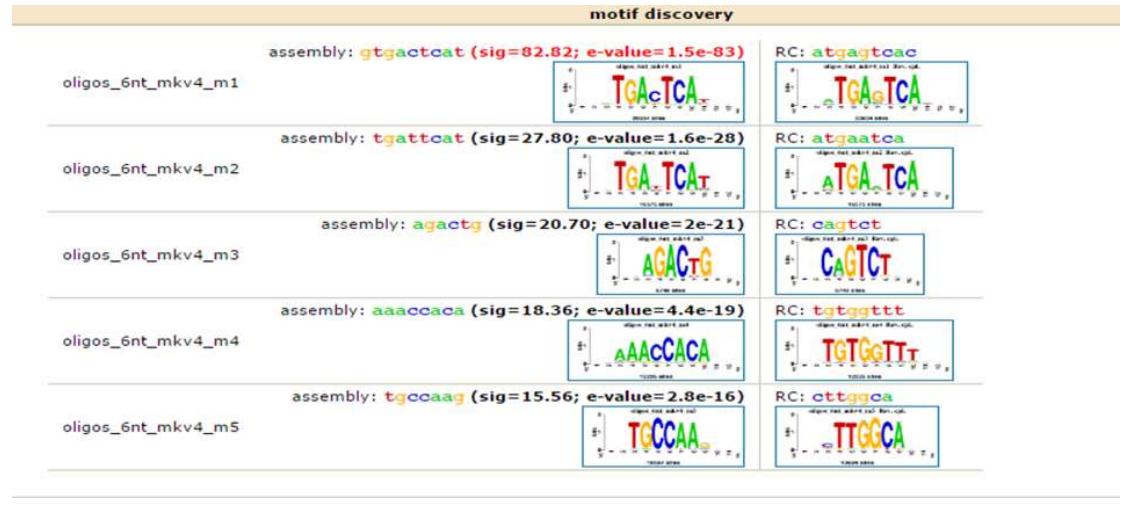
# RSAT – Regulatory Sequence Analysis Tools

[http://rsat.sb-roscoff.fr/peak-motifs\\_form.cgi](http://rsat.sb-roscoff.fr/peak-motifs_form.cgi)

The screenshot shows the RSAT peak-motifs pipeline interface. The left sidebar contains a navigation menu with sections like Genomes and genes, Sequence tools, Matrix tools, Build control sets, Motif discovery, Pattern matching, Comparative genomics, NGS - ChIP-seq (selected), peak-motifs, fetch-sequences from UCSC, sequences from bed/gff/vcf, random genome fragments, Genetic variations, Conversion/Utilities, Drawing, SOAP Web services, Help & Contact (with links to the RSAT team, Training material, Tutorials, Publications, Credits, Download, Motif databases, and Data). The main content area is titled "RSAT - peak-motifs" and describes a "Pipeline for discovering motifs in massive ChIP-seq peak sequences." It features two main input fields: "Peak Sequences" (mandatory) and "Control sequences". Each field has a "Title" input, a "Paste your sequence (fasta format)" text area, and a "Choose File" button. Below each field is a note about ".gz compressed files supported" and a "URL of a sequence file available on a Web server (e.g. Galaxy)." input field. There are also "Mask" dropdown menus. A note at the bottom left says "(I only have coordinates in a BED file, how to get sequences ?)". Below the input fields are several expandable sections: "Reduce peak sequences", "Motif discovery parameters", "Compare discovered motifs with databases (e.g. against JASPAR) or custom reference motifs", "Locate motifs and export predicted sites as custom UCSC tracks", and "Reporting options". Under "Reporting options", there is an "Output" section with radio buttons for "display" and "email", and a text input field. A note states "Note: email output is preferred for very large datasets or many comparisons with motifs collections". At the bottom are buttons for "GO", "Reset", "DEMO single", "DEMO test vs ctrl", "[MANUAL]", and "[TUTORIAL]".

# 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

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)

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

The screenshot shows the MEME Suite 5.0.5 interface with the following sections:

- Motif Discovery:** Options include MEME, DREME, MEME-ChIP, GLAM2, and MoMo. **Motif Discovery** is currently selected.
- Data Submission Form:** A search icon is present. Text instructions: "Perform motif discovery on DNA, RNA, protein or custom alphabet datasets." Radio buttons for "Select the motif discovery mode": Classic mode (selected), Discriminative mode, and Differential Enrichment mode.
- Select the sequence alphabet:** Options: Use sequences with a standard alphabet or specify a custom alphabet. Radio buttons: DNA, RNA or Protein (selected), Custom. Buttons: Choose File, No file chosen.
- Input the primary sequences:** Input field: Enter sequences in which you want to find motifs. Buttons: Upload sequences, Choose File, No file chosen.
- Select the site distribution:** Question: How do you expect motif sites to be distributed in sequences? Radio buttons: Zero or One Occurrence Per Sequence (zoops) (selected).
- Select the number of motifs:** Question: How many motifs should MEME find? Input field: 3.
- Input job details:** (Optional) Enter your email address. (Optional) Enter a job description.
- Advanced options:** Note: if the combined form inputs exceed 80MB the job will be rejected. Buttons: Start Search, Clear Input.

# 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 one 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](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 pre-existing larger USF1 data set containing the top 5211 regions:

```
> data(BCRANKout)
```

## 2.3 BCRANK output

An object of type `BCRANKresult` is returned:

```
> BCRANKout
```

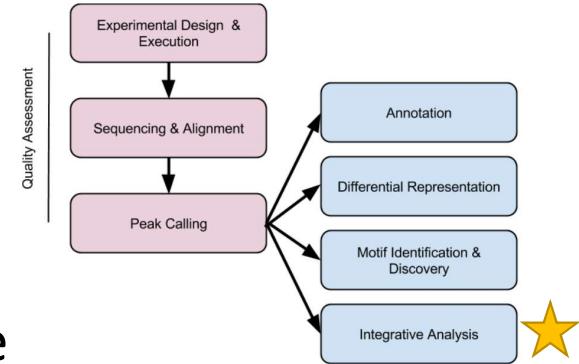
An object of class "BCRANKresult"

Top 25 DNA motifs predicted by BCRANK:

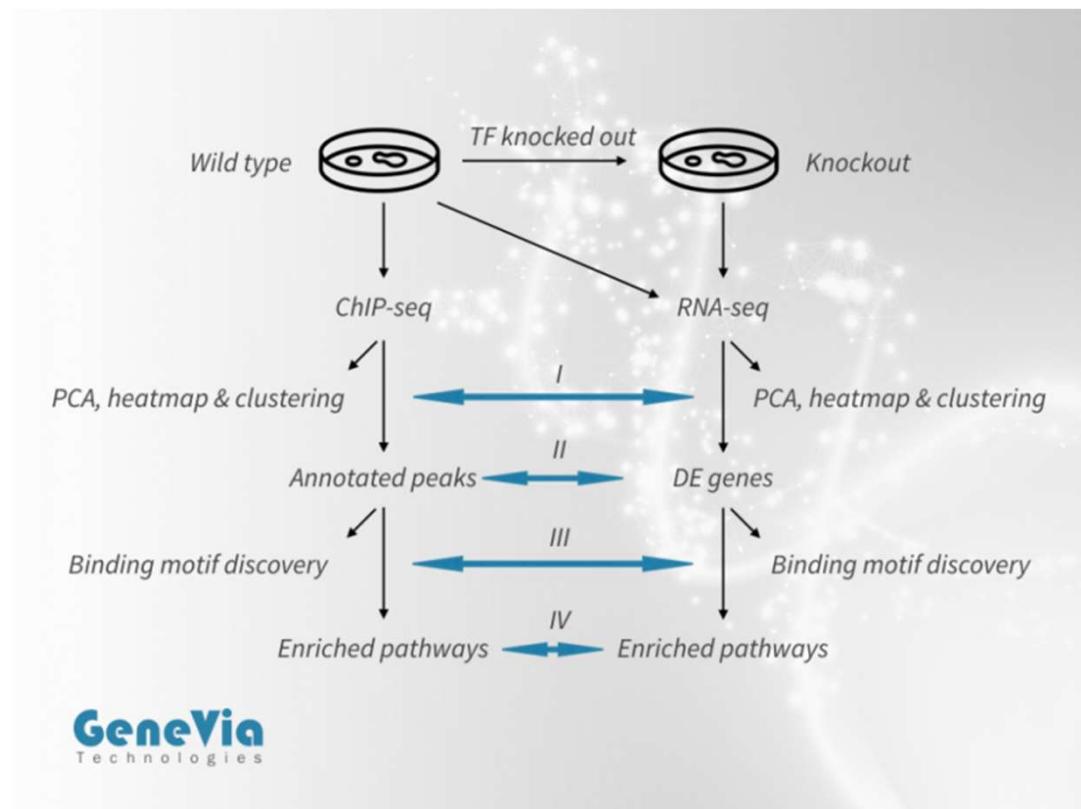
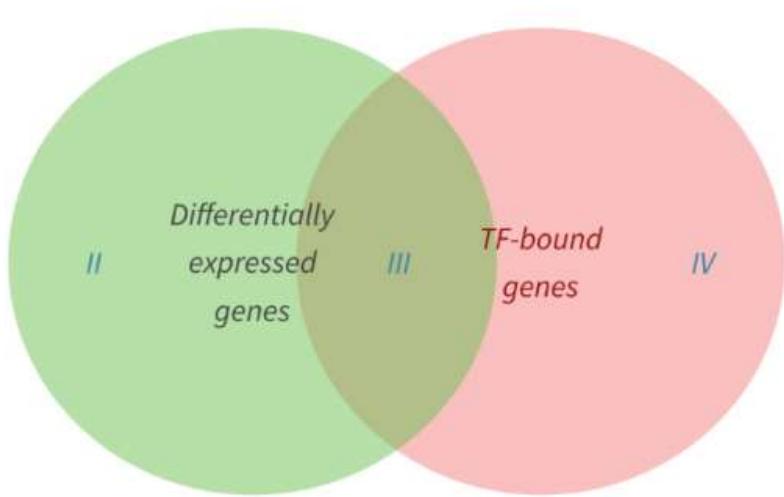
	Consensus	Score
1	GTCACGTG	316.34270
2	CACGTGAC	304.59499
3	CGCGGA	147.04100
4	GCGAST	135.22207
5	AHATAATAA	128.92440
6	GCGGNGCG	121.87198
7	TNCDGGGCG	119.76454
8	GCAGGGVNG	119.65945
9	CCCGCNTBY	118.77481
10	GDGCGGHGH	115.19679
11	TNCGCCNDG	112.95972
12	CGGGNGMGC	111.93052
...		

# 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?

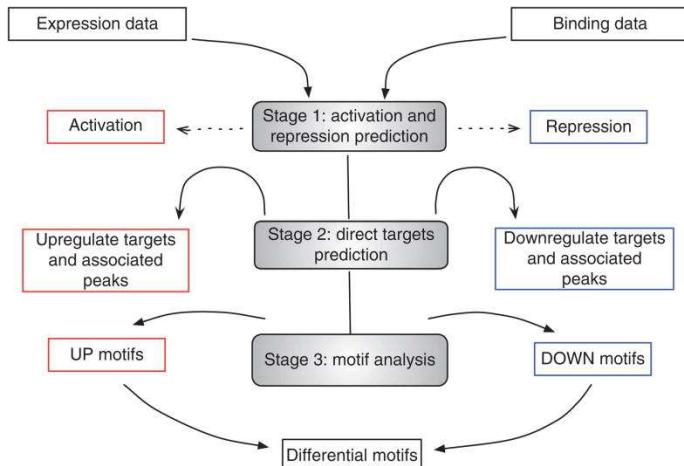


# ChIP-Seq RNA-Seq integration



**GeneVia**  
Technologies

# BETA



Source: Wang et al, 2013

# BETA

Binding and Expression Target Analysis

Introduction | Citation | Run on Webserver | Download | Installation | Tutorial | Contact

## Summary

Binding and Expression Target Analysis (BETA) is a software package that integrates ChIP-seq of transcription factors or chromatin regulators with differential gene expression data to infer direct target genes. BETA has three functions: (1) to predict whether the factor has activating or repressive function; (2) to infer the factor's target genes; and (3) to identify the motif of the factor and its collaborators which might modulate the factor's activating or repressive function. Here we describe the implementation and features of BETA to demonstrate its application to several datasets. BETA requires ~2GB RAM and 1h for the whole procedure.

## Introduction

BETA is a free software to do Transcription Factor and Chromatin Regulator target analysis. Three subcommands of BETA make it user friendly.

BETA basic: TF activating and repressive function prediction and direct targets detecting.  
[Read More](#)

BETA plus: TF activating and repressive function prediction, direct targets detecting and motif analysis on target regions.  
[Read More](#)

BETA minus: TF target genes prediction based on regulatory potential score with only binding data.  
[Read More](#)

# 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
- $\text{peak} \sim \text{GO} + s(\log_{10}\text{length})$ 
  - 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
  - $s(\log_{10}\text{length})$  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.