Bioinformatic approaches to regulatory genomics and epigenomics

376-1347-00L | week 06

Pierre-Luc Germain

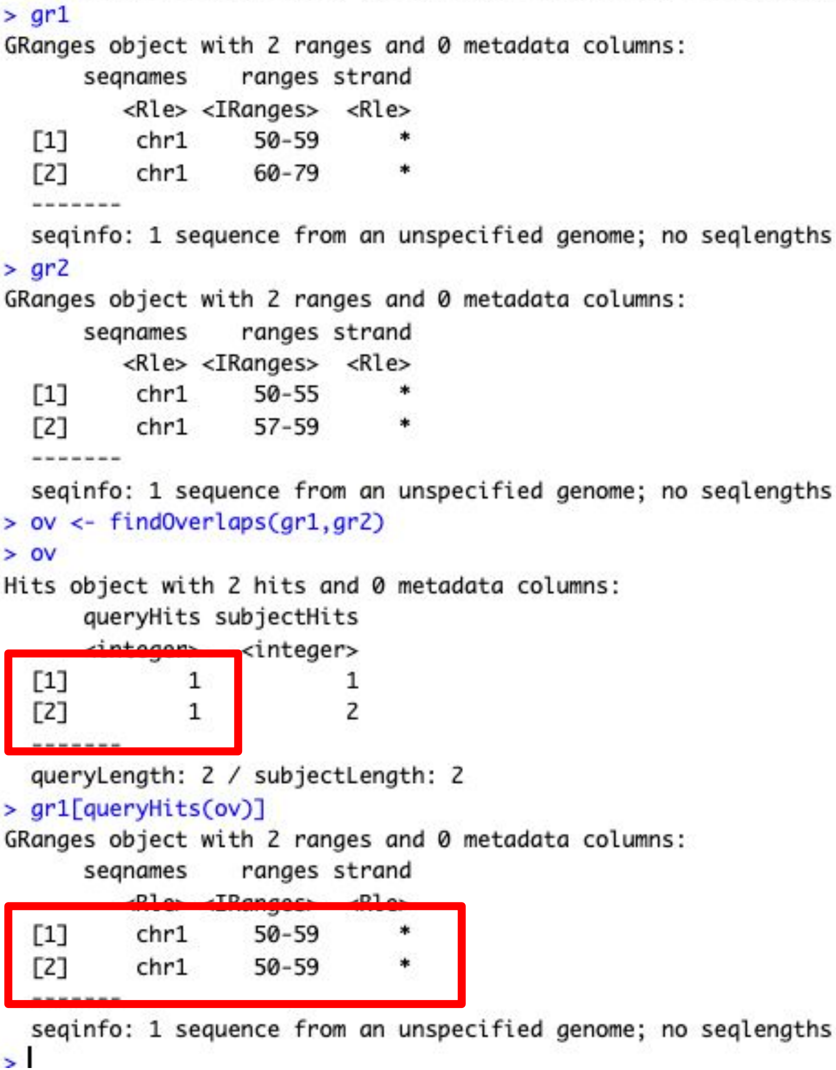
ETH Zürich

Plan

- New packages to install (see slack)
- Debriefing on last week’s assignment
- Overview of transcription factors and their binding specificity
- DNA motifs and related analysis

Recap

`findOverlaps():`

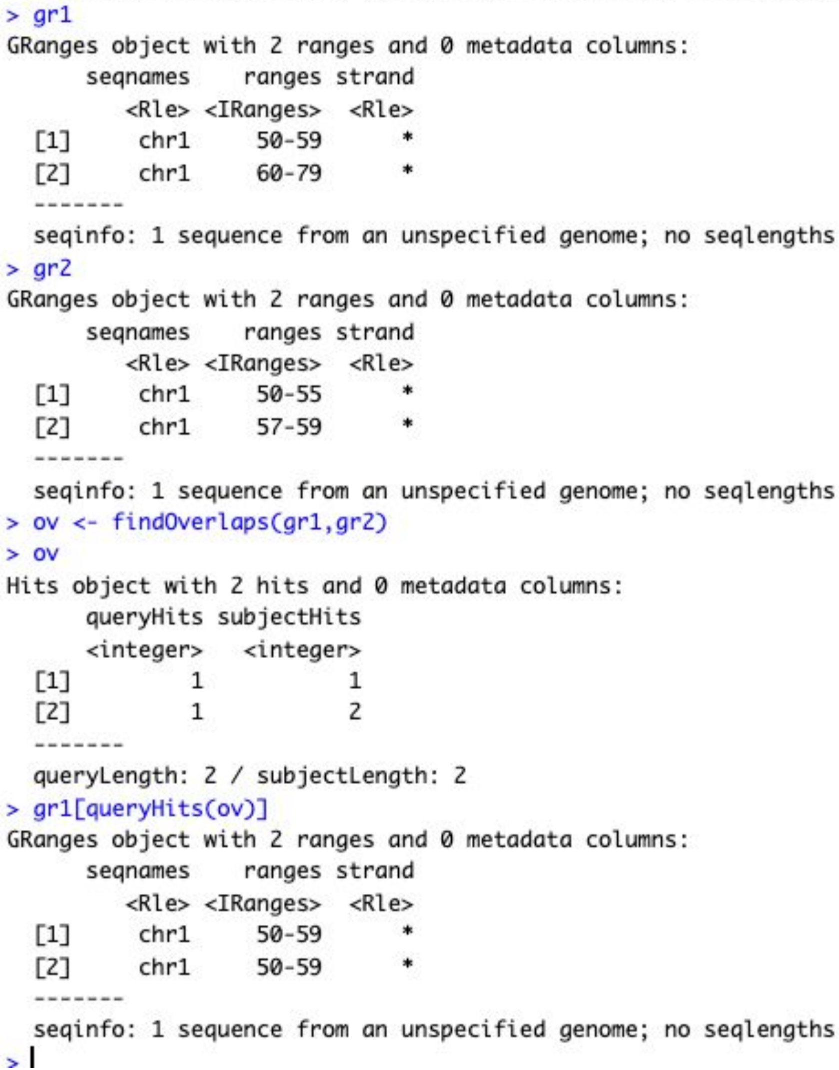


```
> gr1
GRanges object with 2 ranges and 0 metadata columns:
      seqnames    ranges strand
         <Rle> <IRanges>  <Rle>
  [1]     chr1     50-59      *
  [2]     chr1     60-79      *
  -------
  seqinfo: 1 sequence from an unspecified genome; no seqlengths
> gr2
GRanges object with 2 ranges and 0 metadata columns:
      seqnames    ranges strand
         <Rle> <IRanges>  <Rle>
  [1]     chr1     50-55      *
  [2]     chr1     57-59      *
  -------
  seqinfo: 1 sequence from an unspecified genome; no seqlengths
> ov <- findOverlaps(gr1,gr2)
> ov
Hits object with 2 hits and 0 metadata columns:
      queryHits subjectHits
      <integer>   <integer>
  [1]         1           1
  [2]         1           2
  -------
  queryLength: 2 / subjectLength: 2
> gr1[queryHits(ov)]
GRanges object with 2 ranges and 0 metadata columns:
      seqnames    ranges strand
         <Rle> <IRanges>  <Rle>
  [1]     chr1     50-59      *
  [2]     chr1     50-59      *
  -------
  seqinfo: 1 sequence from an unspecified genome; no seqlengths
> 
```

Recap

`findOverlaps():`

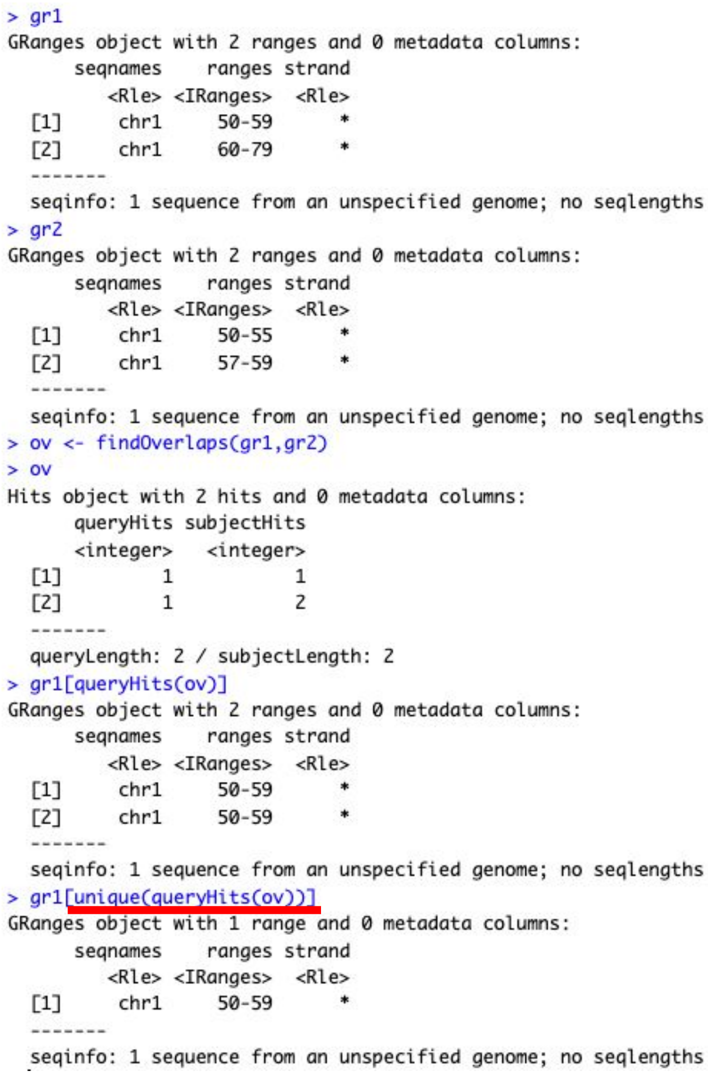
Depending on what you aim to do,
you do not want to have the duplicates.



```
> gr1
GRanges object with 2 ranges and 0 metadata columns:
      seqnames    ranges strand
         <Rle> <IRanges>  <Rle>
  [1]     chr1     50-59      *
  [2]     chr1     60-79      *
  -------
  seqinfo: 1 sequence from an unspecified genome; no seqlengths
> gr2
GRanges object with 2 ranges and 0 metadata columns:
      seqnames    ranges strand
         <Rle> <IRanges>  <Rle>
  [1]     chr1     50-55      *
  [2]     chr1     57-59      *
  -------
  seqinfo: 1 sequence from an unspecified genome; no seqlengths
> ov <- findOverlaps(gr1,gr2)
> ov
Hits object with 2 hits and 0 metadata columns:
      queryHits subjectHits
      <integer>   <integer>
  [1]         1           1
  [2]         1           2
  -------
  queryLength: 2 / subjectLength: 2
> gr1[queryHits(ov)]
GRanges object with 2 ranges and 0 metadata columns:
      seqnames    ranges strand
         <Rle> <IRanges>  <Rle>
  [1]     chr1     50-59      *
  [2]     chr1     50-59      *
  -------
  seqinfo: 1 sequence from an unspecified genome; no seqlengths
> 
```

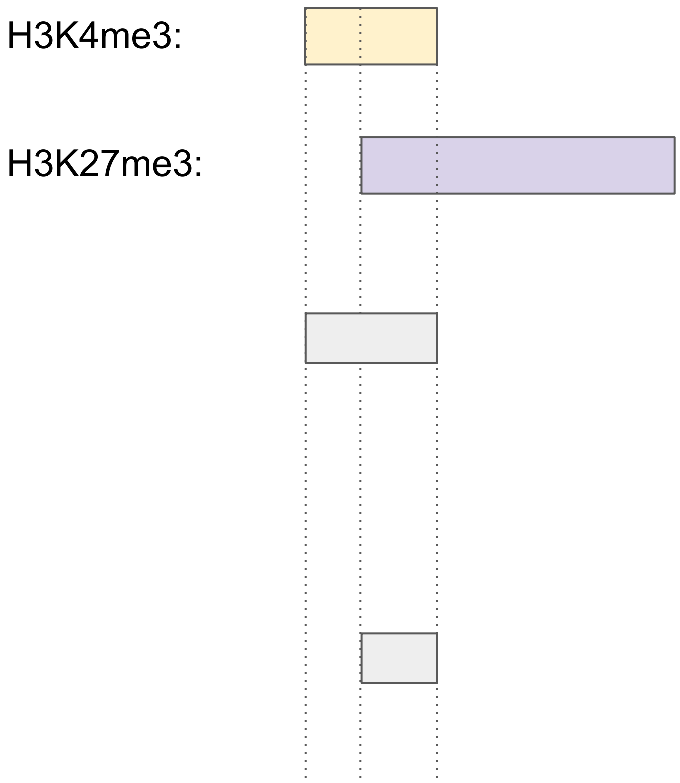
Recap

use either, depending on the aim, `unique()` or `overlapsAny()` or `subsetByOverlaps()`



```
> gr1
GRanges object with 2 ranges and 0 metadata columns:
      seqnames    ranges strand
         <Rle> <IRanges>  <Rle>
  [1]     chr1     50-59      *
  [2]     chr1     60-79      *
  -------
  seqinfo: 1 sequence from an unspecified genome; no seqlengths
> gr2
GRanges object with 2 ranges and 0 metadata columns:
      seqnames    ranges strand
         <Rle> <IRanges>  <Rle>
  [1]     chr1     50-55      *
  [2]     chr1     57-59      *
  -------
  seqinfo: 1 sequence from an unspecified genome; no seqlengths
> ov <- findOverlaps(gr1,gr2)
> ov
Hits object with 2 hits and 0 metadata columns:
      queryHits subjectHits
      <integer>   <integer>
  [1]         1           1
  [2]         1           2
  -------
  queryLength: 2 / subjectLength: 2
> gr1[queryHits(ov)]
GRanges object with 2 ranges and 0 metadata columns:
      seqnames    ranges strand
         <Rle> <IRanges>  <Rle>
  [1]     chr1     50-59      *
  [2]     chr1     50-59      *
  -------
  seqinfo: 1 sequence from an unspecified genome; no seqlengths
> gr1[unique(queryHits(ov))]
GRanges object with 1 range and 0 metadata columns:
      seqnames    ranges strand
         <Rle> <IRanges>  <Rle>
  [1]     chr1     50-59      *
  -------
  seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

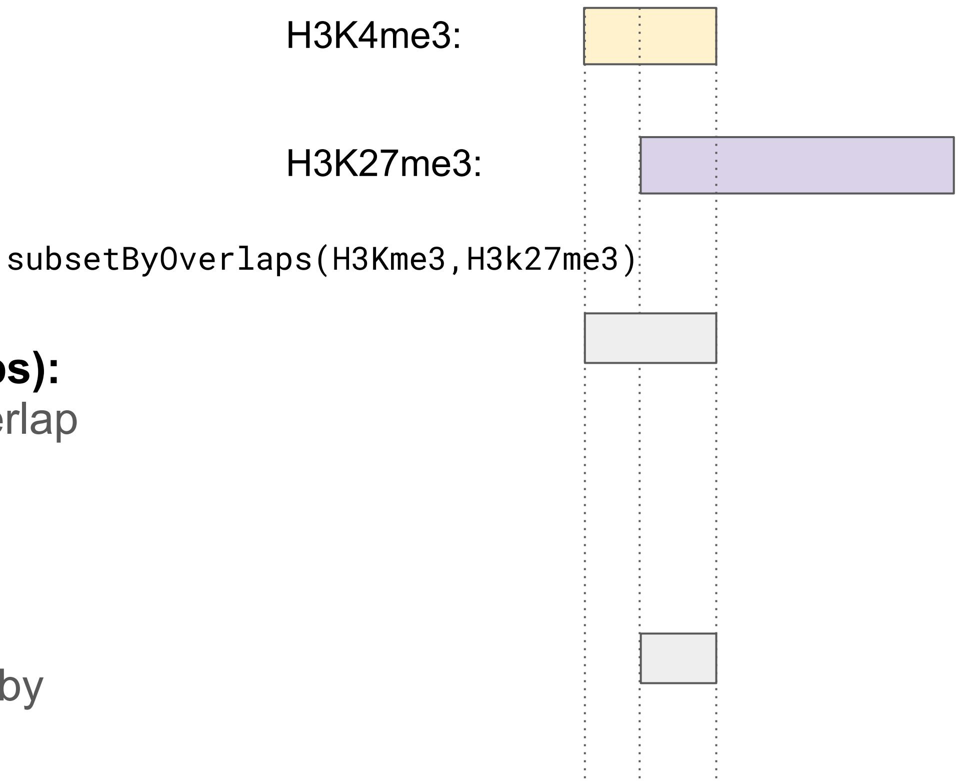
Debriefing: Intersection & overlap
The example of bivalent domains



- **method one (overlapsAny/subsetByOverlaps):** find the H3K4me3 peaks that overlap a H3K27me3 domain

- **method two (intersect):** find the regions that are covered by both H3K4me3 and H3K27me3

Debriefing: Intersection & overlap
The example of bivalent domains

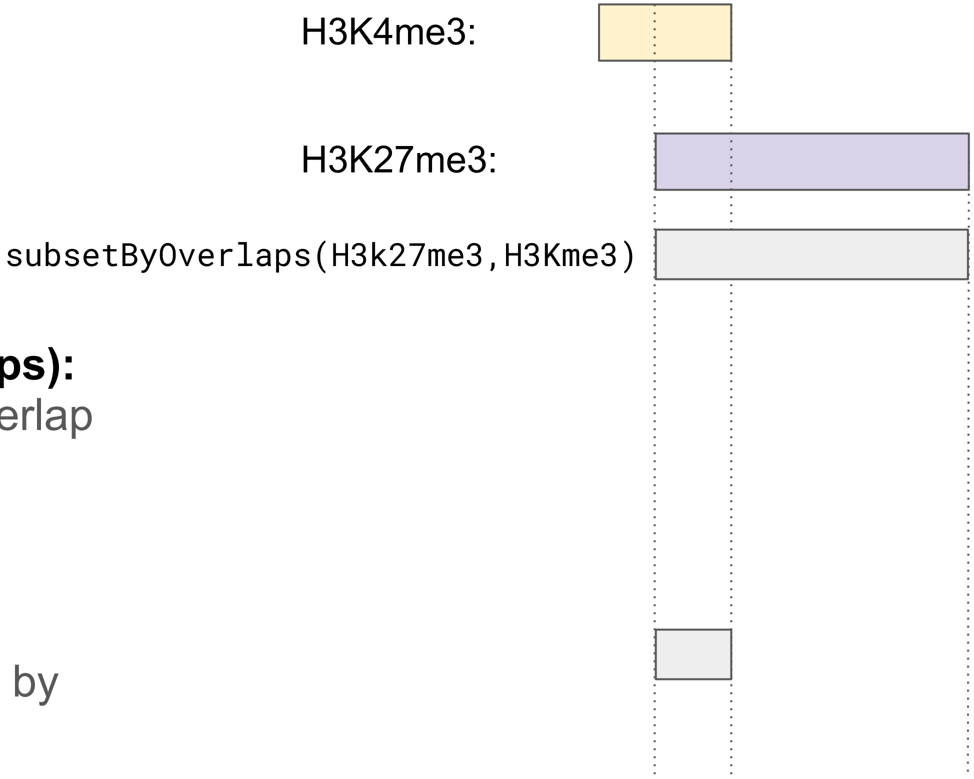


```
subsetByOverlaps(H3Kme3,H3k27me3)
```

- **method one (overlapsAny/subsetByOverlaps):** find the H3K4me3 peaks that overlap a H3K27me3 domain

- **method two (intersect):** find the regions that are covered by both H3K4me3 and H3K27me3

Debriefing: Intersection & overlap
The example of bivalent domains



H3K4me3:

H3K27me3:

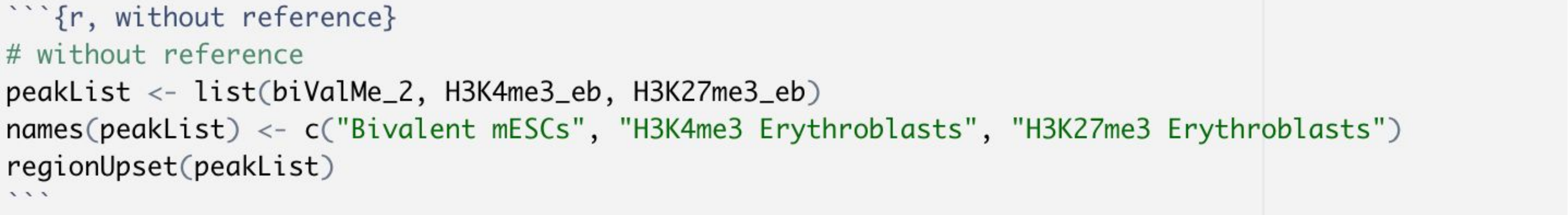
`subsetByOverlaps(H3k27me3,H3Kme3)`

- **method one (overlapsAny/subsetByOverlaps):** find the H3K4me3 peaks that overlap a H3K27me3 domain

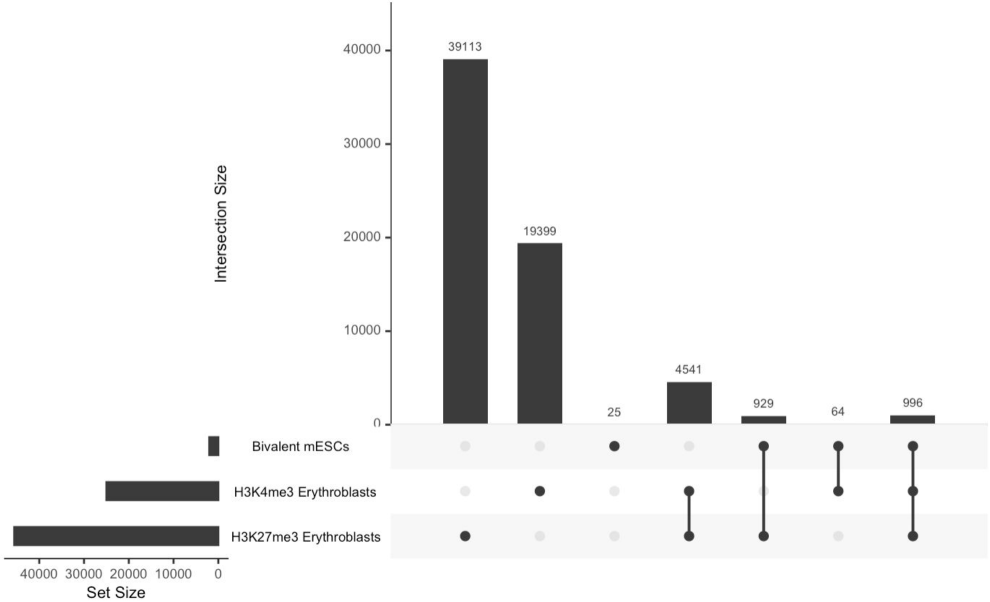
- **method two (intersect):** find the regions that are covered by both H3K4me3 and H3K27me3

Debriefing: upset plots

Using references for upset plot

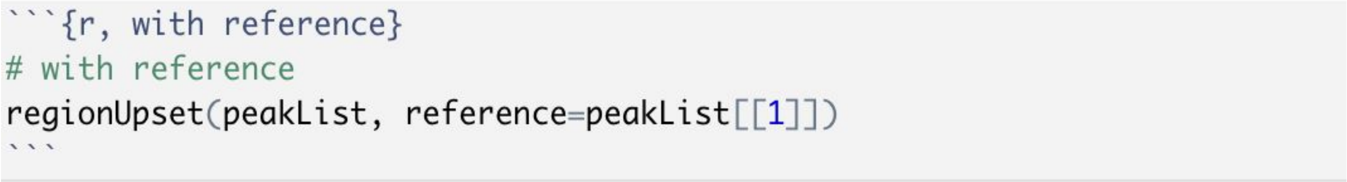


~~~
```{r, without reference}
# without reference
peakList <- list(biValMe_2, H3K4me3_eb, H3K27me3_eb)
names(peakList) <- c("Bivalent mESCs", "H3K4me3 Erythroblasts", "H3K27me3 Erythroblasts")
regionUpset(peakList)
```
~~~

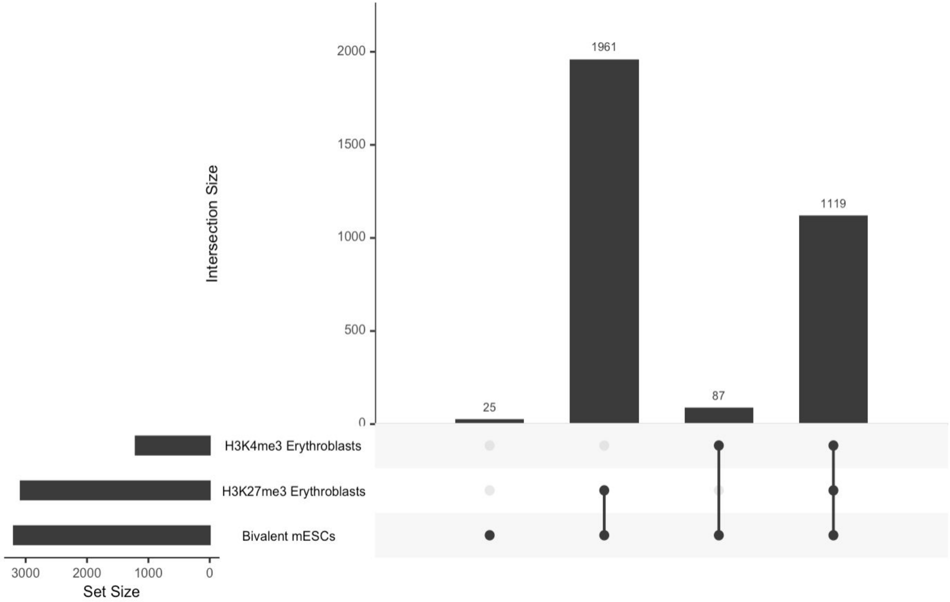


Debriefing: upset plots

Using references for upset plot

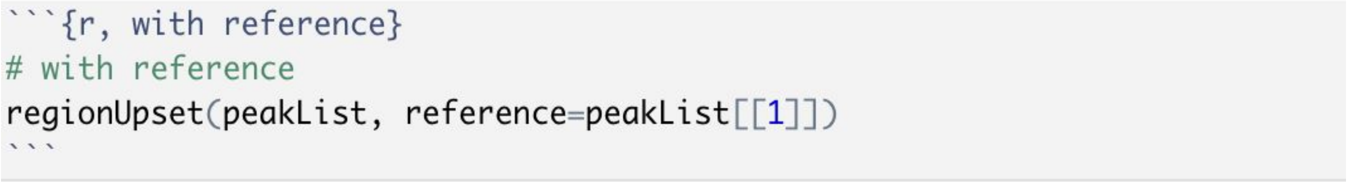


~~~
```{r, with reference}
# with reference
regionUpset(peakList, reference=peakList[[1]])
```
~~~

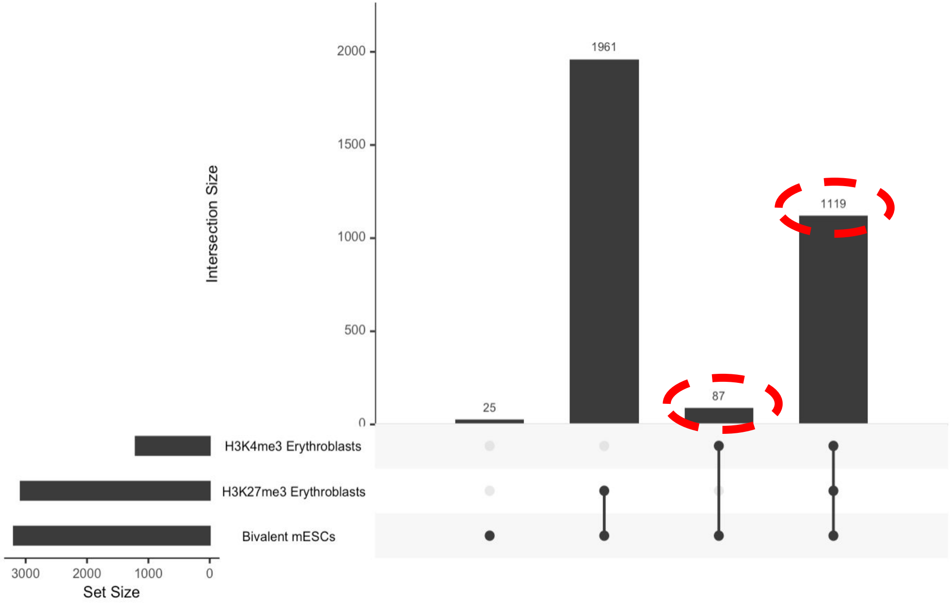


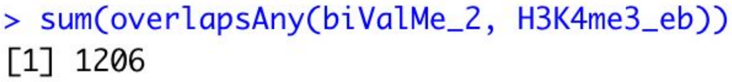
Debriefing on the assignments

Using references for upset plot



~~~
```{r, with reference}
# with reference
regionUpset(peakList, reference=peakList[[1]])
```
~~~



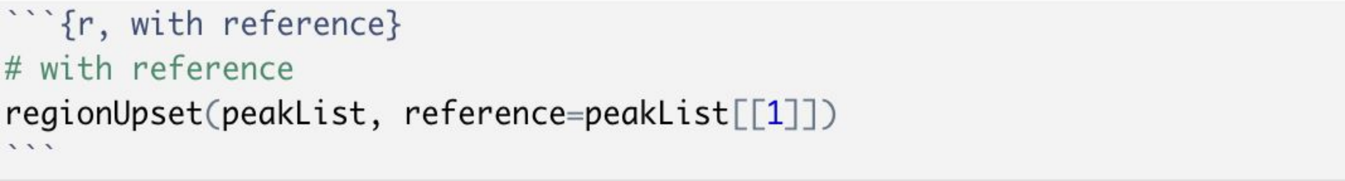


```
> sum(overlapsAny(biValMe_2, H3K4me3_eb))
[1] 1206
```

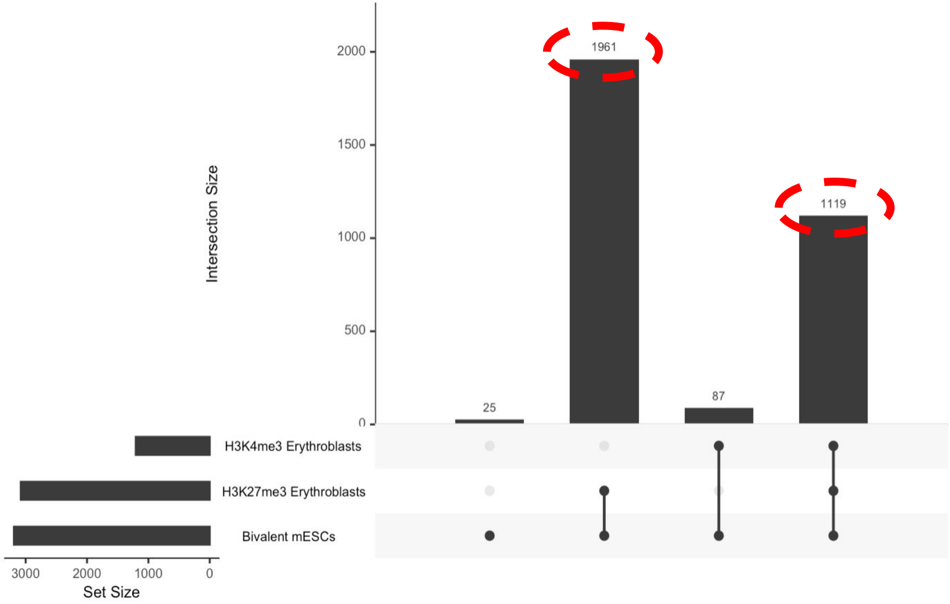
=87+1119

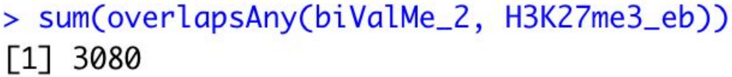
Debriefing on the assignments

Using references for upset plot



~~~
```{r, with reference}
# with reference
regionUpset(peakList, reference=peakList[[1]])
```
~~~



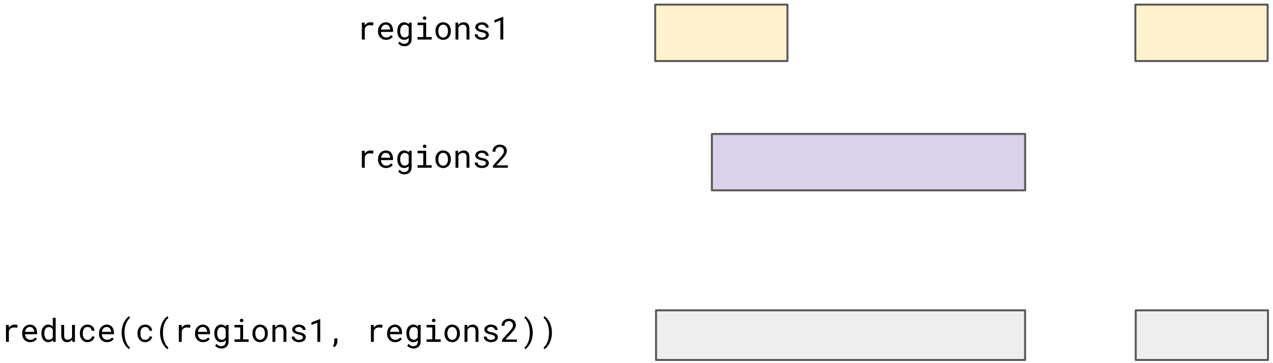


```
> sum(overlapsAny(biValMe_2, H3K27me3_eb))
[1] 3080
```

=1916+1119

Debriefing on the assignments

When no reference is specified, one is created automatically by merging and *reducing* the regions (unless otherwise specified in the arguments):



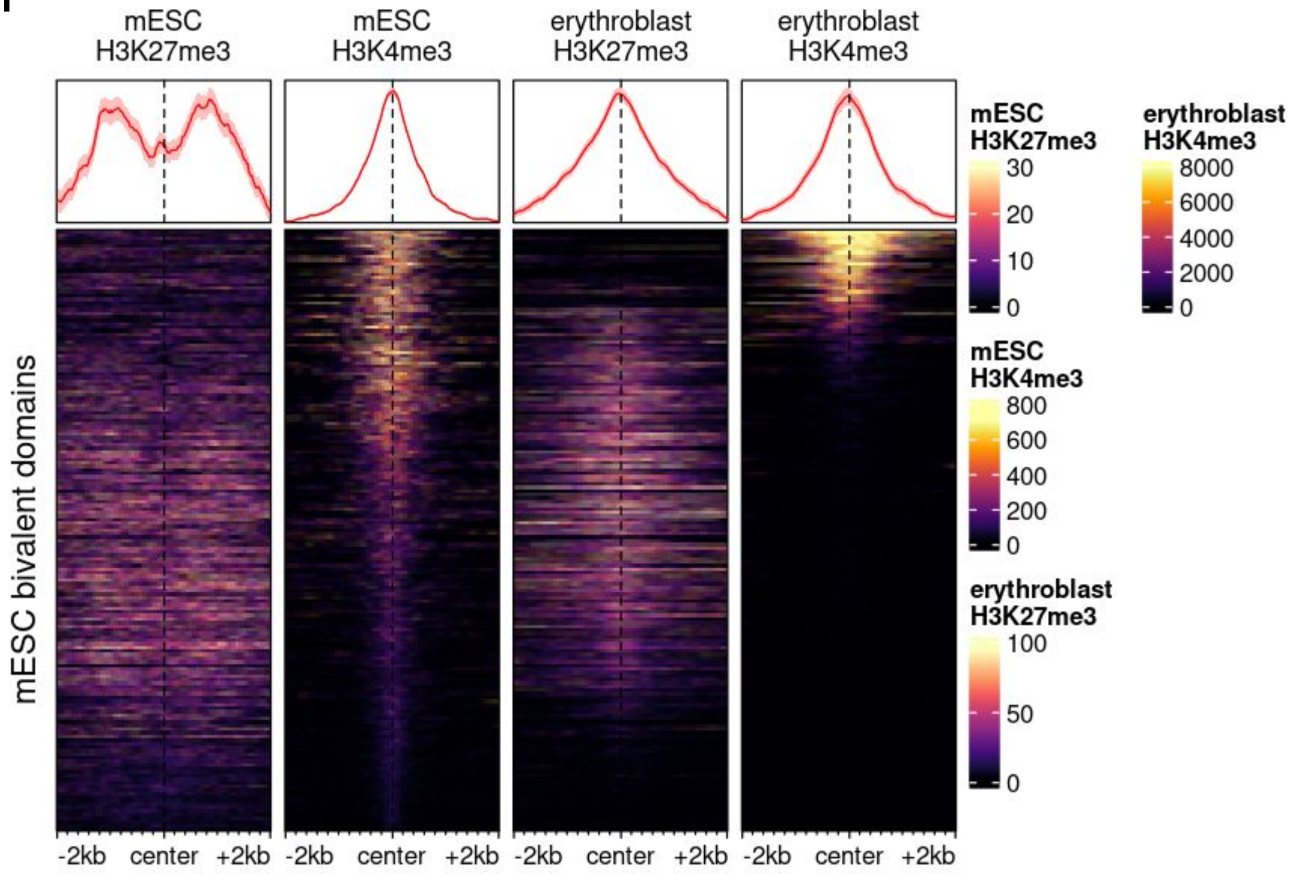
regions1

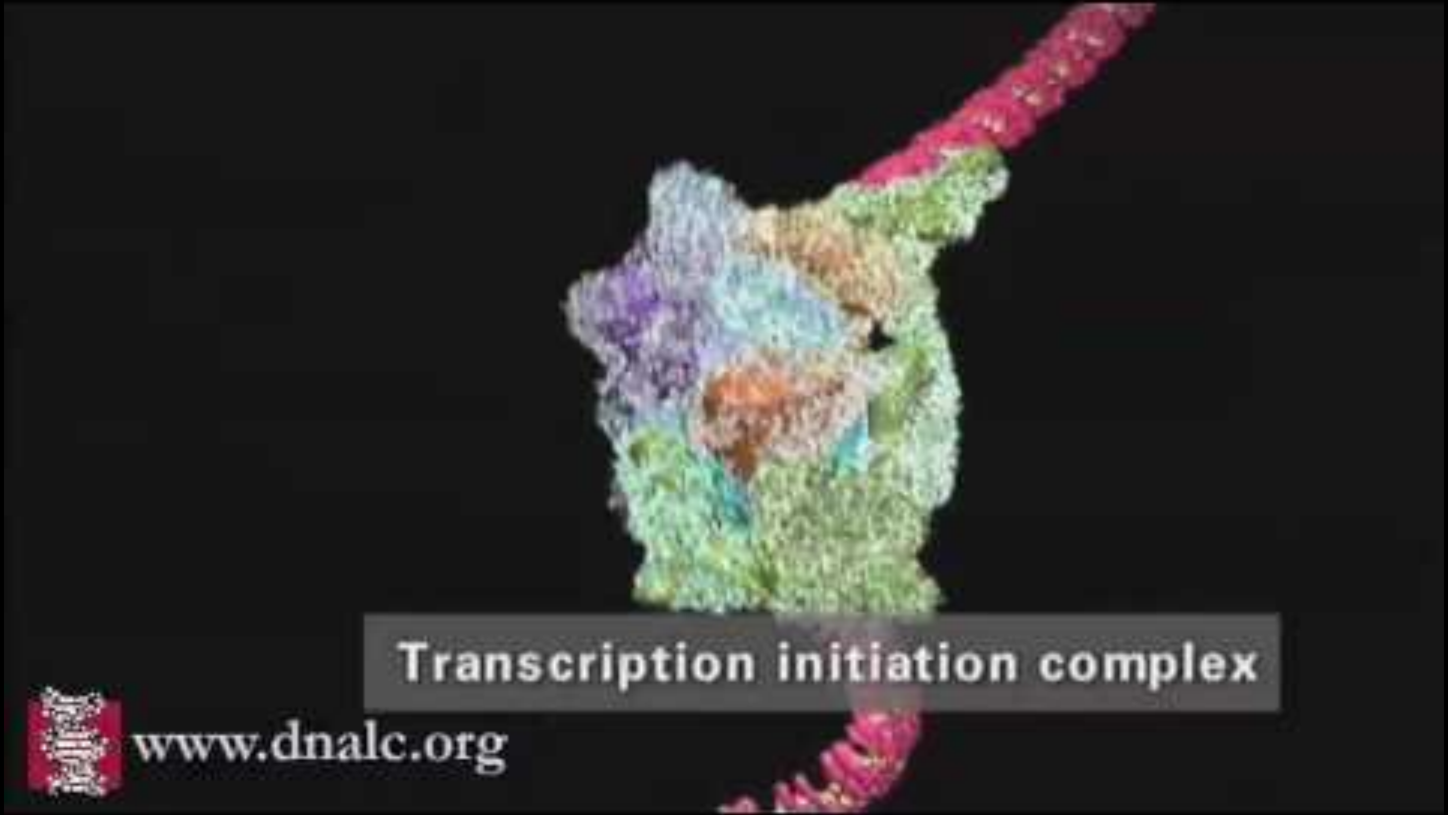
regions2

reduce(c(regions1, regions2))

Embryonic bivalent domains binarize into active and inactive upon differentiation

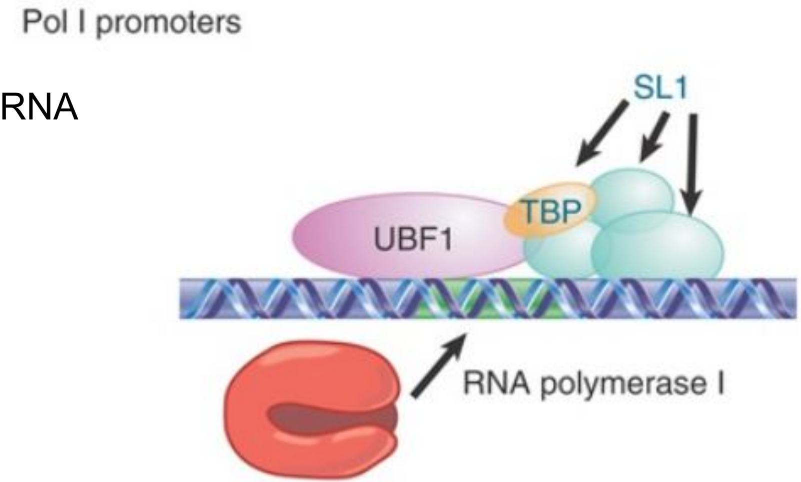
```
Bivalent <-
intersect(mESC_K27me3,
mESC_K4m3)

Bw <- c(4 experiments)
O <- signal2Matrix(Bw,
regions=Bivalent)

plotEnrichedHeatmap(O,
multiScale=TRUE)
```



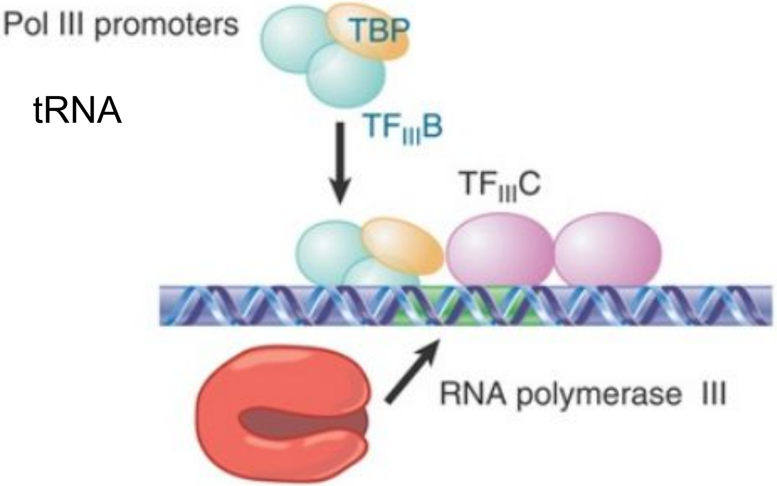


https://youtu.be/SMtWvDbfHLo

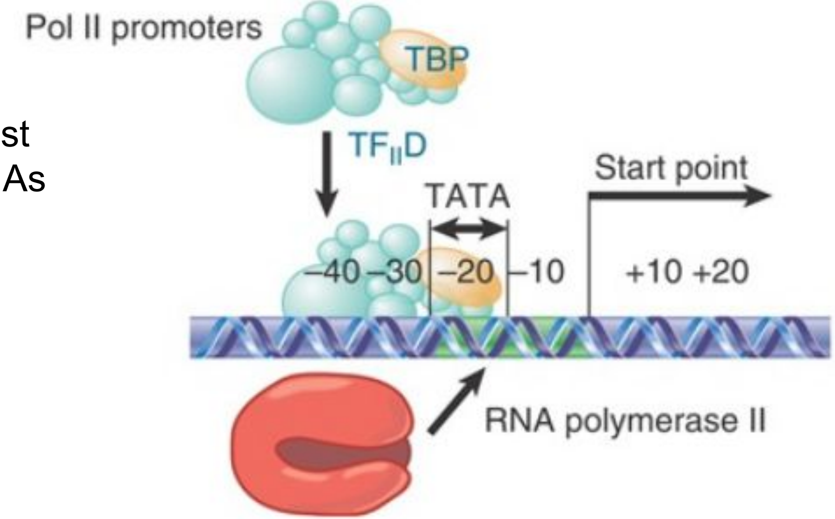
(See also https://youtu.be/WW9lIYM_FC0)



rRNA



tRNA

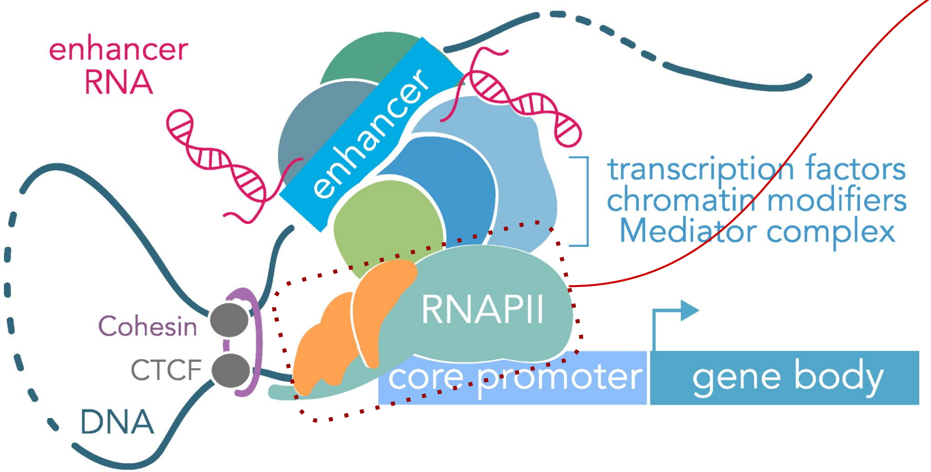


Most RNAs

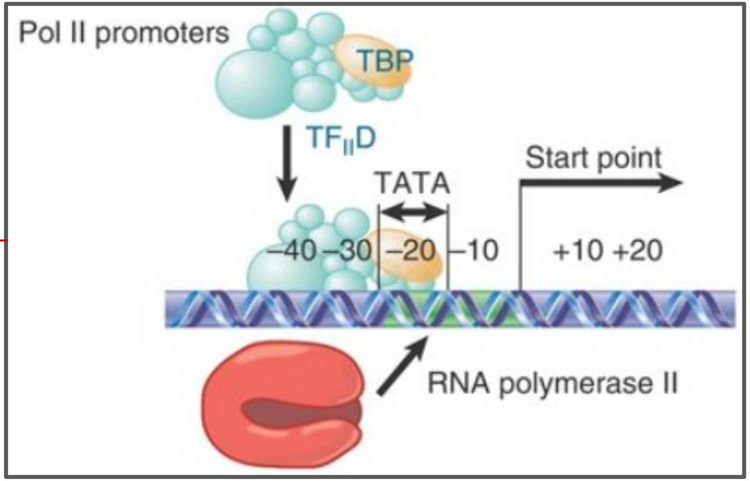
(Adapted from Krebs, Goldstein and Kilpatrick, Genes XII, 2018)

Additional regulatory elements

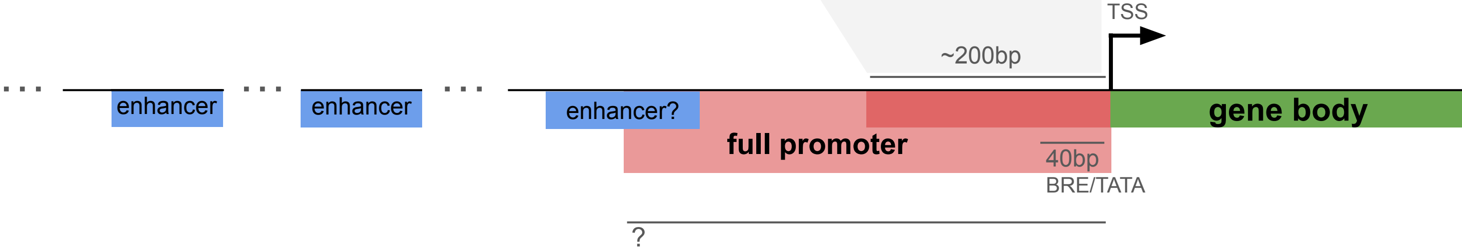
Enhancer-driven gene regulation



(Carullo and Day, Genes 2019)



"function as non-cell-type-specific ‘on switches’ providing similar expression levels to their associated gene" (Agarwal et al., biorxiv 2023)



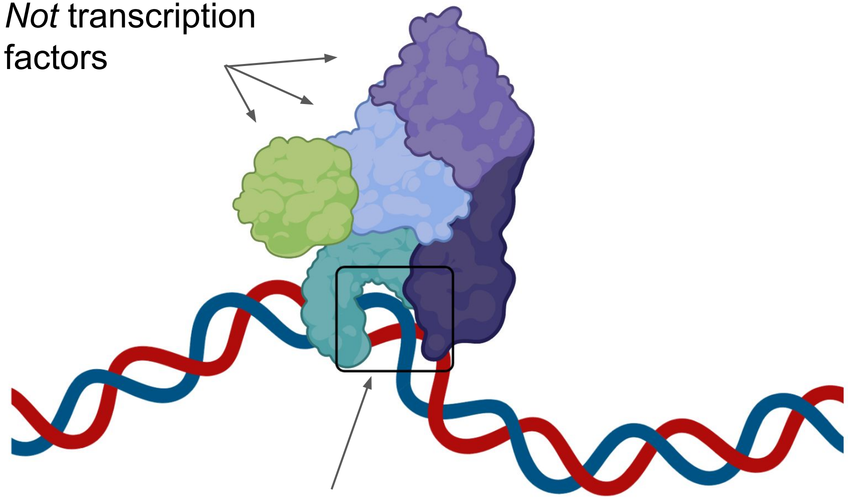
What is a transcription factor?

Proteins capable of both:

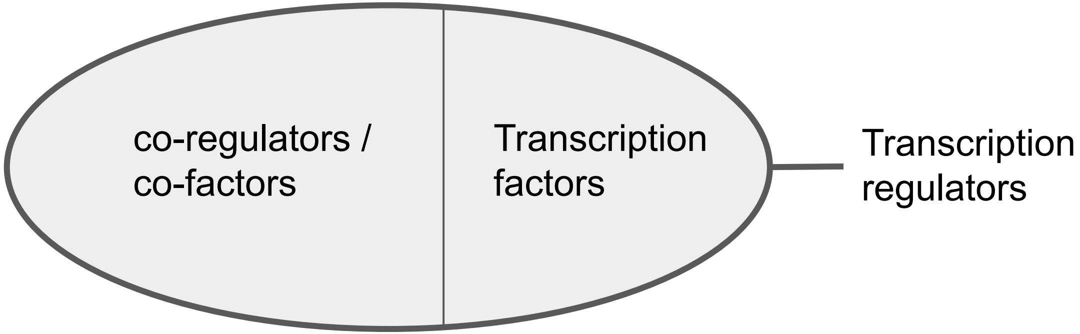
1) Binding DNA in a sequence-specific manner

2) Regulating transcription

(Lambert et al., Cell 2018)



(biorender)

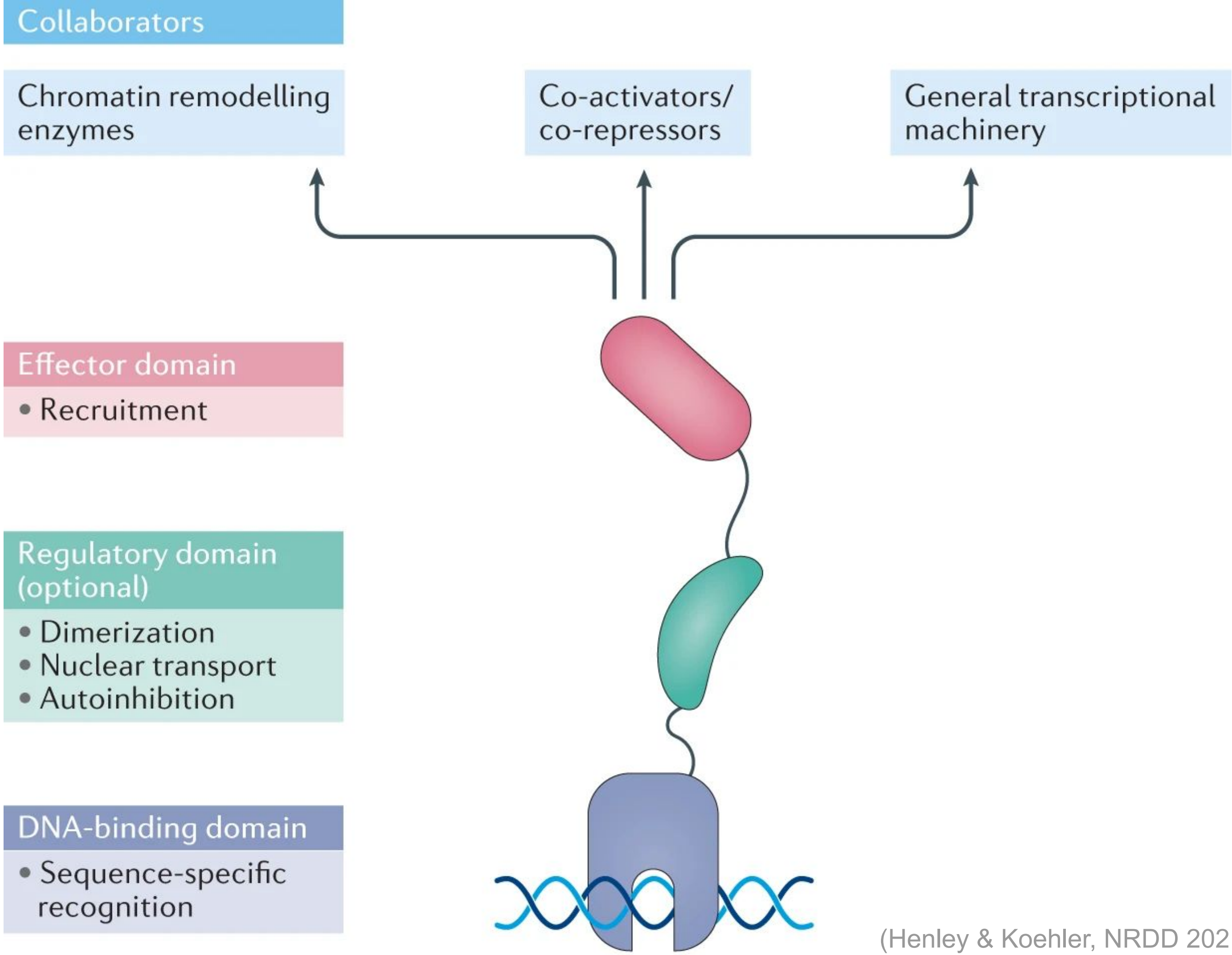


co-regulators / co-factors

Transcription factors

Transcription regulators

Anatomy of a transcription factor (TF)



Collaborators

Chromatin remodelling enzymes

Co-activators/ co-repressors

General transcriptional machinery

Effector domain

- Recruitment

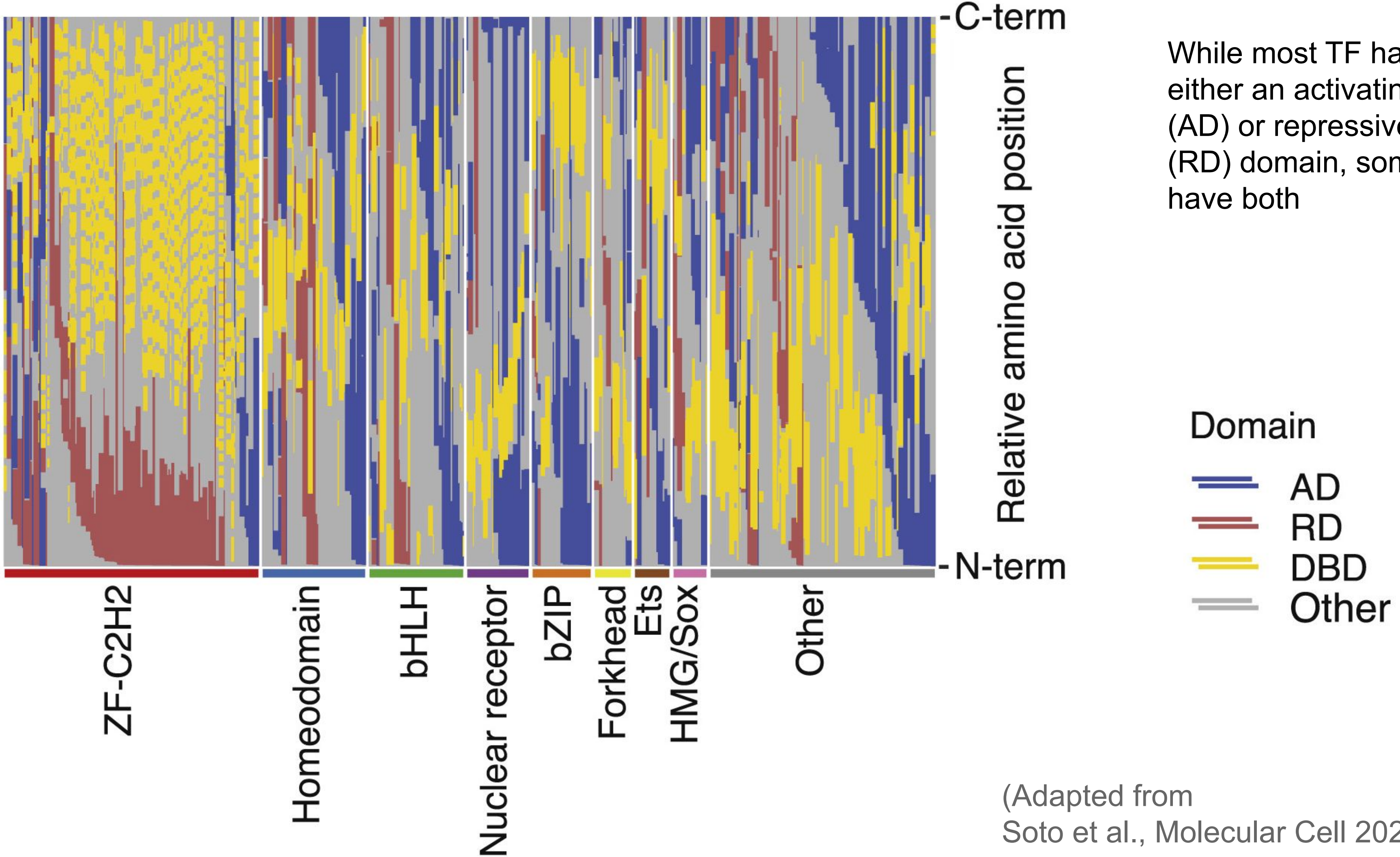
Regulatory domain (optional)

- Dimerization
- Nuclear transport
- Autoinhibition

DNA-binding domain

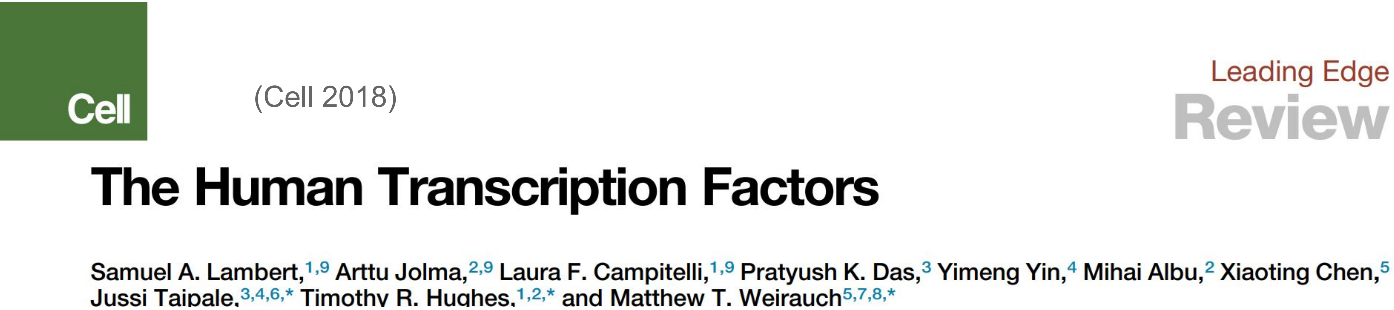
- Sequence-specific recognition

(Henley & Koehler, NRDD 2021)



While most TF have either an activating (AD) or repressive (RD) domain, some have both

(Adapted from Soto et al., Molecular Cell 2021)



Cell

(Cell 2018)

Leading Edge
Review

The Human Transcription Factors

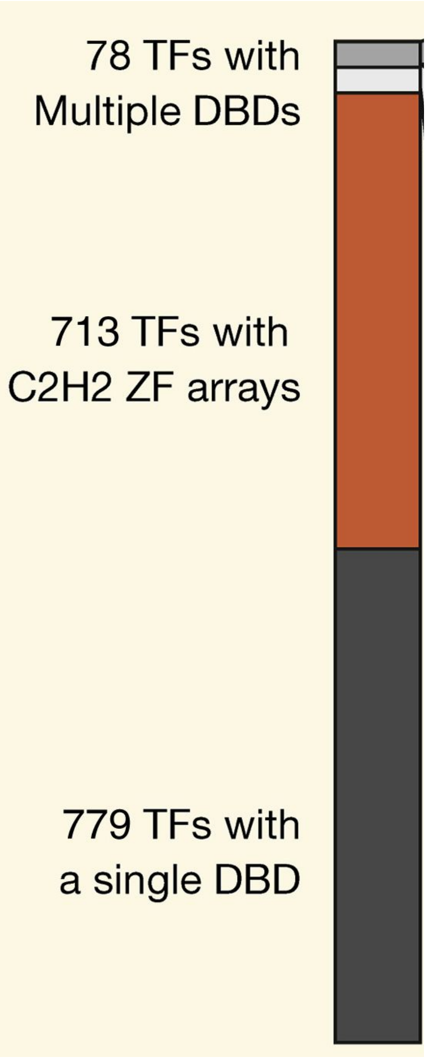
Samuel A. Lambert,[1,9] Arttu Jolma,[2,9] Laura F. Campitelli,[1,9] Pratyush K. Das,[3] Yimeng Yin,[4] Mihai Albu,[2] Xiaoting Chen,[5] Jussi Taipale,[3,4,6,*] Timothy R. Hughes,[1,2,*] and Matthew T. Weirauch[5,7,8,*]

Proteins capable of both:

1) Binding DNA in a sequence-specific manner

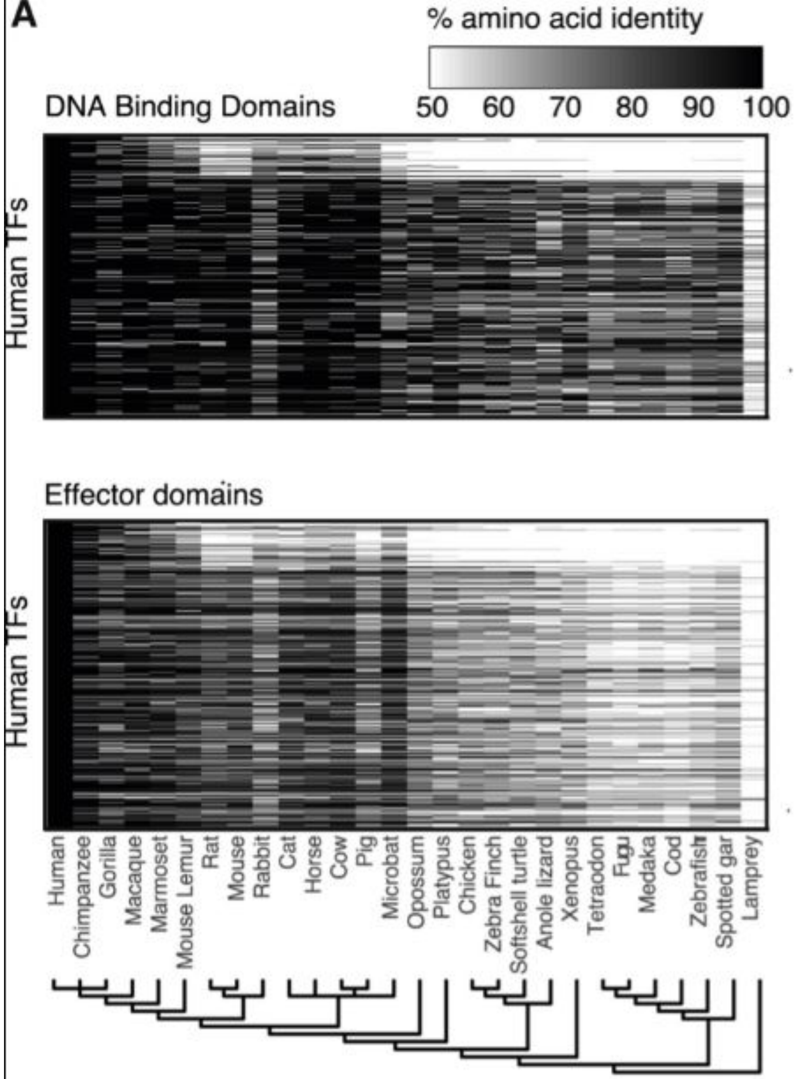
2) Regulating transcription

According to their census, humans have 1570 transcription factors

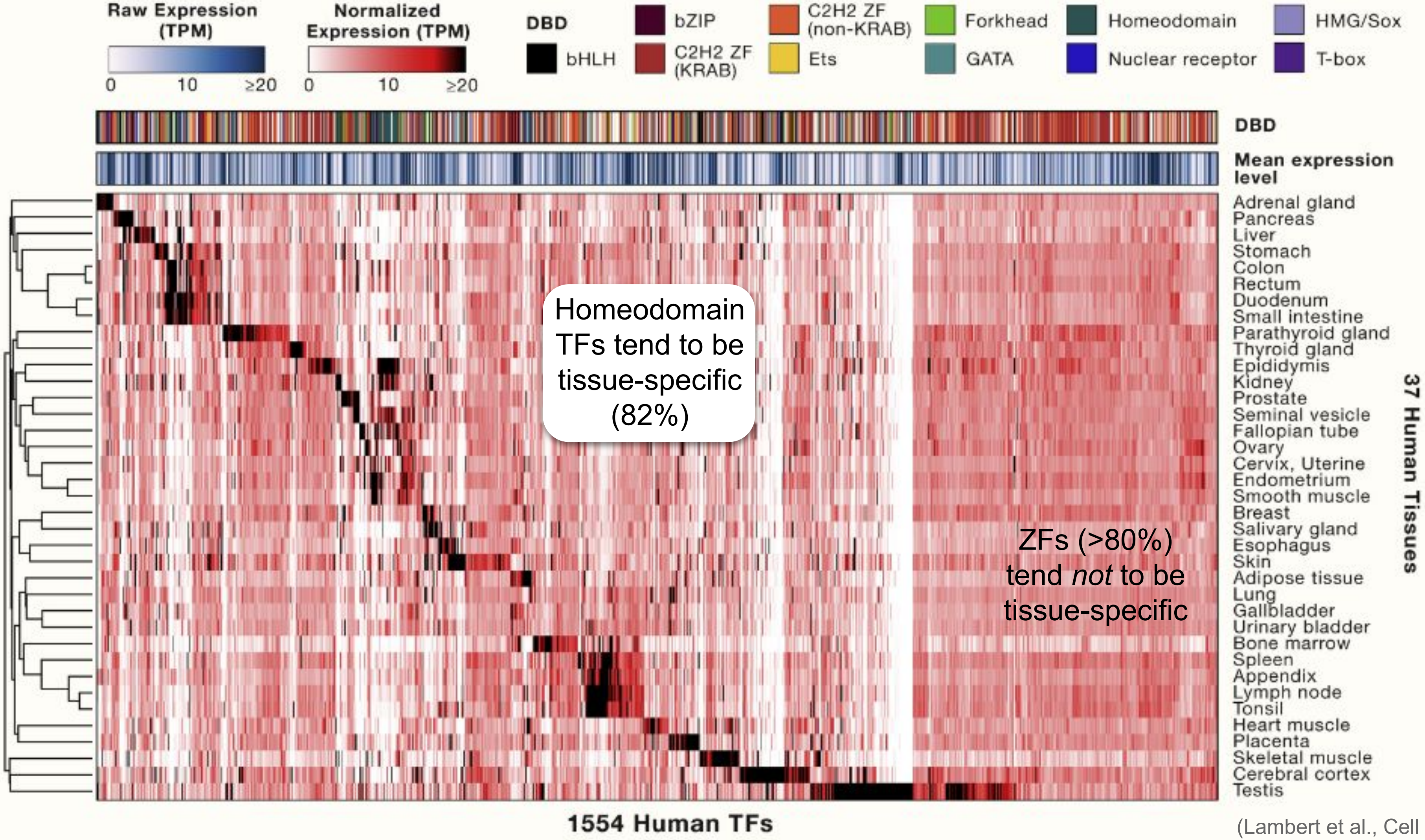


Transcription factors are highly conserved

DNA binding domains show much higher conservation than effector domains



(Soto et al., Molecular Cell 2021)



Homeodomain TFs tend to be tissue-specific (82%)

ZFs (>80%) tend *not* to be tissue-specific

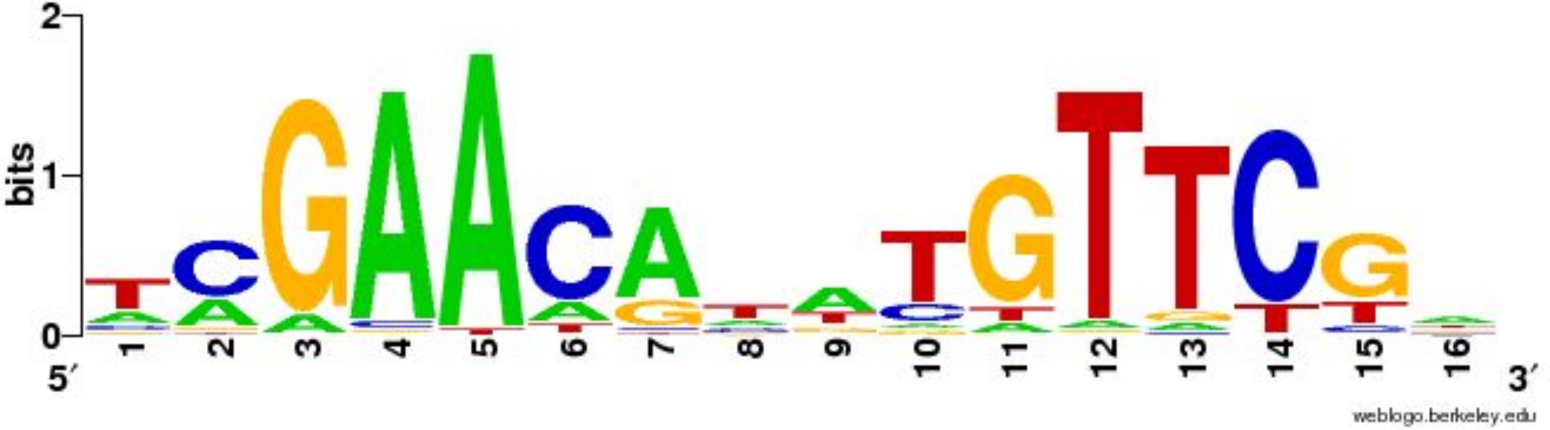
(Lambert et al., Cell 2018)

Sequence-specificity

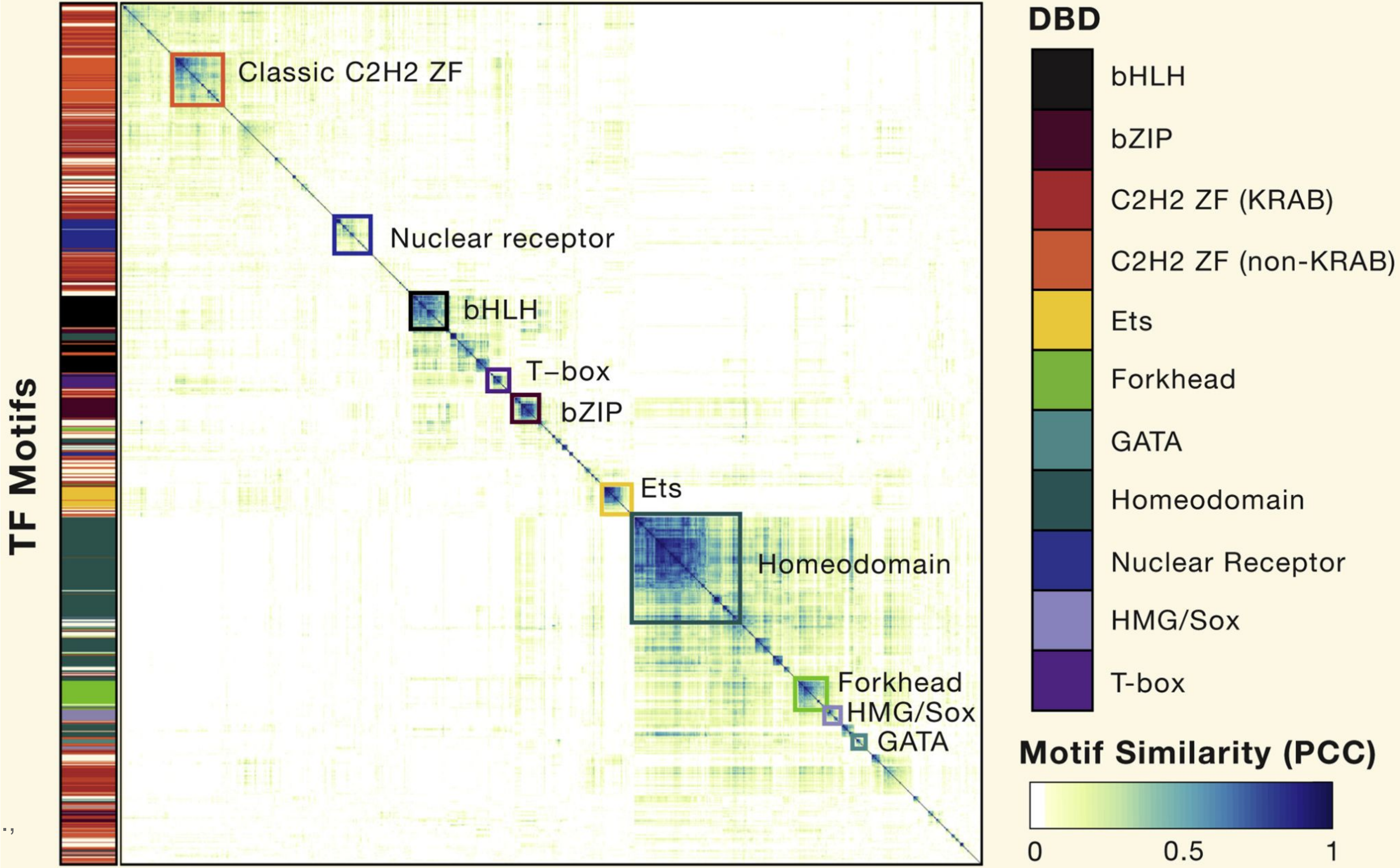
E.g. The LexA bacterial TF recognizes the consensus sequence

5'-GAACAnnTGTTC-3'

-log2(p-value)

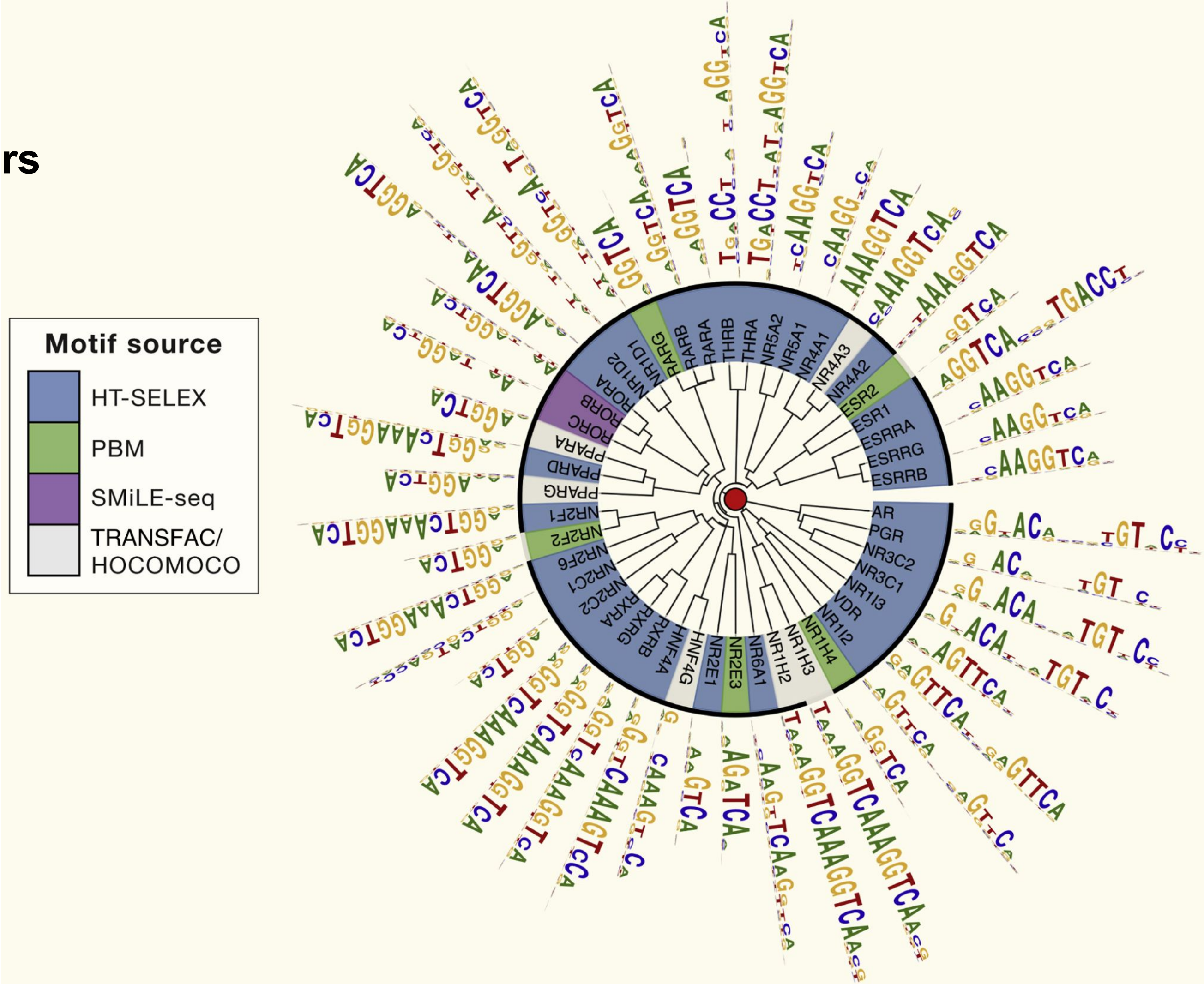


weblogo.berkeley.edu



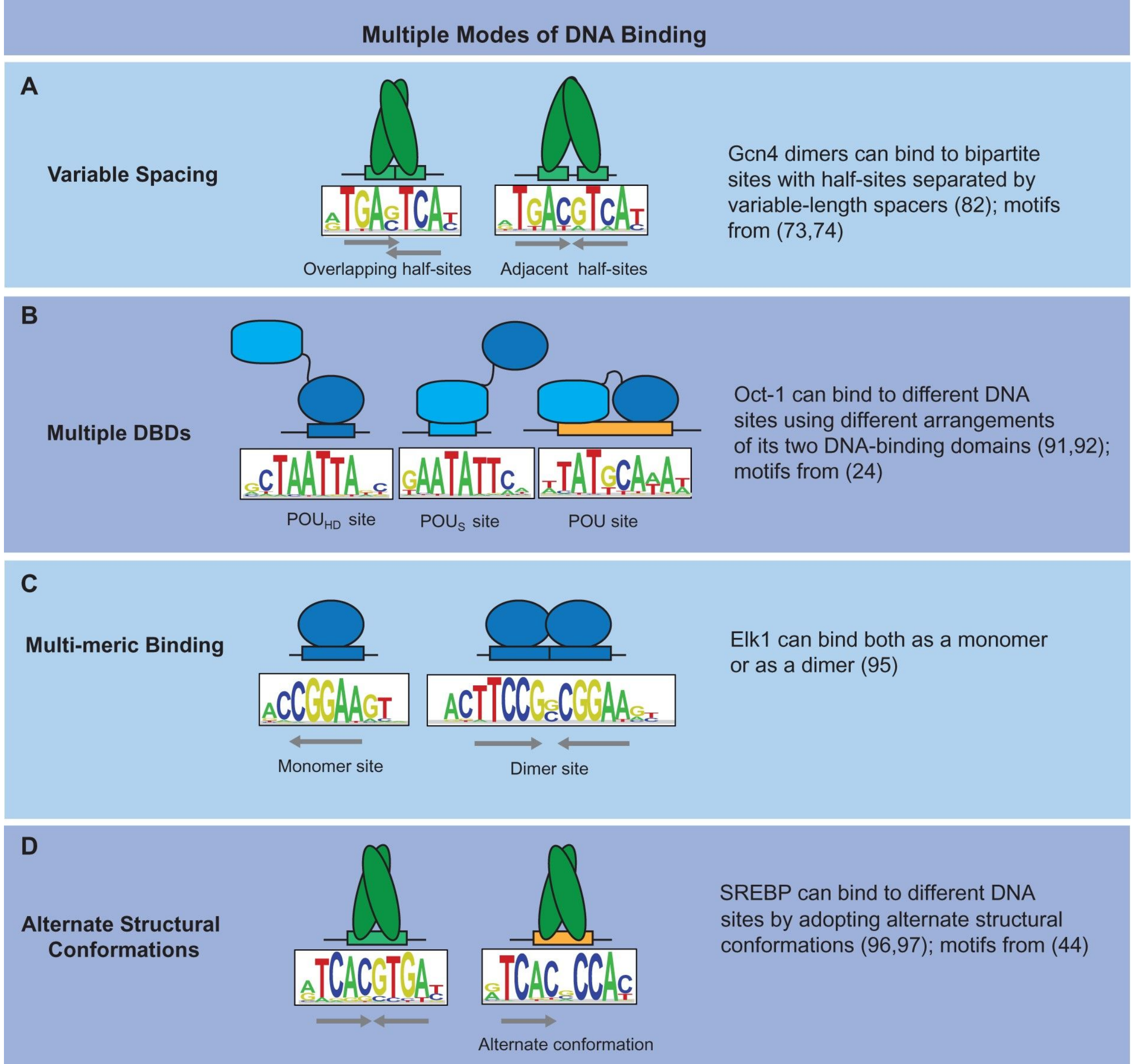
(Lambert et al., Cell 2018)

An example of TF motif degeneracy:
Nuclear hormone receptors



(Lambert et al., Cell 2018)

Variations in DNA binding specificity



Gcn4 dimers can bind to bipartite sites with half-sites separated by variable-length spacers (82); motifs from (73,74)

Oct-1 can bind to different DNA sites using different arrangements of its two DNA-binding domains (91,92); motifs from (24)

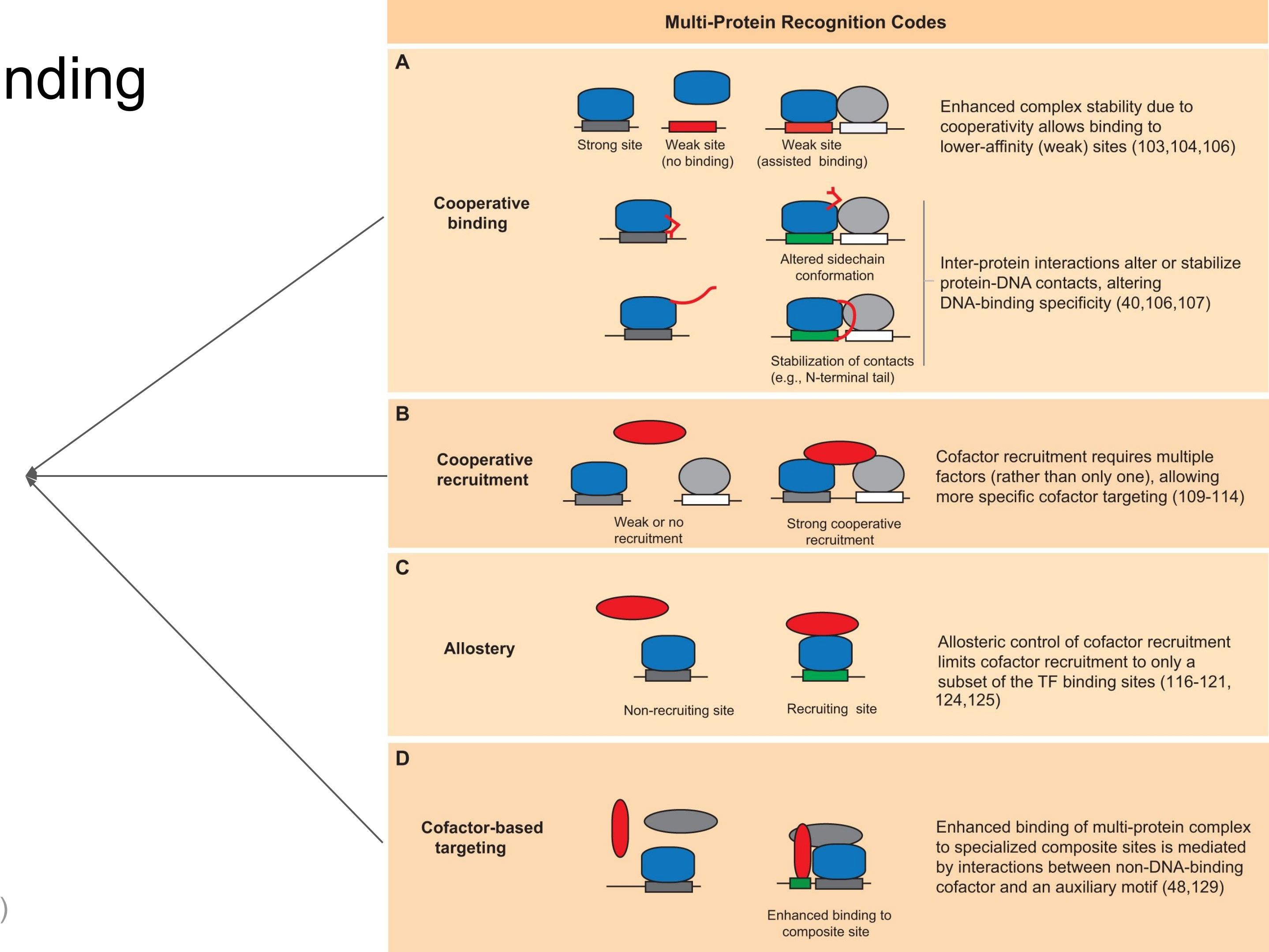
Elk1 can bind both as a monomer or as a dimer (95)

SREBP can bind to different DNA sites by adopting alternate structural conformations (96,97); motifs from (44)

(Siggers and Gordân, NAR 2014)

Cooperative binding

Highly combinatorial binding of TFs



Enhanced complex stability due to cooperativity allows binding to lower-affinity (weak) sites (103,104,106)

Inter-protein interactions alter or stabilize protein-DNA contacts, altering DNA-binding specificity (40,106,107)

Cofactor recruitment requires multiple factors (rather than only one), allowing more specific cofactor targeting (109-114)

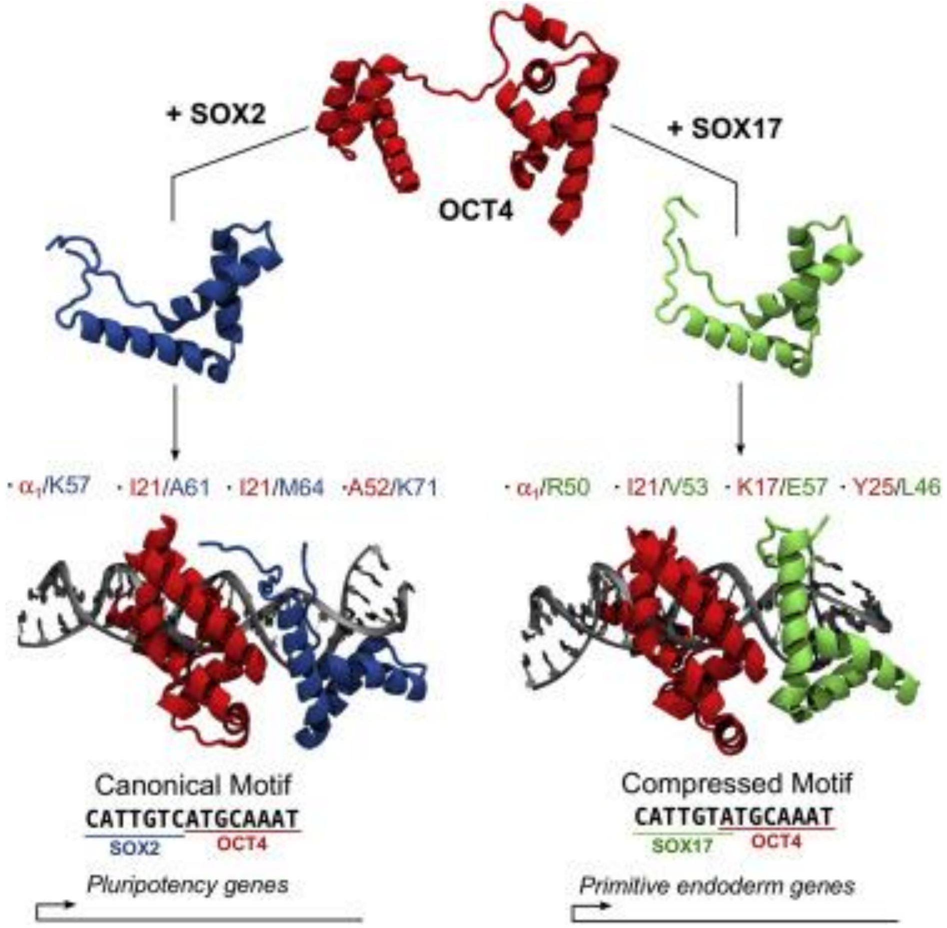
Allosteric control of cofactor recruitment limits cofactor recruitment to only a subset of the TF binding sites (116-121, 124,125)

Enhanced binding of multi-protein complex to specialized composite sites is mediated by interactions between non-DNA-binding cofactor and an auxiliary motif (48,129)

(Siggers and Gordân, NAR 2014)

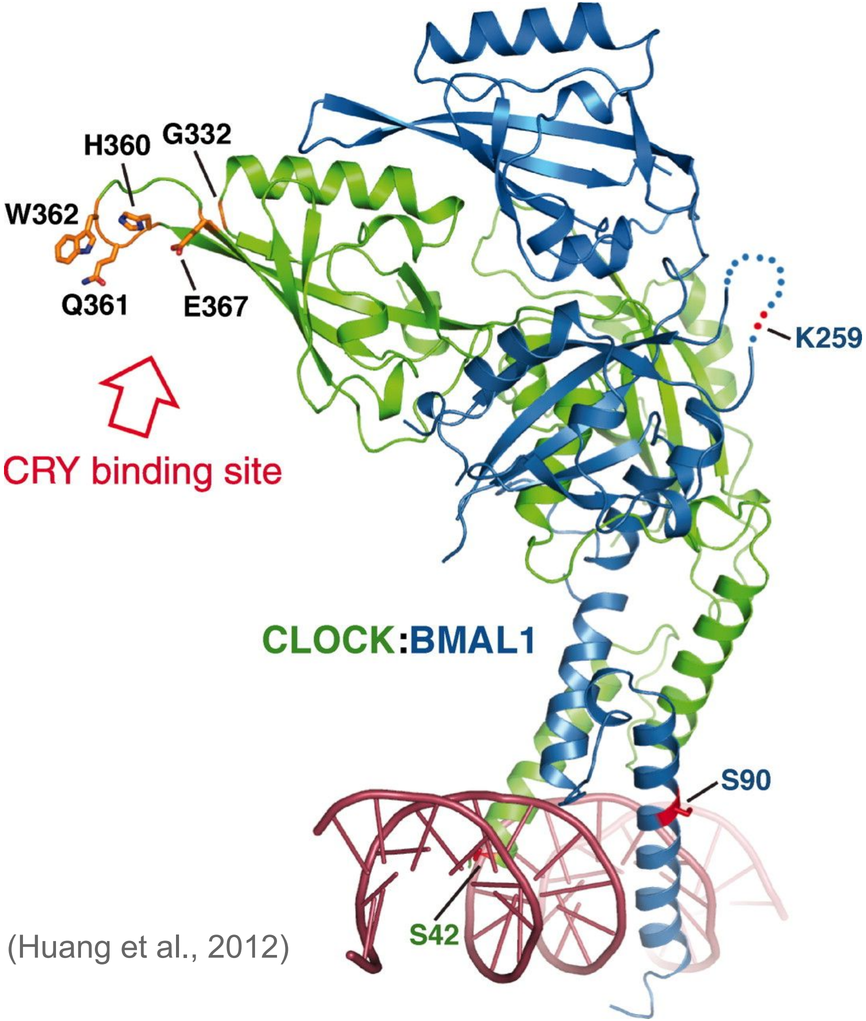
Two examples of Cooperative binding

OCT4 (POU5f1) binding upon differentiation

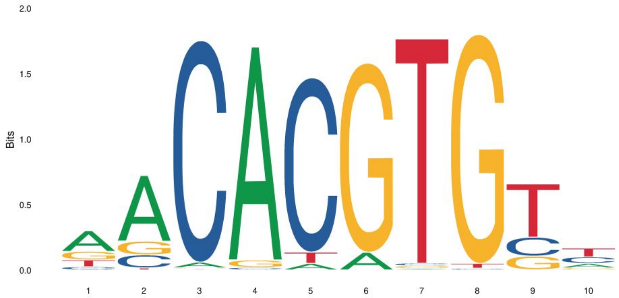

+ SOX2
+ SOX17
OCT4
· α_1/K57 · I21/A61 · I21/M64 ·A52/K71
· α_1/R50 · I21/V53 · K17/E57 · Y25/L46
Canonical Motif
CATTGTCATGCAAAT
SOX2
OCT4
Pluripotency genes
Compressed Motif
CATTGTATGCAAAT
SOX17
OCT4
Primitive endoderm genes

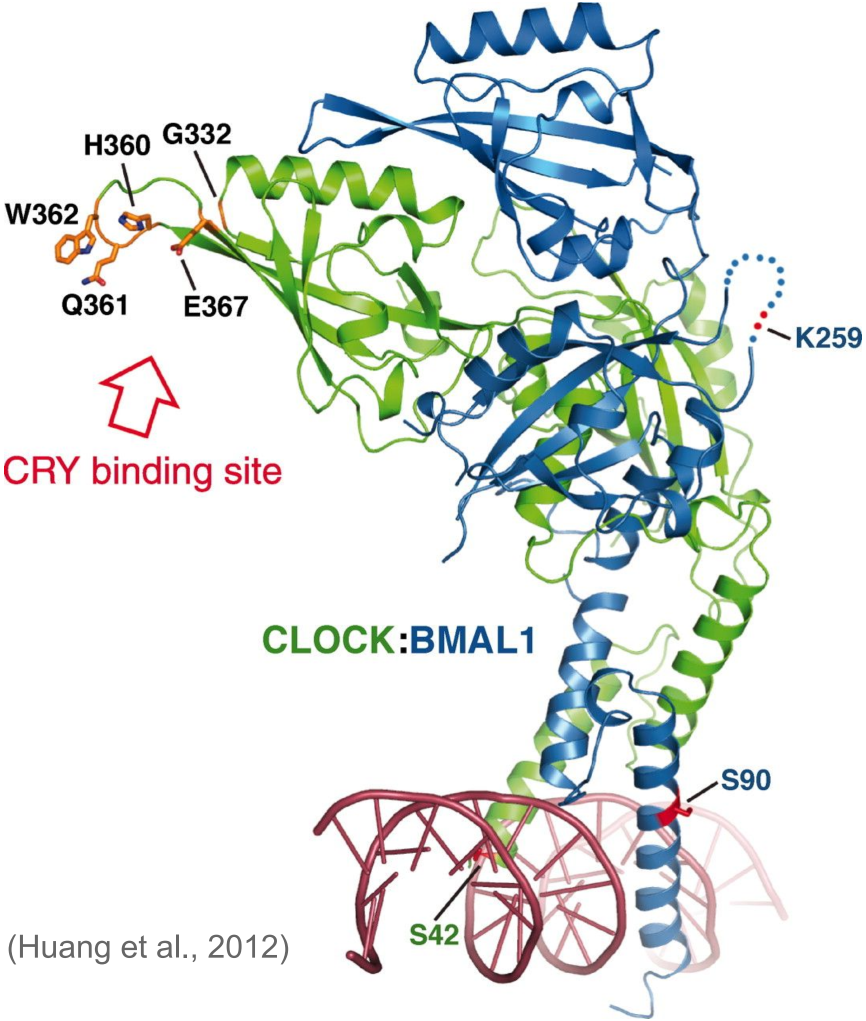
(Merino et al., *Structure* 2014)

Clock-Bmal-Cry during circadian rythm

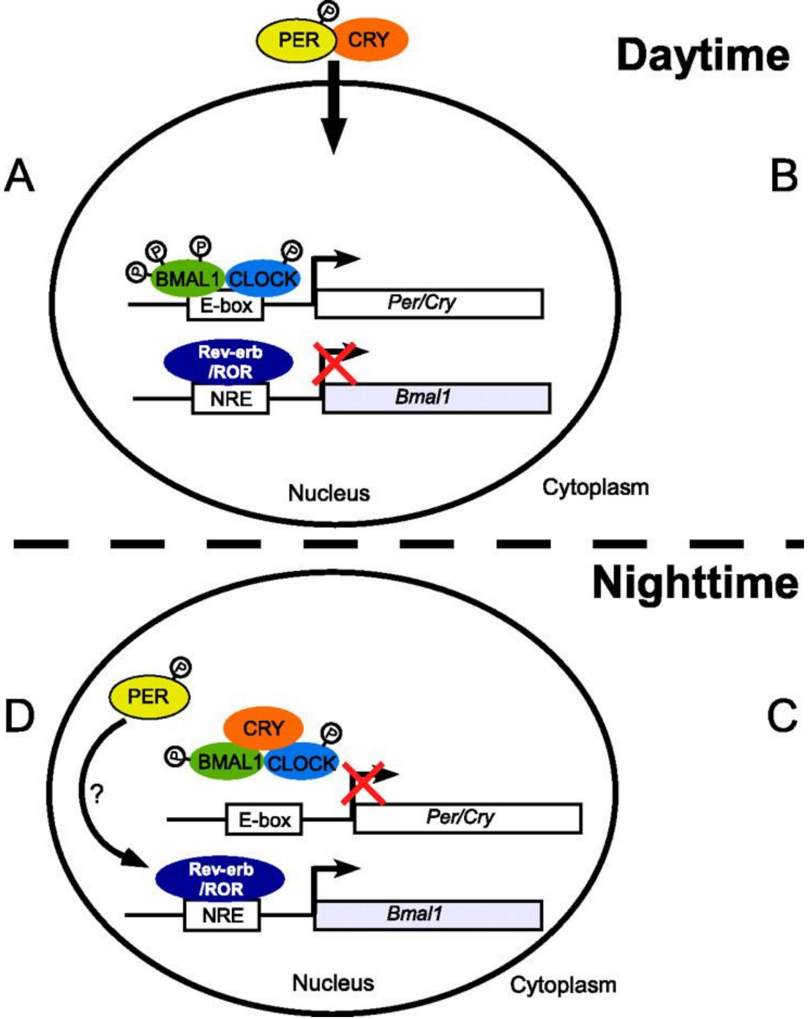


(Huang et al., 2012)





(Huang et al., 2012)



(Adapted from Ye et al., JBC 2011)

Motif analysis

- **Motif discovery** aims at finding **new** motifs that are enriched in a set of sequences (e.g. peaks) versus a background
  - Example method: MEME (Meme suite)
  - Bioconductor method: `rGADEM` package (see also the `memes` R package)

- **Motif enrichment** analysis aims at finding **known** motifs that are enriched in a set of sequences (e.g. peaks) versus a background
  - Example method: AME (Meme suite)
  - Bioconductor method: `PWMEnrich` package

- **Motif scanning** aims at finding the **occurrences of known** motifs in a set of sequences (methodologically fairly simple – which method doesn’t matter much)
  - Bioconductor method: `motifmatchr`
  - (other options are the `TFBSTools` R package and FIMO of the Meme suite)

Genetic variation at TF binding sites

- Genetic variation at TF binding sites can affect the binding of the protein, and hence impact development and health
- Nevertheless, while most coding sequences show evidence of **evolutionary constraint** (e.g. purifying selection), only a small fraction of TF binding sites (11.6% of footprints) show evidence of constraint – the vast majority appears to be evolving neutrally

(Vierstra et al., Nature 2020)

- This suggests a degree of (at least partial) redundancy between regulatory elements

Assignment

- Choose a transcription factor, e.g. CREB1, REST, GATA5, EGR1, GCR
  (or any of your choice that has a motif and available ChIPseq data)

- Download the peaks for that factor (whatever organism/cell type, just make sure you use the corresponding genome!)

- Identify the instances of the factor's motif

- Answer the following questions:
  - Of all the peaks, what proportion contains a motif for the factor?
    - Expected form of an answer: of the XX peaks, XX (XX%) contain a motif
  - Of all instances of that motif in the genome (or in one chromosome), what proportion is bound by the factor (i.e. has a peak)?
    - Expected form of an answer: of the XX motif instances, XX (XX%) overlap a peak

Don't forget to *render* your markdown and push it as `assignment.html` !