RBP database: the ENCODE eCLIP resource for RNA binding protein targets

Eric Van Nostrand
elvannostrand@ucsd.edu
Yeo Lab, UCSD
06/08/2016

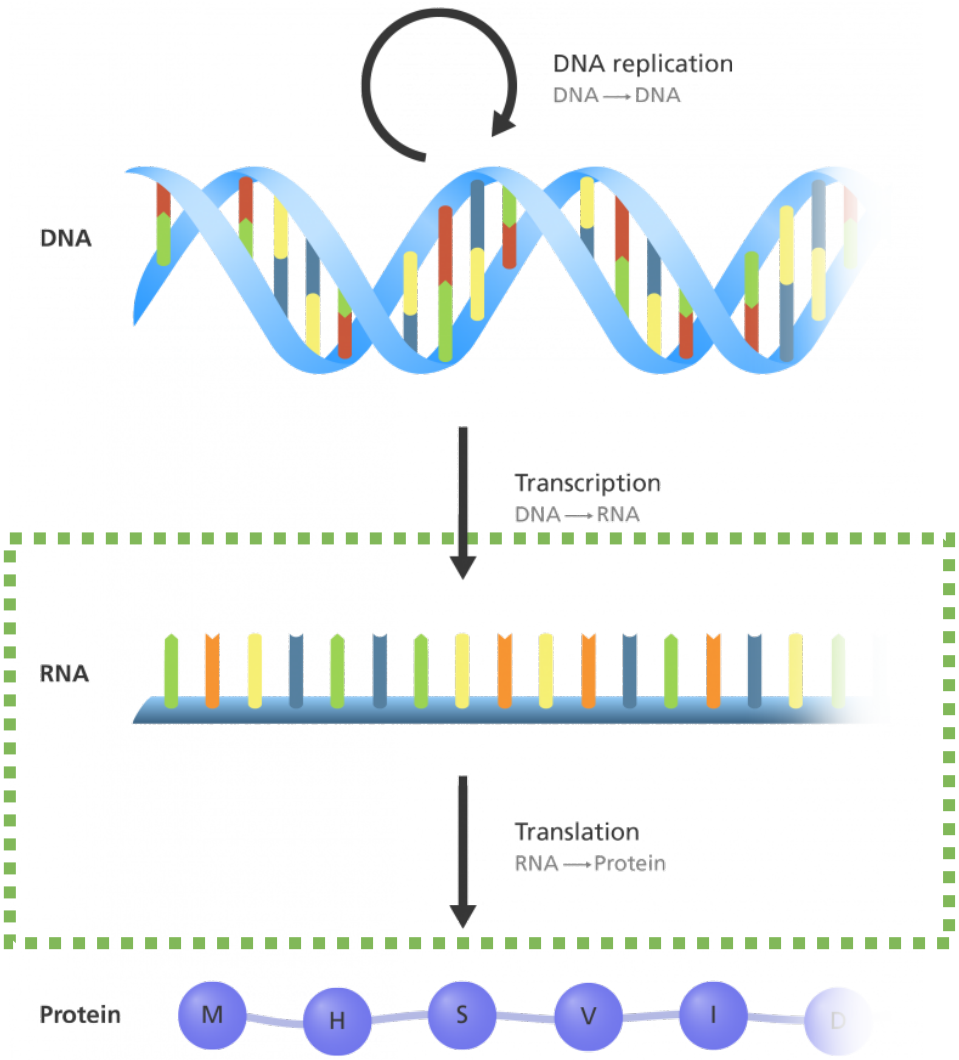
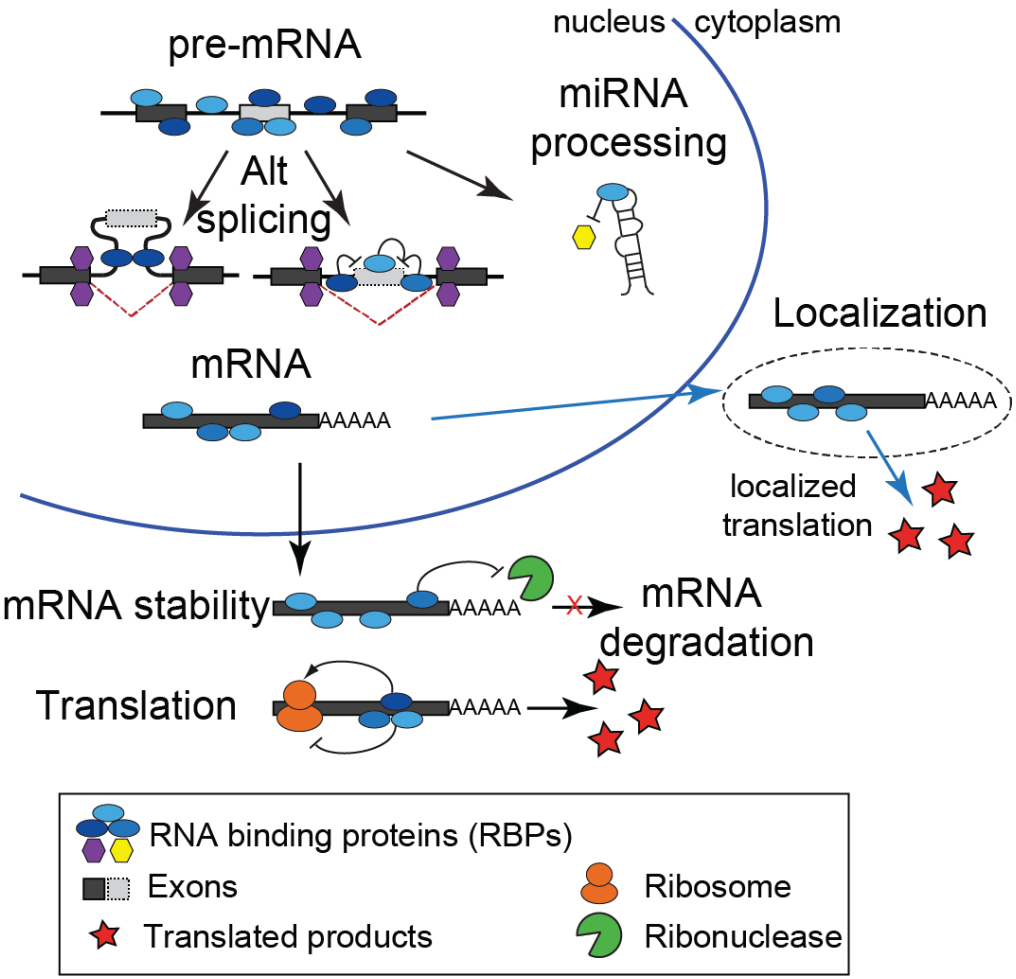


Image adapted from *Genome Research Limited*

Each step of RNA processing is highly regulated



- RNA binding proteins (RBPs) act as trans factors to regulate RNA processing steps
- Estimated >1000 RBPs in human
- RNA processing plays critical roles in development and human physiology
- Mutation or alteration of RNA binding proteins plays critical roles in disease

Stephanie Huelga

ENCORE: ENCODE RNA regulation group

ENCORE

250 RNA Binding Proteins

K562 & HepG2 cells

Yeo CLIP-Seq

Fu (ChIP-Seq)

Bind-N-Seq

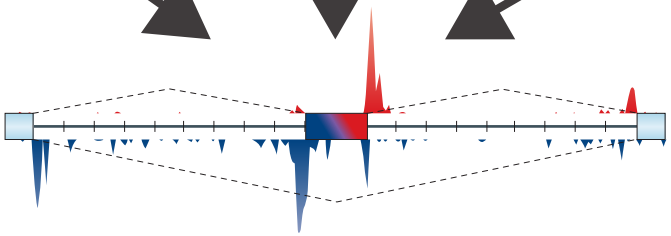
Burge

RNAi & RNA-Seq

Graveley

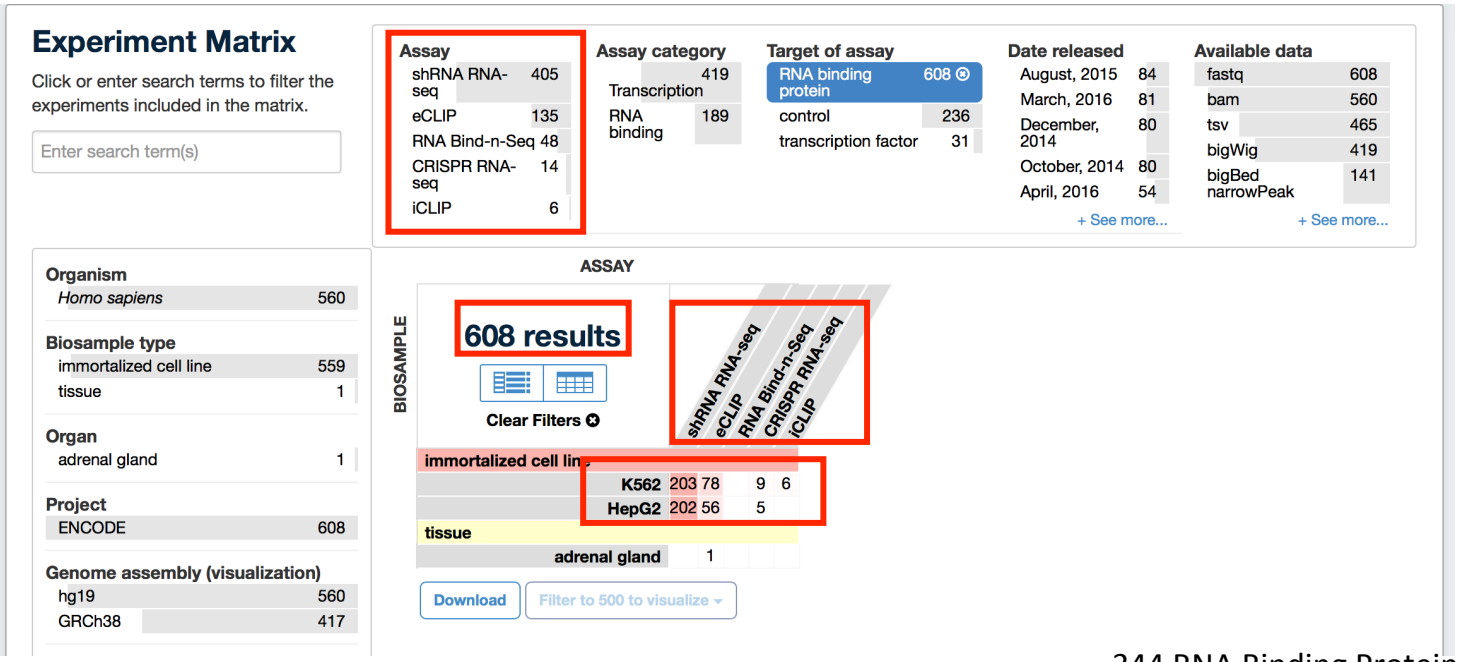
RBP Localization

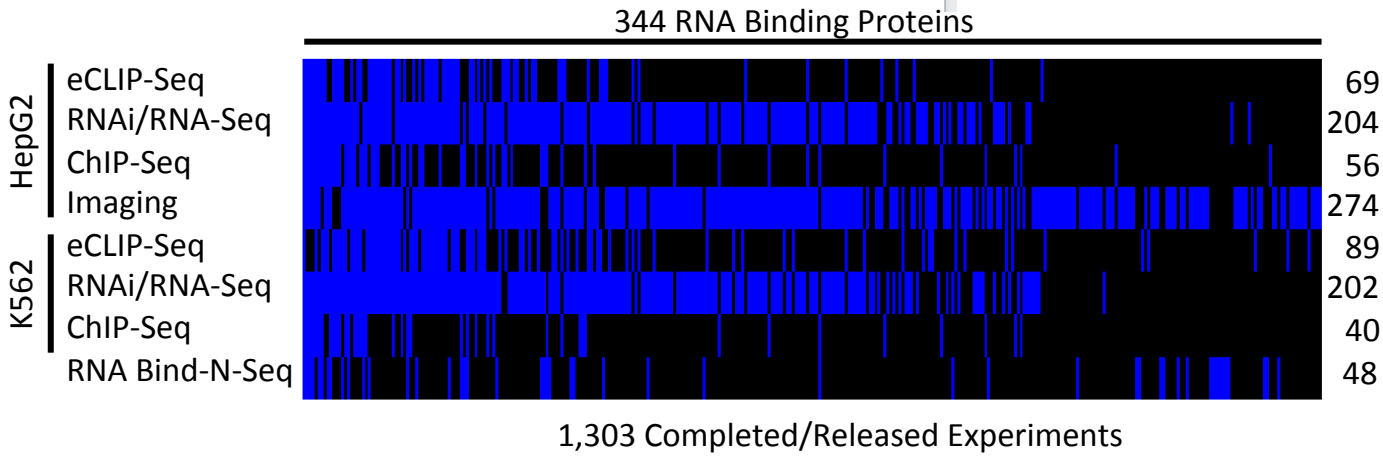
Lécuyer



RBP Data Production Overview

(Released data only as of 6/8/16)

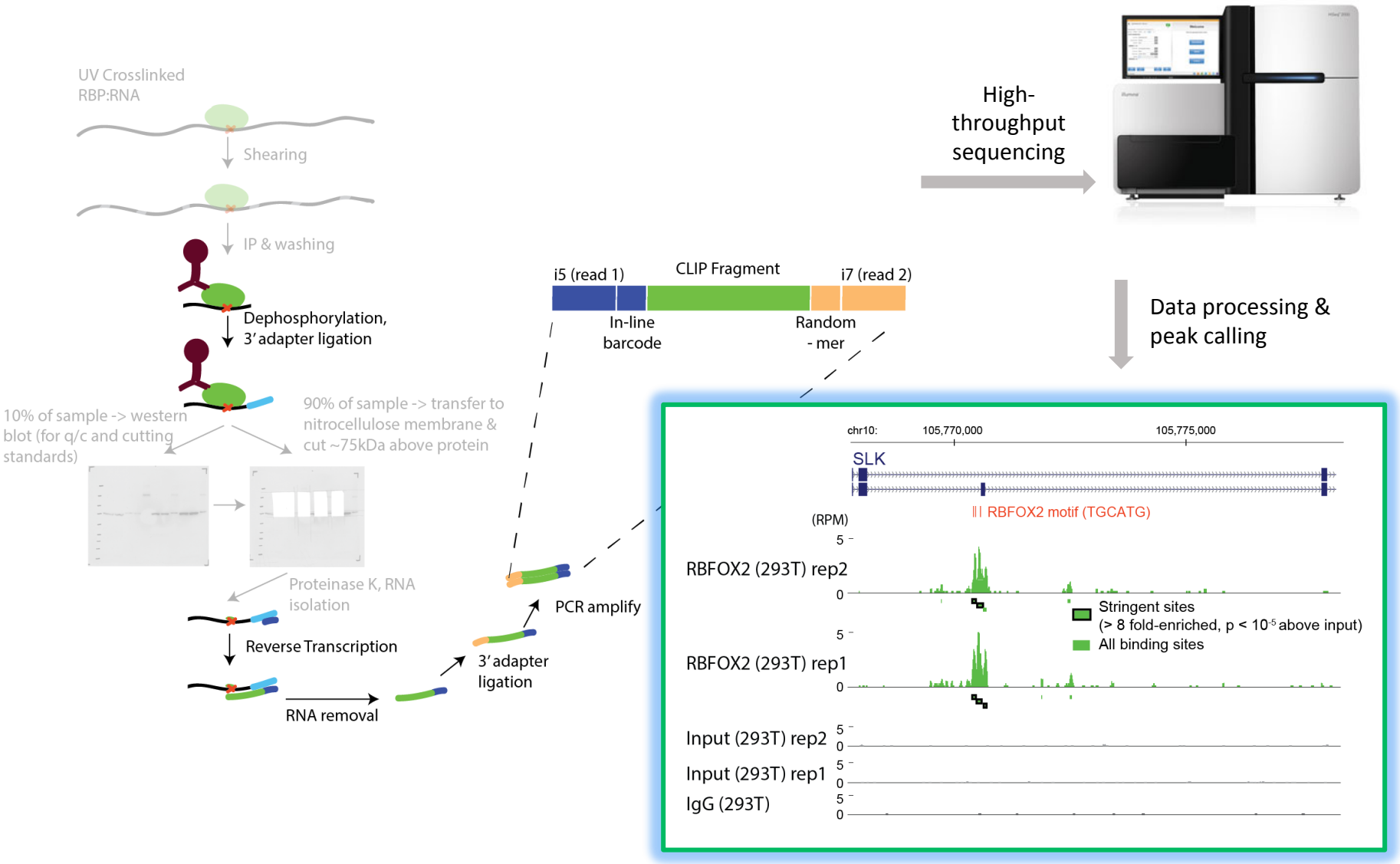




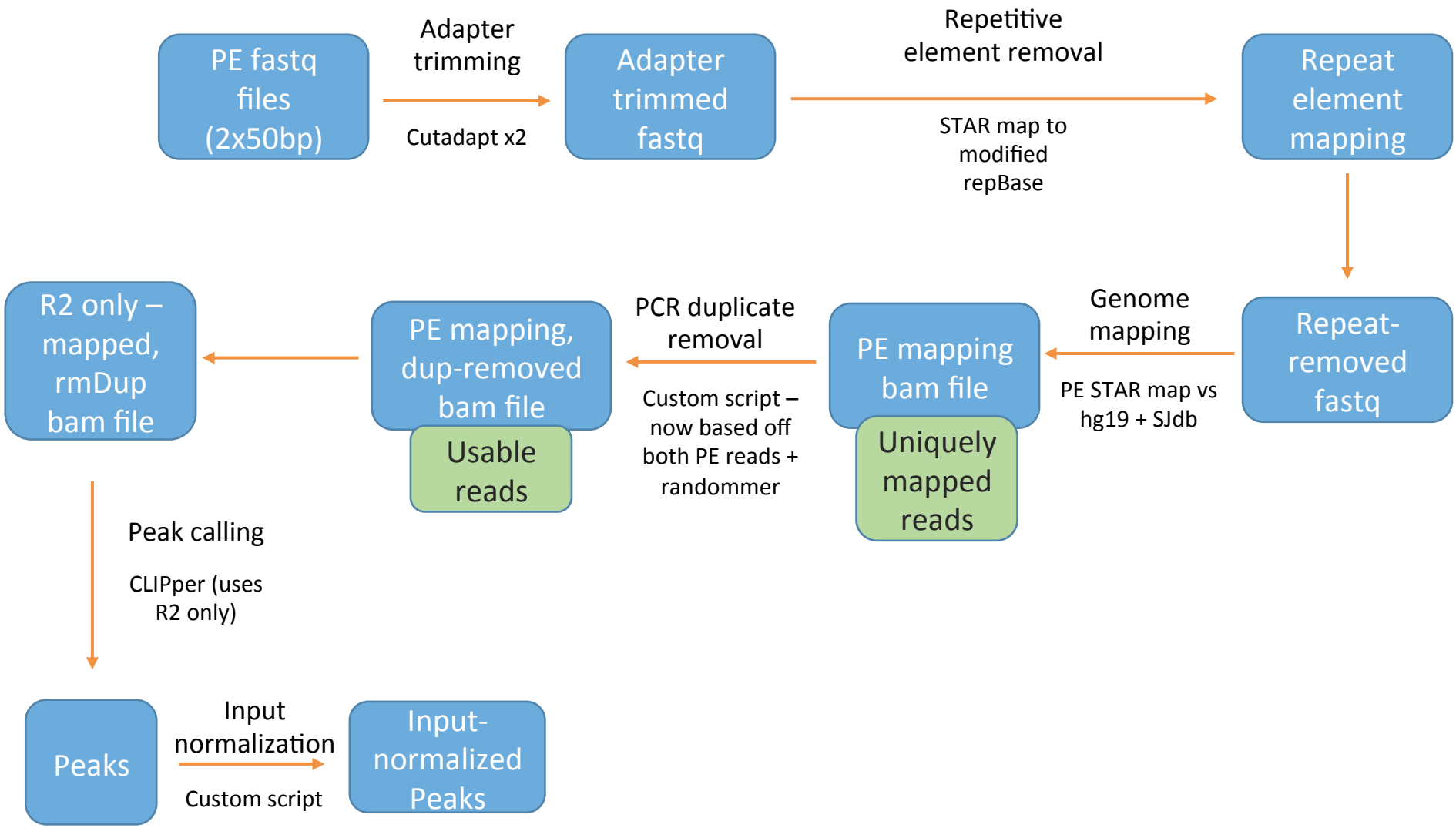
Outline

- eCLIP overview
  - Method outline
  - ENCODE submitted data structure
  - ENCODE eCLIP pipeline walkthrough
- What kinds of analyses can be done?
- Tools coming soon

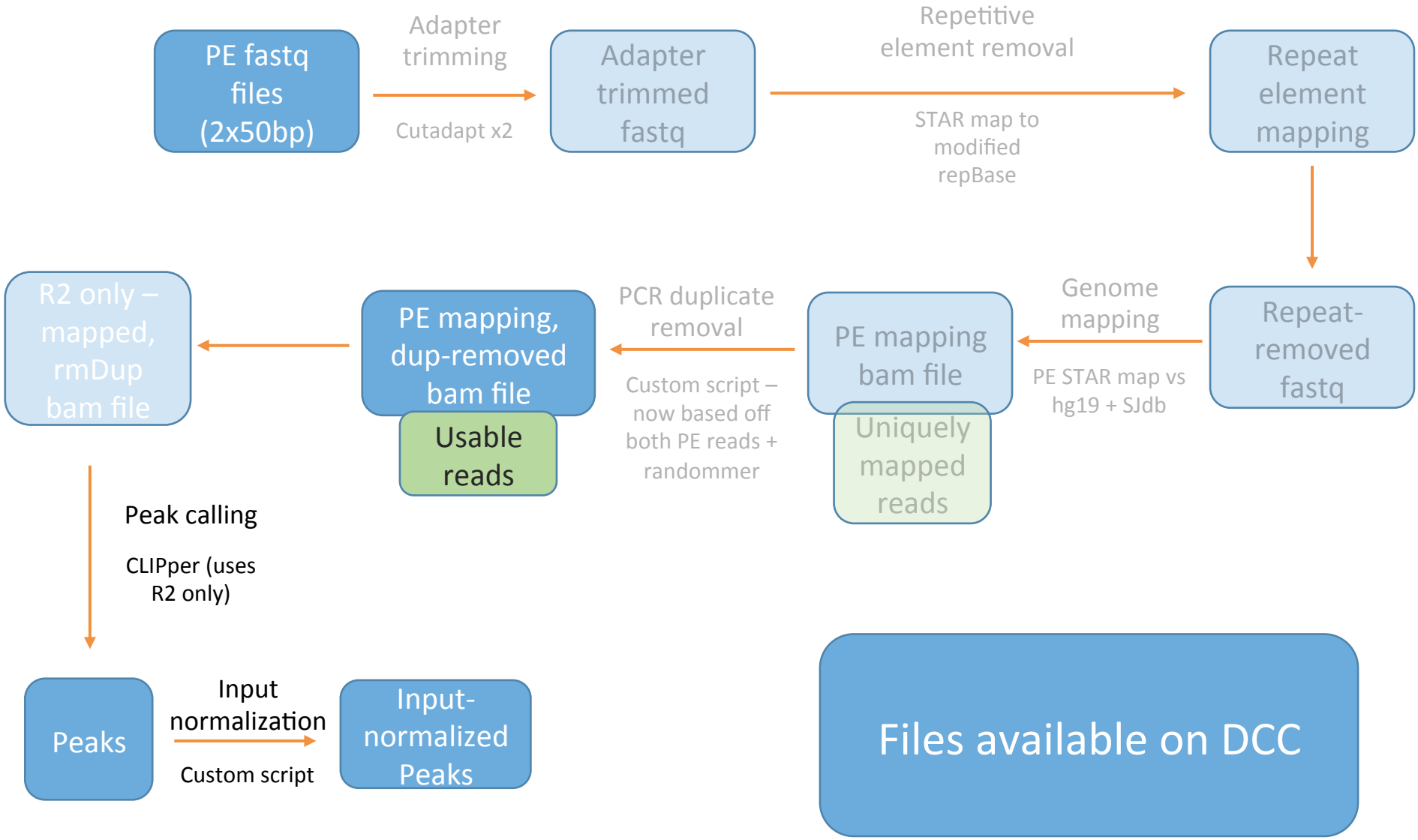
Identification of RNA binding protein targets by eCLIP-seq



eCLIP computational pipeline



eCLIP computational pipeline



ENCODE Data Encyclopedia Materials & Methods Help Search...

Experiment Matrix

Click or enter search terms to filter the experiments included in the matrix.

Enter search term(s)

Assay
- eCLIP 270
- RNA Bind-n-Seq 106
- RIP-seq 43
- iCLIP 6

Assay category
- RNA binding 270

Target of assay
- RNA binding protein 135
- control 135
- transcription factor 7

Date released
- April, 2016 108
- November, 2015 87
- July, 2015 65
- December, 2015 10

Available data
- bam 270
- fastq 270
- bigBed narrowPeak 135
- bed narrowPeak 53

Organism
- *Homo sapiens* 270

Biosample type
- immortalized cell line 268
- tissue 2

Organ
- adrenal gland 2

Project
- ENCODE 270

Genome assembly (visualization)
- hg19 135

ASSAY

BIOSAMPLE

270 results

Clear Filters

eCLIP

immortalized cell line	
K562	156
HepG2	112
tissue	
adrenal gland	2

Download Visualize

ENCODE UNIVERSITY OF CALIFORNIA SANTA CRUZ Stanford University

Contact Terms of Use Submitter sign-in

ENCODE Data Encyclopedia Materials & Methods Help

RBFOX2

Data Type

Dataset	7
Experiment	7
Biosample	4
AntibodyLot	2
Publication	2

+ See more...

Showing 18 of 18 results

RBFOX2 (*Homo sapiens*) — Target

External resources: ENSEMBL:ENSG00000100320 HGNC:FOX2 HGNC:RBM9 GeneID:23543 HGNC:HRNBP2 UniProtKB:O43251 HGNC:RTA

RBFOX2 eCLIP mock input (*Homo sapiens*) — Target

External resources: *None submitted*

RNA Bind-n-Seq — Experiment

Target: RBFOX2
Lab: Chris Burge, MIT
Project: ENCODE

ENCSR441HLP
released

RBFOX2 (*Homo sapiens*) — Antibody

Source: GeneTex
Product ID / Lot ID: GTX116327 / 40555

ENCAB507HJJ

RBFOX2 (*Homo sapiens*) — Antibody

Source: Bethyl Labs
Product ID / Lot ID: A300-864A / 2

ENCAB592TEY

K562 (*Homo sapiens*, adult 53 year) — Biosample

Type: immortalized cell line
Summary: Homo sapiens K562 immortalized cell line transient RNAi knockdown shRNA...
RNAi target: RBFOX2
Culture harvest date: 2015-03-05
Source: ATCC

ENCBS677KBE
released

ENCODE Data Encyclopedia Materials & Methods Help RBFOX2

Clear Filters

Assay category
RNA binding 2

Assay
RNA Bind-n-Seq 3
eCLIP 2
shRNA RNA-seq 2

Project
ENCODE 2

RFA
ENCODE3 2

Experiment status
released 2

Genome assembly (visualization)
hg19 1

Organism
Homo sapiens 2

Target of assay
RNA binding protein 1
control 1

Biosample type
immortalized cell line 2

Life stage
child 2

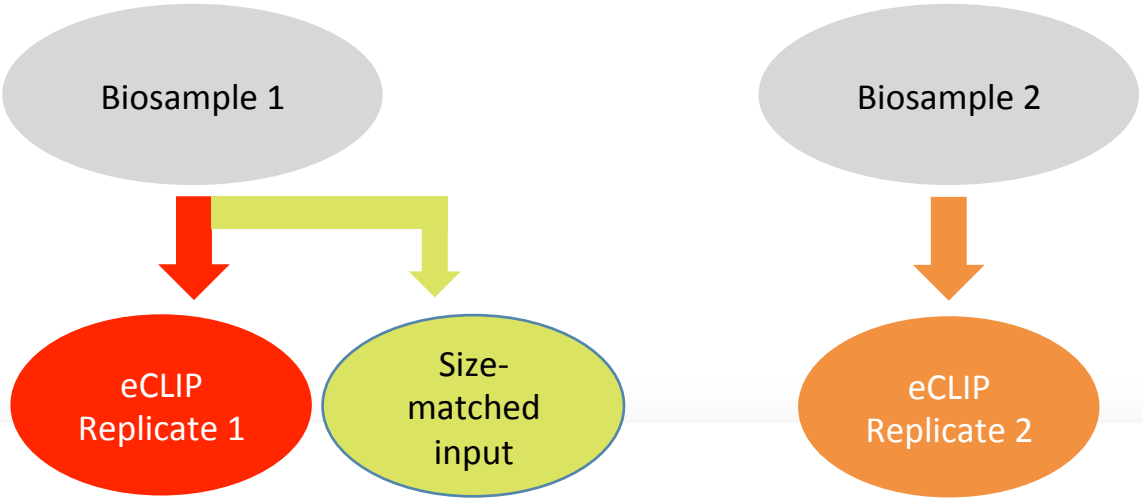
Pipeline

Showing 2 of 2 results

Download Visualize

eCLIP of HepG2
Homo sapiens, child 15 year
Target: RBFOX2 eCLIP mock input
Lab: Gene Yeo, UCSD
Project: ENCODE
Experiment
ENCSR799EKA
released

eCLIP of HepG2
Homo sapiens, child 15 year
Target: RBFOX2
Lab: Gene Yeo, UCSD
Project: ENCODE
Experiment
ENCSR987FTF
released



Experiment summary for ENCSR987FTF

Status: **released**

Summary

Assay:	eCLIP
Target:	RBFOX2
Biosample summary:	HepG2 (*Homo sapiens*, child 15 year male)
Biosample Type:	immortalized cell line
Replication type:	isogenic
Description:	eCLIP experiment on HepG2 against RBFOX2
Nucleic acid type:	RNA
Size range:	175-300
Lysis method:	see document
Extraction method:	see document
Fragmentation method:	see document
Size selection method:	agarose gel extraction
Platform:	HiSeq 2000
Controls:	ENCSR799EKA

Attribution

ENCODE PHASE 3

Lab:	Gene Yeo, UCSD
Award PI:	Brenton Graveley, UConn
Project:	ENCODE
Aliases:	gene-yeo:204
Date released:	2015-07-15

Isogenic replicates

Isogenic replicate	Technical replicate	Summary	Biosample	Antibody	Library
1	1	Homo sapiens HepG2 immortalized cell line	ENCBS547JWV	ENCAB592TEY	ENCLB180GIG
2	1	Homo sapiens HepG2 immortalized cell line	ENCBS537ADD	ENCAB592TEY	ENCLB696TLV

File summary

Visualize Data

Raw data files

Accession	File type	Biological replicate	Library	Run type	Read	Lab	Date added	File size	Audit status	File status
ENCFF647KDW	fastq	1	ENCLB180GIG	PE 50nt	R2	Gene Yeo, UCSD	2016-03-22	335 MB	✓	released
ENCFF172GUS	fastq	1	ENCLB180GIG	PE 50nt	R1	Gene Yeo, UCSD	2016-03-22	313 MB	✓	released
ENCFF289OFA	fastq	2	ENCLB696TLV	PE 50nt	R2	Gene Yeo, UCSD	2016-03-22	338 MB	✓	released
ENCFF591SSP	fastq	2	ENCLB696TLV	PE 50nt	R1	Gene Yeo, UCSD	2016-03-22	315 MB	✓	released

R1 + R2 fastq files

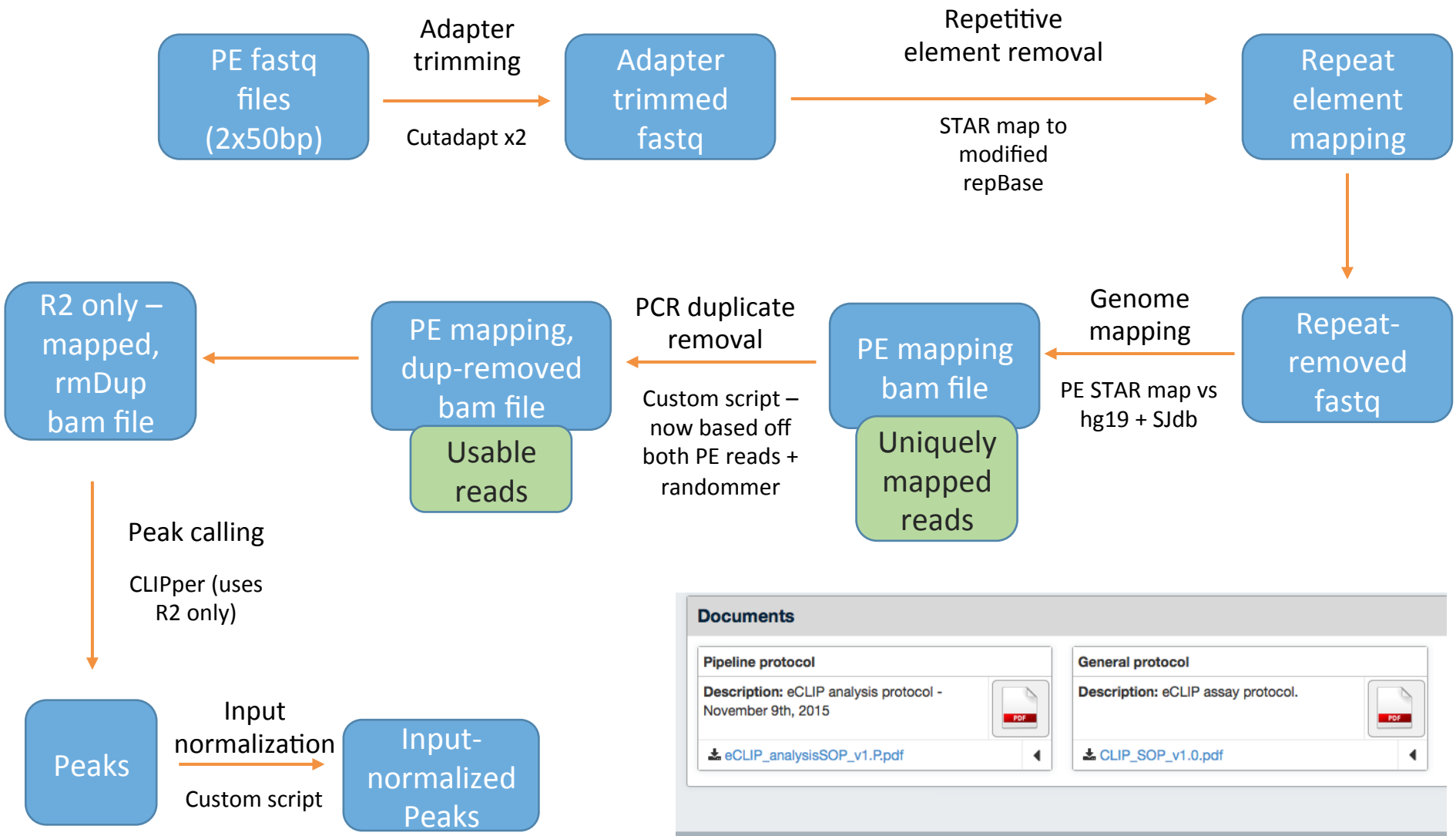
Processed data files

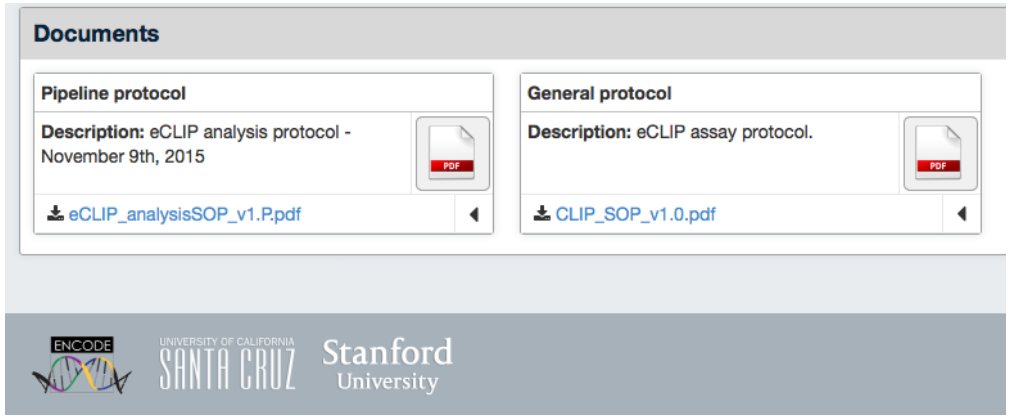
Accession	File type	Output type	Biological replicate	Mapping assembly	Lab	Date added	File size	Audit status	File status
ENCFF735HJV	bed narrowPeak	peaks	1	hg19	Gene Yeo, UCSD	2016-04-05	784 kB		in progress
ENCFF475KXE	bigBed narrowPeak	peaks	1	hg19	Gene Yeo, UCSD	2016-03-24	2.67 MB	✓	released
ENCFF994WPX	bam	alignments	1	hg19	Gene Yeo, UCSD	2016-03-23	365 MB	✓	released
ENCFF030USB	bed narrowPeak	peaks	2	hg19	Gene Yeo, UCSD	2016-04-05	665 kB		in progress
ENCFF026CVE	bigBed narrowPeak	peaks	2	hg19	Gene Yeo, UCSD	2016-03-24	2.7 MB	✓	released
ENCFF154BQS	bam	alignments	2	hg19	Gene Yeo, UCSD	2016-03-23	359 MB	✓	released

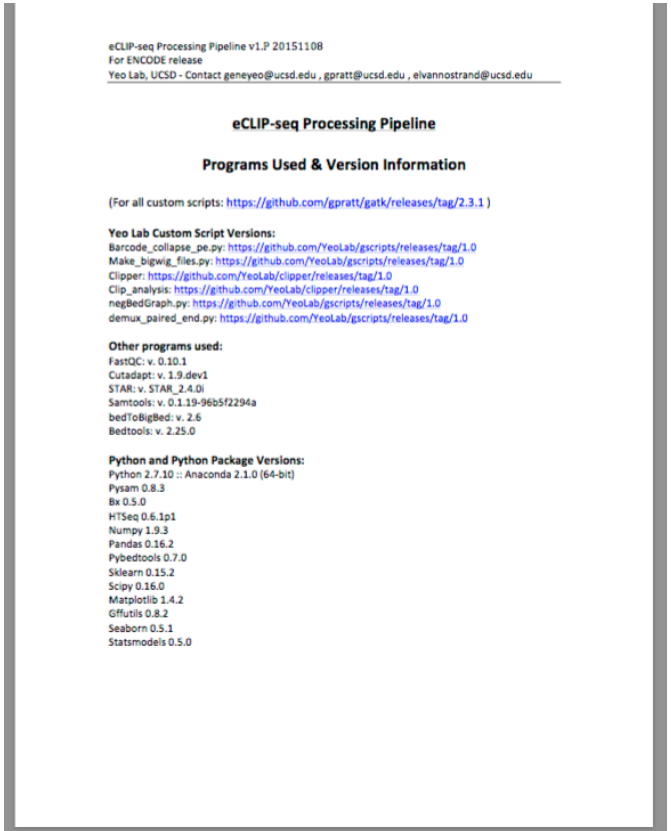
Input-normalized peaks

Paired-end mapping (STAR)

eCLIP computational pipeline







eCLIP-seq Processing Pipeline v1.P 20151108
For ENCODE release
Yeo Lab, UCSD - Contact geneyeo@ucsd.edu , gpratt@ucsd.edu , elvannostrand@ucsd.edu

eCLIP-seq Processing Pipeline

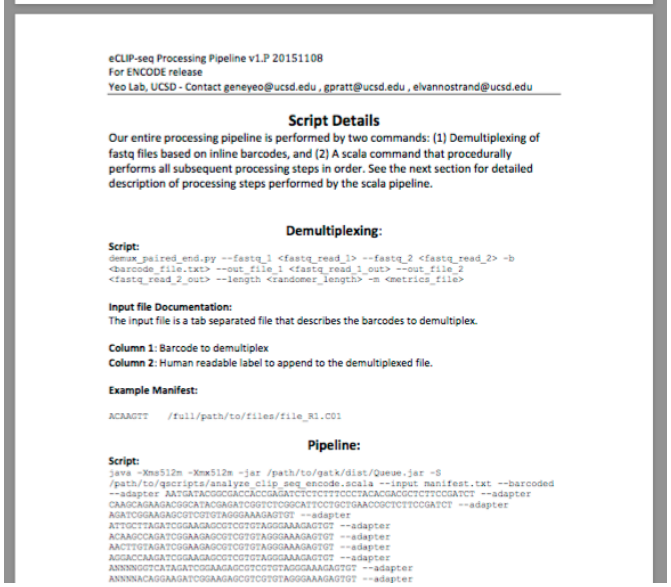
Programs Used & Version Information

(For all custom scripts: https://github.com/gpratt/gatk/releases/tag/2.3.1)

Yeo Lab Custom Script Versions:
Barcode_collapse_pe.py: https://github.com/YeoLab/gscripts/releases/tag/1.0
Make_bigwig_files.py: https://github.com/YeoLab/gscripts/releases/tag/1.0
Clipper: https://github.com/YeoLab/clipper/releases/tag/1.0
Clip_analysis: https://github.com/YeoLab/clipper/releases/tag/1.0
negBedGraph.py: https://github.com/YeoLab/gscripts/releases/tag/1.0
demux_paired_end.py: https://github.com/YeoLab/gscripts/releases/tag/1.0

Other programs used:
FastQC: v. 0.10.1
Cutadapt: v. 1.9.dev1
STAR: v. STAR_2.4.0i
Samtools: v. 0.1.19-96b5f2294a
bedToBigBed: v. 2.6
Bedtools: v. 2.25.0

Python and Python Package Versions:
Python 2.7.10 :: Anaconda 2.1.0 (64-bit)
Pysam 0.8.3
Bx 0.5.0
HTSeq 0.6.1p1
Numpy 1.9.3
Pandas 0.16.2
Pybedtools 0.7.0
Sklearn 0.15.2
Scipy 0.16.0
Matplotlib 1.4.2
Gffutils 0.8.2
Seaborn 0.5.1
Statsmodels 0.5.0



eCLIP-seq Processing Pipeline v1.P 20151108
For ENCODE release
Yeo Lab, UCSD - Contact geneyeo@ucsd.edu , gpratt@ucsd.edu , elvannostrand@ucsd.edu

Script Details

Our entire processing pipeline is performed by two commands: (1) Demultiplexing of fastq files based on inline barcodes, and (2) A scala command that procedurally performs all subsequent processing steps in order. See the next section for detailed description of processing steps performed by the scala pipeline.

Demultiplexing:

Script:

```
demux_paired_end.py --fastq_1 <fastq_read_1> --fastq_2 <fastq_read_2> -b
<barcode_file.txt> --out_file_1 <fastq_read_1_out> --out_file_2
<fastq_read_2_out> --length <randomer_length> -m <metrics_file>
```

Input file Documentation:
The input file is a tab separated file that describes the barcodes to demultiplex.

Column 1: Barcode to demultiplex
Column 2: Human readable label to append to the demultiplexed file.

Example Manifest:

```
ACAAGTT    /full/path/to/files/file_R1.C01
```

Pipeline:

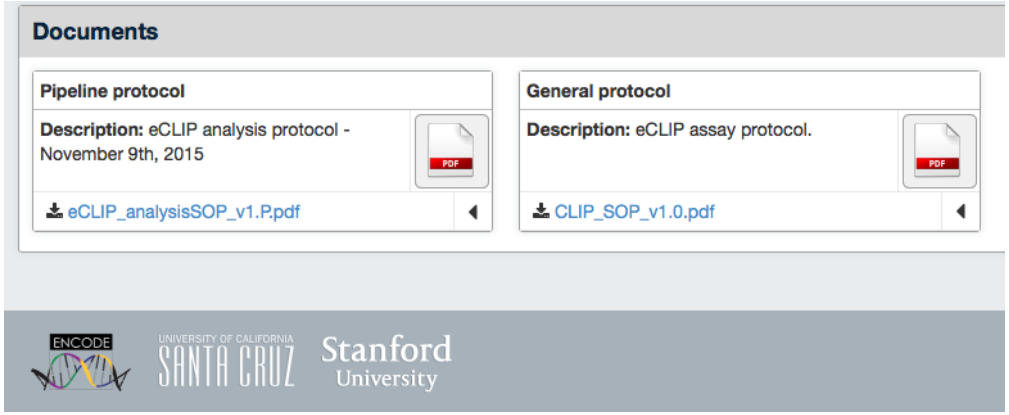
Script:

```
java -Xms512m -Xmx512m -jar /path/to/gatk/dist/Queue.jar -S
/path/to/qscripts/analyze_clip_seq_encode.scala --input manifest.txt --barcoded
--adapter AATGATACGGCGACCACCGAGATCTCTCTTCCCTACACGACGCTCTTCCGATCT --adapter
CAAGCAGAAGACGGCATACGAGATCGGTCTCGGCATTCCTGCTGAACCGCTCTTCCGATCT --adapter
AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT --adapter
ATTGCTTAGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT --adapter
ACAAGCCAGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT --adapter
AACTTGTAGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT --adapter
AGGACCAAGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT --adapter
ANNNNGGTCATAGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT --adapter
ANNNNACAGGAAGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT --adapter
```

- Analysis SOP available at:

https://www.encodeproject.org/documents/dde0b669-0909-4f8b-946d-3cb9f35a6c52/@@download/attachment/eCLIP_analysisSOP_v1.P.pdf

Linked at bottom of each eCLIP experiment:



Demultiplexing
(already has been done for files on ENCODE DCC)

Demultiplexing:

Script:

```
demux_paired_end.py --fastq_1 <fastq_read_1> --fastq_2 <fastq_read_2> -b
<barcode_file.txt> --out_file_1 <fastq_read_1_out> --out_file_2
<fastq_read_2_out> --length <randomer_length> -m <metrics_file>
```

Input file Documentation:

The input file is a tab separated file that describes the barcodes to demultiplex.

Column 1: Barcode to demultiplex
Column 2: Human readable label to append to the demultiplexed file.

Example Manifest:

```
ACAAGTT    /full/path/to/files/file_R1.C01
```

File details: fastq files

- **@CCAAC** = random-mer (first 5 or 10nt of sequenced read2) – has been removed from the 5' end of read2 and appended to read name
- Any in-line barcode has been removed (as part of demultiplexing)

DATASET.R1.fastq.gz:

```
@CCAAC:SN1001:449:HGTN3ADXX:1:1101:1373:1964
1:N:0:1
CAAATGCCCCTGAGGACAAAGCTGCTGCCGGGCCTCTCTCTCTG
+
FFFFFFIIFIIIFIIFIFIFIIIIIIIIIIIIIIIIIIIIIIFI
@CAGAT:SN1001:449:HGTN3ADXX:1:1101:1669:1914
1:N:0:1
TTAGAGACAGGGTCTCGCTCCGTTGCTCAGGCTGGAGTGCAGTG
+
FFFFFFIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
...
```

DATASET.R2.fastq.gz:

```
@CCAAC:SN1001:449:HGTN3ADXX:1:1101:1373:1964
2:N:0:1
GAGAGAGGAGTGGGAAGTTGGGATAGTACCCAGAGAGAGAGGCCCG
+
FFFFFBFFBFBFFFFFIFFFIFFIFIIIIIIFIIIIFFIFIIFFIF
@CAGAT:SN1001:449:HGTN3ADXX:1:1101:1669:1914
2:N:0:1
TTGTACCACTGCACTCCAGCCTGAGCAACGGAGCGAGACCCTGTCT
+
FFFFFFFFIIIIIIIIIIIIIIIIIIIIIIIFIIIIIIIIIIIIII
...
```

Adaptor trimming:

Inline barcode description:

Each inline barcode is ligated to the 5' end of Read1 and its id and sequence are listed below:

```
A01      ATTGCTTAGATCGGAAGAGCGTCGTGT
B06      ACAAGCCAGATCGGAAGAGCGTCGTGT
C01      AACTTGTAGATCGGAAGAGCGTCGTGT
D08      AGGACCAAGATCGGAAGAGCGTCGTGT
A03      ANNNNGGTCATAGATCGGAAGAGCGTCGTGT
G07      ANNNNACAGGAAGATCGGAAGAGCGTCGTGT
A04      ANNNNAAGCTGAGATCGGAAGAGCGTCGTGT
F05      ANNNNGTATCCAGATCGGAAGAGCGTCGTGT
RiL19/none    AGATCGGAAGAGCGTCGTGT
```

Cutadapt round 1: Takes output from demultiplexed files. Run to trim off both 5' and 3' adapters on both reads

```
cutadapt -f fastq --match-read-wildcards  --times 1  -e 0.1  -O 1  --
quality-cutoff 6  -m 18  -a NNNNNAGATCGGAAGAGCACACGTCTGAACTCCAGTCAC  -g
CTTCCGATCTACAAGTT -g CTTCCGATCTTGGTCCT  -A AACTTGTAGATCGGA -A
AGGACCAAGATCGGA -A ACTTGTAGATCGGAA -A GGACCAAGATCGGAA -A CTTGT
AGATCGGAAG -A GACCAAGATCGGAAG -A TTGTAGATCGGAAGA -A ACCAAGATCGGAAGA -A
TGTAGATCGGAAGAG -A CCAAGATCGGAAGAG -A GTAGATCGGAAGAGC -A CAAGATCGGAAGAGC
-A TAGATCGGAAGAGCG -A AAGATCGGAAGAGCG -A AGATCGGAAGAGCGT -A
GATCGGAAGAGCGTC -A ATCGGAAGAGCGTCG -A TCGGAAGAGCGTCGT -A CGGAAGAGCGTCGTG
-A GGAAGAGCGTCGTGT  -o
/full/path/to/files/file_R1.C01.fastq.gz.adapterTrim.fastq.gz  -p
/full/path/to/files/file_R2.C01.fastq.gz.adapterTrim.fastq.gz
/full/path/to/files/file_R1.C01.fastq.gz
/full/path/to/files/file_R2.C01.fastq.gz >
/full/path/to/files/file_R1.C01.fastq.gz.adapterTrim.metrics
```

Cutadapt round 2: Takes output from cutadapt round 1. Run to trim off the 3' adapters on read 2, to control for double ligation events.

```
cutadapt -f fastq --match-read-wildcards  --times 1  -e 0.1  -O 5  --
quality-cutoff 6  -m 18  -A AACTTGTAGATCGGA -A AGGACCAAGATCGGA -A
ACTTGTAGATCGGAA -A GGACCAAGATCGGAA -A CTTGTAGATCGGAAG -A GACCAAGATCGGAAG
-A TTGTAGATCGGAAGA -A ACCAAGATCGGAAGA -A TGTAGATCGGAAGAG -A
CCAAGATCGGAAGAG -A GTAGATCGGAAGAGC -A CAAGATCGGAAGAGC -A TAGATCGGAAGAGCG
-A AAGATCGGAAGAGCG -A AGATCGGAAGAGCGT -A GATCGGAAGAGCGTC -A
ATCGGAAGAGCGTCG -A TCGGAAGAGCGTCGT -A CGGAAGAGCGTCGTG -A GGAAGAGCGTCGTGT
-o /full/path/to/files/file_R1.C01.fastq.gz.adapterTrim.round2.fastq.gz
-p /full/path/to/files/file_R2.C01.fastq.gz.adapterTrim.round2.fastq.gz
/full/path/to/files/file_R1.C01.fastq.gz.adapterTrim.fastq.gz
/full/path/to/files/file_R2.C01.fastq.gz.adapterTrim.fastq.gz >
/full/path/to/files/file_R1.C01.fastq.gz.adapterTrim.round2.metrics
```

Adaptor trimming:

- Key consideration – we've observed that adaptor-concatamer fragments (even at extremely low frequency) yield high-scoring eCLIP peaks
- Difficult to trim all with one pass
  - Cutadapt (by default) will miss adaptors with 5' truncations
- To avoid this, we err on the side of over-trimming

Repetitive element removal

- Majority of RNA in most cells are rRNA / tRNA / repeats
- These can map and cause strange artifacts (particularly rRNA, as a 40nt rRNA read with 1 or 2 sequencing errors can map uniquely to one of the various rRNA pseudogenes in the genome)
- To avoid false positives, we FIRST map all reads against a RepBase database, and only take reads that remain unmapped for further processing

STAR rmRep: Takes output from cutadapt round 2. Maps to human specific version of RepBase used to remove repetitive elements, helps control for spurious artifacts from rRNA (& other) repetitive reads.

```
STAR  --runMode alignReads  --runThreadN 16  --genomeDir
/path/to/RepBase_human_database_file --genomeLoad LoadAndRemove  --
readFilesIn
/full/path/to/files/file_R1.C01.fastq.gz.adapterTrim.round2.fastq.gz
/full/path/to/files/file_R2.C01.fastq.gz.adapterTrim.round2.fastq.gz --
outSAMunmapped Within  --outFilterMultimapNmax 30  --
outFilterMultimapScoreRange 1  --outFileNamePrefix
/full/path/to/files/file_R1.C01.fastq.gz.adapterTrim.round2.rep.bam --
outSAMattributes All  --readFilesCommand zcat  --outStd BAM_Unsorted  --
outSAMtype BAM Unsorted  --outFilterType BySJout  --outReadsUnmapped
Fastx  --outFilterScoreMin 10  --outSAMattrRGline ID:foo  --alignEndsType
EndToEnd >
/full/path/to/files/file_R1.C01.fastq.gz.adapterTrim.round2.rep.bam
```

Mapping to human genome

- We perform paired-end mapping with STAR to the human genome plus splice junction database, keeping only uniquely mapped reads

STAR genome mapping: Takes output from STAR rmRep. Maps unique reads to the human genome

```
STAR  --runMode alignReads  --runThreadN 16  --genomeDir
/path/to/STAR_database_file --genomeLoad LoadAndRemove  --readFilesIn
/full/path/to/files/file_R1.C01.fastq.gz.adapterTrim.round2.rep.bamUnmapp
ed.out.mate1
/full/path/to/files/file_R1.C01.fastq.gz.adapterTrim.round2.rep.bamUnmapp
ed.out.mate2  --outSAMunmapped Within  --outFilterMultimapNmax 1  --
outFilterMultimapScoreRange 1  --outFileNamePrefix
/full/path/to/files/file_R1.C01.fastq.gz.adapterTrim.round2.rmRep.bam  --
outSAMattributes All  --outStd BAM_Unsorted  --outSAMtype BAM Unsorted  -
-outFilterType BySJout  --outReadsUnmapped Fastx  --outFilterScoreMin 10
--outSAMattrRGline ID:foo  --alignEndsType EndToEnd >
/full/path/to/files/file_R1.C01.fastq.gz.adapterTrim.round2.rmRep.bam
```

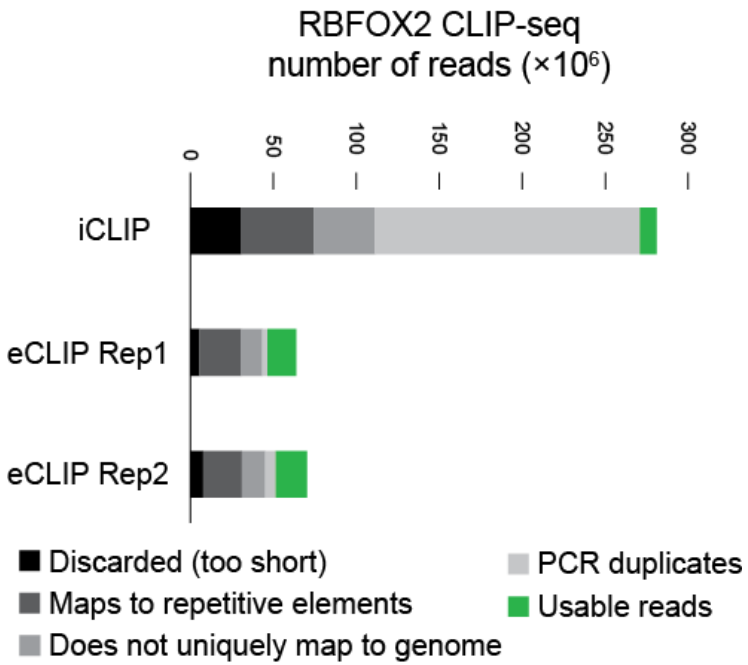
PCR duplicate removal

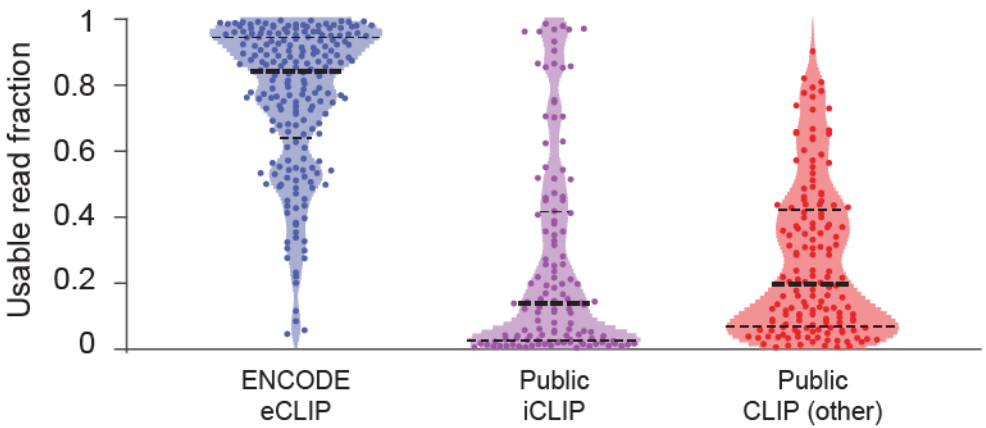
- Next, we compare reads that map to the same location (based on the mapped start of R1 and start of R2) based on their random-mer sequence
  - If two reads map to the same position and have the same random-mer, one is discarded
- Input: bam file containing only uniquely mapped reads
- Output: bam file containing only "Usable" (uniquely mapped, non-PCR duplicate) reads

Barcode_collapse_pe: takes output from STAR genome mapping. Custom random-mer-aware script for PCR duplicate removal.

```
barcode_collapse_pe.py  --bam
/full/path/to/files/file_R1.C01.fastq.gz.adapterTrim.round2.rmRep.bam  --
out_file
/full/path/to/files/file_R1.C01.fastq.gz.adapterTrim.round2.rmRep.rmDup.b
am  --metrics_file
/full/path/to/files/file_R1.C01.fastq.gz.adapterTrim.round2.rmRep.rmDup.m
etrics
```

eCLIP significantly decreases PCR duplication rate





File details: bam files

CCTTG = random-mer (first 5 or 10nt of sequenced read2) – has been removed from the 5' end of read2 and appended to read name

```
CCTTG:SN1001:449:HGTN3ADXX:1:1206:8464:69989          147        chr1       14771      255        43M
          =          14681      -133       CACGCGGGCAAAGGCTCCTCCGGGCCCCTCACCAGCCCCAGGT
          B<FFFFFB<0<<<IIFBF<07FFFBFIFFFFFBB<B<BBFFFB NH:i:1     HI:i:1     AS:i:80    nM:i:0     NM:i:0
          MD:Z:43    jM:B:c,-1  jI:B:i,-1  RG:Z:foo

CCCCT:SN1001:449:HGTN3ADXX:2:2101:6568:79173          147        chr1       15206      255        44M
          =          15204      -46        GCGGCGGTTTGAGGAGCCACCTCCCAGCCACCTCGGGGCCAGGG
          FFFFIIIIIIIIIIIIIFFIIIIIIIIFFIIIIIIFFFFFFF          NH:i:1     HI:i:1     AS:i:76    nM:i:2
          NM:i:1     MD:Z:5T38  jM:B:c,-1  jI:B:i,-1  RG:Z:foo
```

Peak calling

Step 1) Initial cluster identification with CLIPper

(spline-fitting with transcript-level background normalization)

Clipper: Takes results from samtools view. Calls peaks on those files.

```
clipper  -b /full/path/to/files/CombinedID.merged.r2.bam  -s hg19  -o
/full/path/to/files/CombinedID.merged.r2.peaks.bed  --bonferroni  --
superlocal  --threshold-method binomial  --save-pickle
```

Step 2) Compare clusters against size-matched input

```
perl overlap_peakfi_with_bam_PE.pl
/full/path/to/desired_output_directory/CombinedID_rep1.merged.r2.bam
/full/path/to/desired_output_directory/CombinedID_INPUT.merged.r2.bam
/full/path/to/desired_output_directory/CombinedID_rep1.merged.r2.peaks.bed
/full/path/to/manfest_file.txt.mapped_read_num
/full/path/to/desired_output_directory/uID_Rep.basedon_uID_Rep.peaks.l2inputnor
mnew.bed
```

Output file has bed format:

```
Chr \t start \t stop \t log10(p-value eCLIP vs SMInput) \t log2(fold-enrichment
in eCLIP vs SMInput) \t strand
```

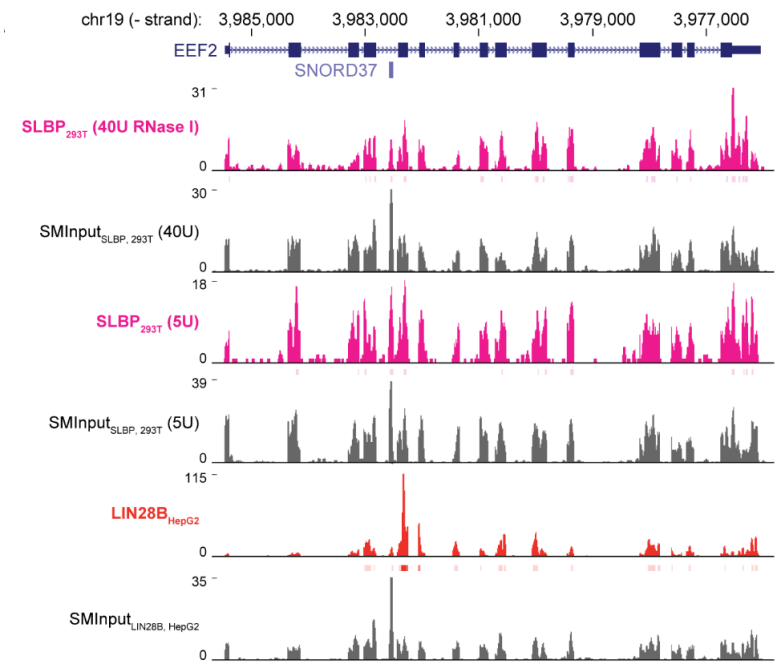
Step 3) Compress clusters (as CLIPper is transcript-level, it can occasionally call overlapping peaks – this step iteratively removes overlapping peaks by keeping the one with greater enrichment above input)

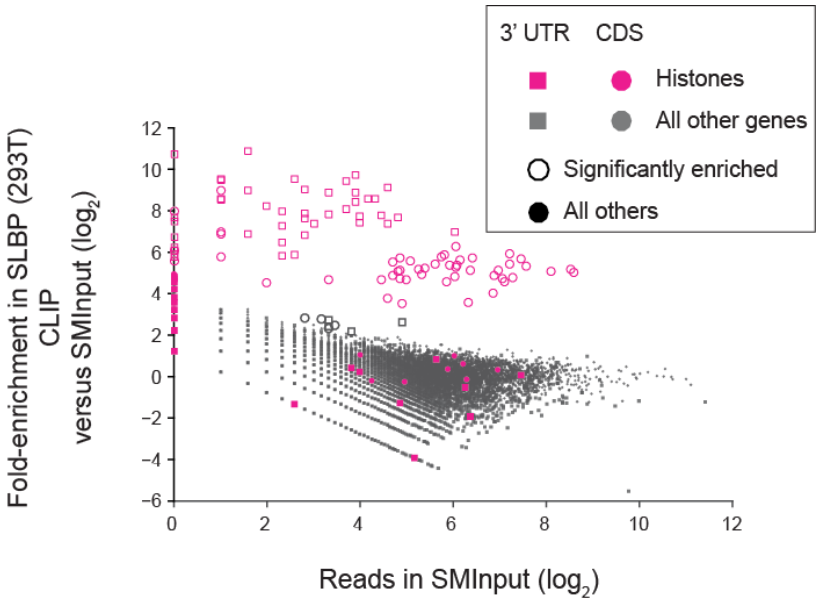
```
perl compress_l2foldenrpeakfi.pl
/full/path/to/desired_output_directory/uID_Rep.basedon_uID_Rep.peaks.l2inputnor
mnew.bed
```

Writes output to bed format file (same columns as above):

```
/full/path/to/desired_output_directory/uID_Rep.basedon_uID_Rep.peaks.l2inputnor
mnew.bed.compressed.bed
```

Why input normalize?

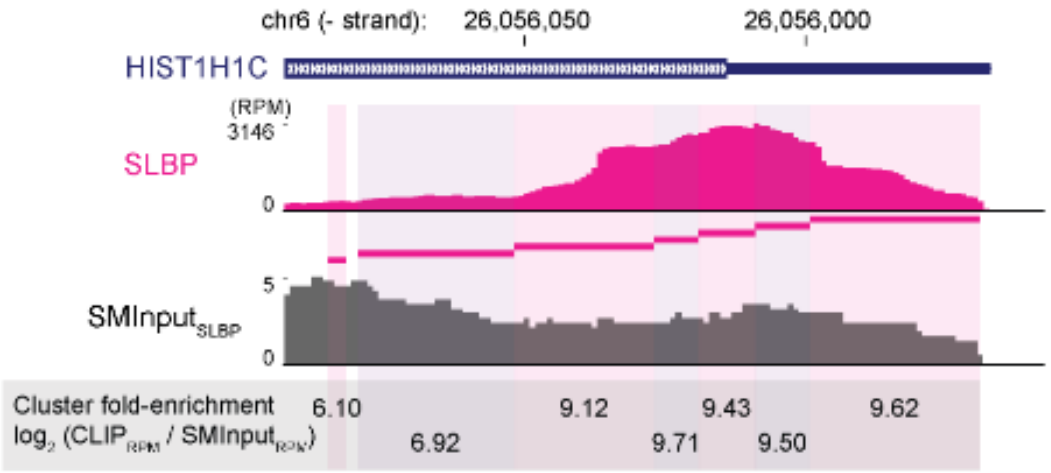


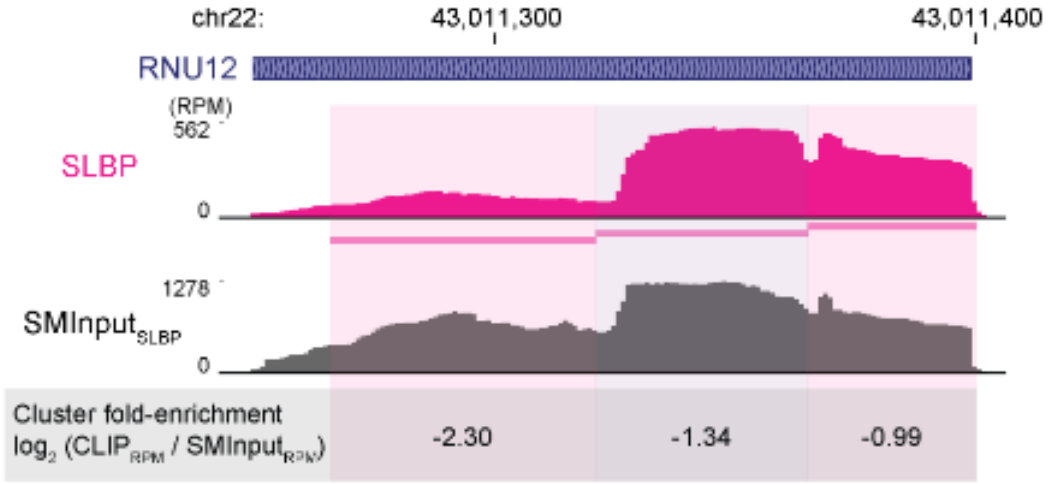


- We see mRNA background at nearly all abundant genes…

… but true signal is highly enriched above this background

Input normalization removes false-positives and identifies confident binding sites





File details: bed narrowPeak (input-normalized peaks)

```
chr \t    start \t  stop \t   dataset_label \t    1000 \t   strand \t log2(eCLIP fold-enrichment
over size-matched input) \t   -log10(eCLIP vs size-matched input p-value) \t    -1 \t     -1
```

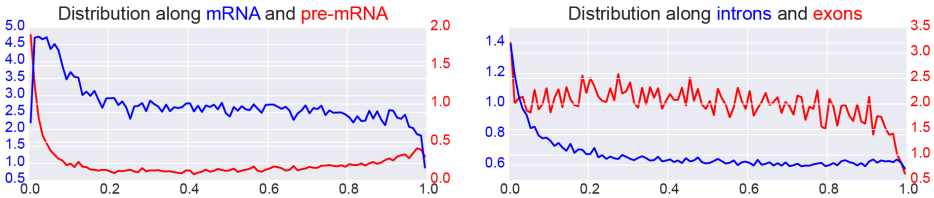
- Note: p-value is calculated by Fisher's Exact test (minimum p-value 2.2×10^{-16}), with chi-square test (–log10(p-value) set to 400 if p-value reported == 0)
- Our typical 'stringent' cutoffs: require -log10(p-value) ≥ 5 and log2(fold-enrichment) ≥ 3

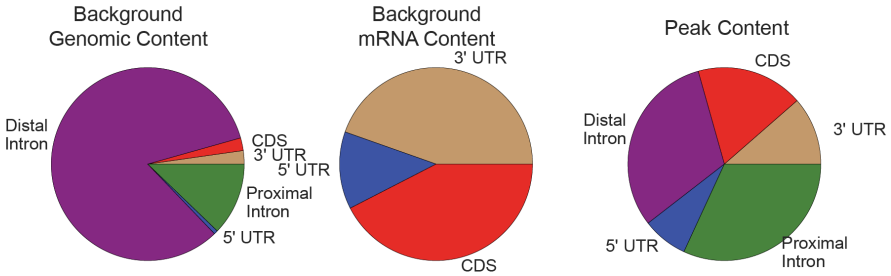
```
track type=narrowPeak visibility=3 db=hg19 name="RBFOX2_HepG2_rep01" description="RBFOX2_HepG2_rep01
input-normalized peaks"

Chr7   4757099   4757219   RBFOX2_HepG2_rep01   1000  +   6.539331235     400        -1        -1

Chr7   99949578  99949652  RBFOX2_HepG2_rep01   1000  +   5.233511963     400        -1        -1

Chr7   1027402   1027481   RBFOX2_HepG2_rep01   1000  +   5.243129966     69.5293984 -1        -1
```

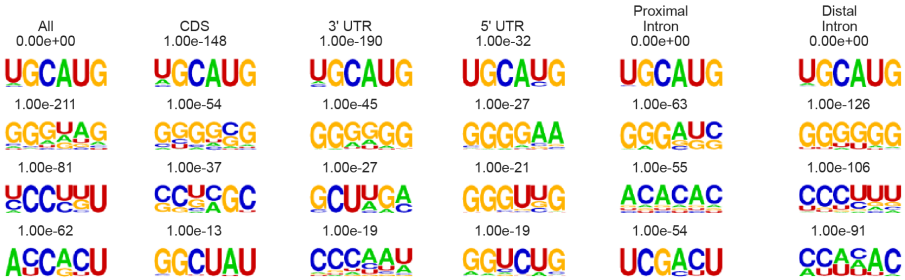
What can we do with the eCLIP database?

Individual RBP analyses

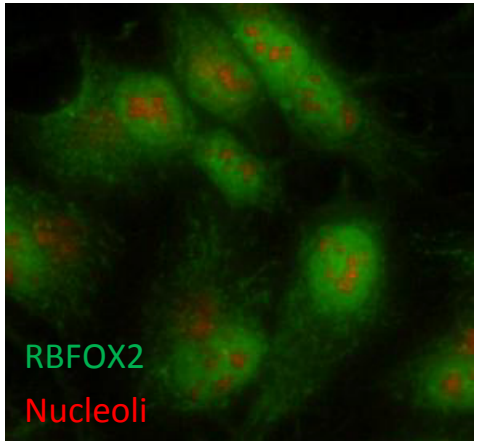
eCLIP analysis



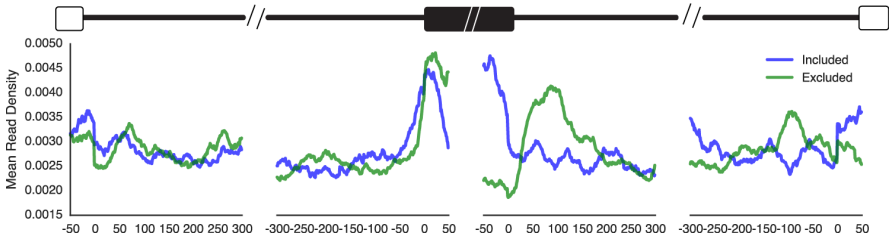




RBP localization

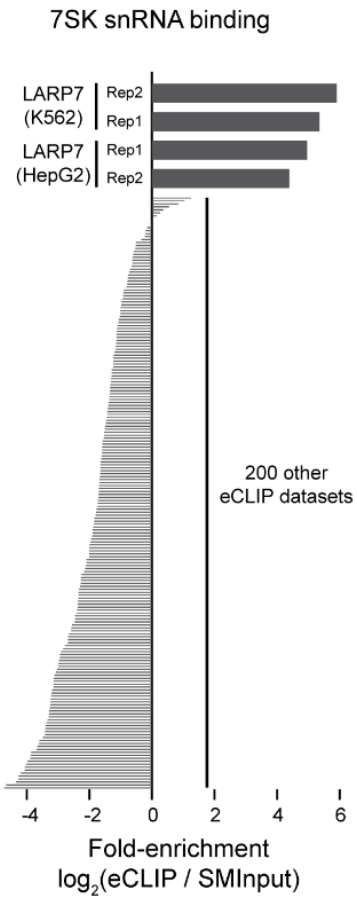


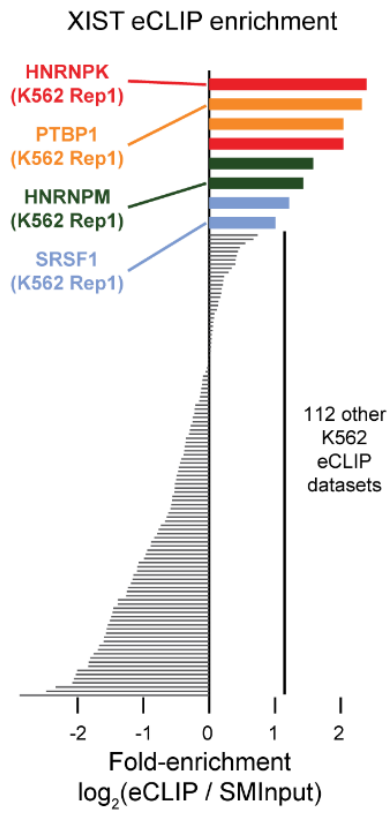
Integration with knockdown RNA-seq

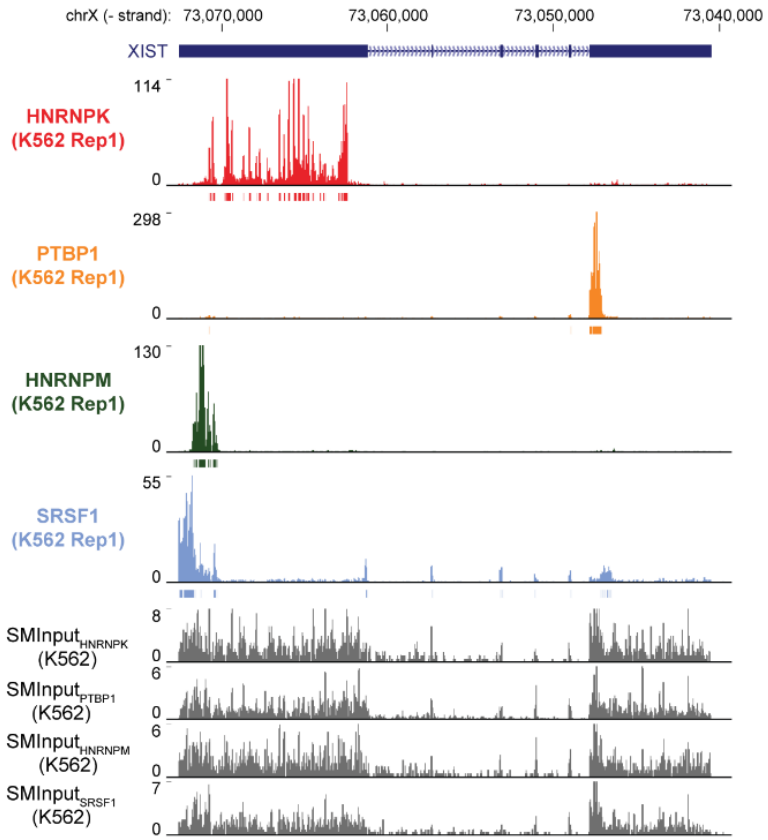


An “RNA-centric” view of RBP-binding

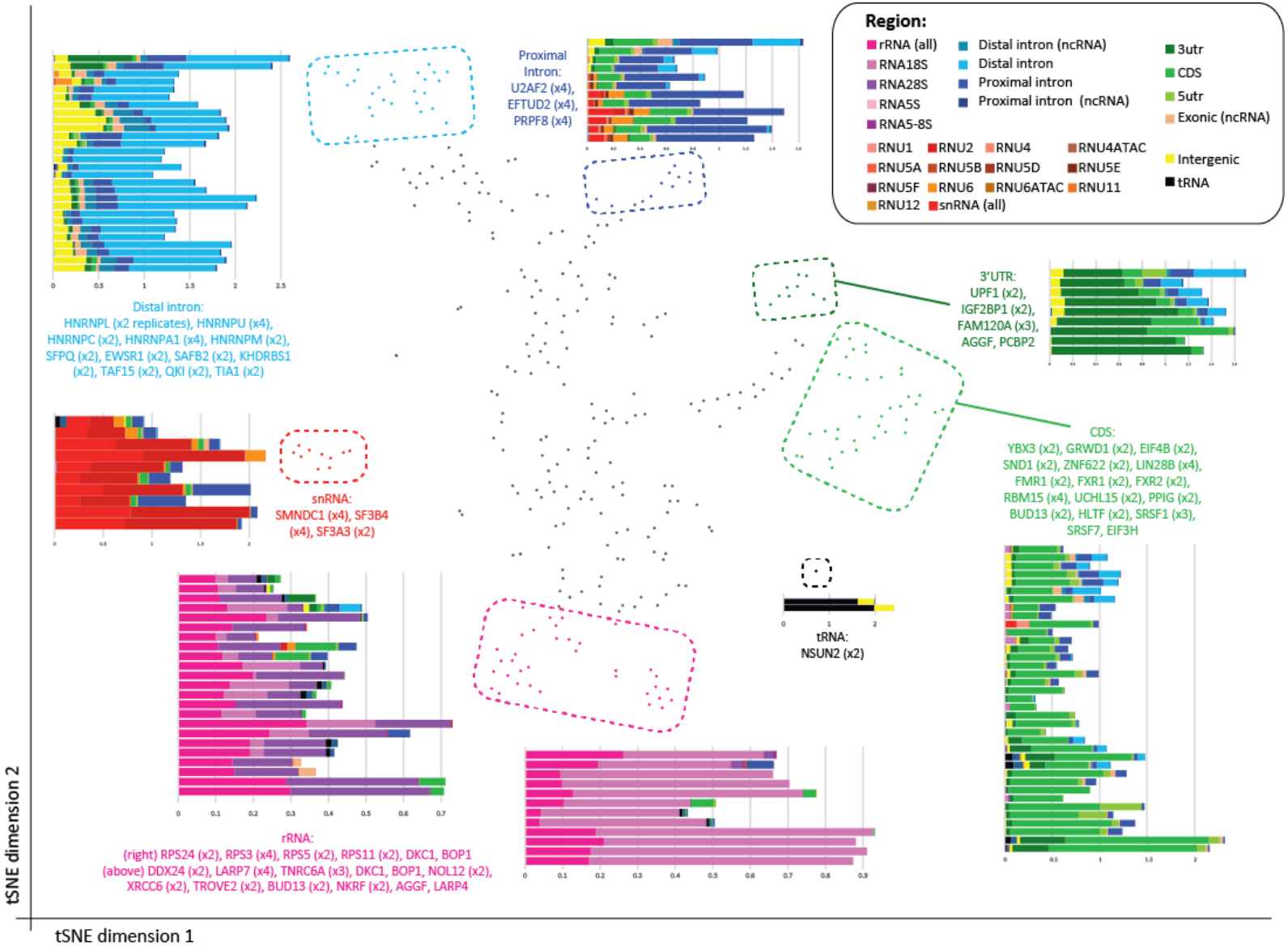
‘*in silico* screen’ of a desired RNA against all CLIP datasets to identify the best-binding RBPs







Integrated global views of RBP binding



Tools available soon (next few months):

- eCLIP processing pipeline on DNA Nexus (should be ready ~July)
  - Followed quickly by IDR & q/c metrics for validating your own eCLIP datasets
- RNA-centric browser (website at alpha stage now)
  - Allow users to query RNAs or genomic regions of interest against our ENCODE eCLIP database
- Integration with ENCODE encyclopedia
- Factorbook-like summaries for each RBP

Acknowledgements

UC San Diego Gene Yeo

ENCODE

ENCODE

Brent Graveley
Chris Burge
Eric Lécuyer
Xiang-Dong Fu

Computational:
Gabriel Pratt
Eric Van Nostrand
Shashank Sathe
Brian Yee

Experimental:
Eric Van Nostrand
Steven Blue
Thai Nguyen
Chelsea Gelboin-Burkhart
Ruth Wang
Ines Rabano

Alumni:
Balaji Sundararaman
Keri Elkins
Rebecca Stanton

National Institutes of Health

Damon Runyon Cancer Research Foundation

NSF

ARCS Advancing Science in America

Funding: NIH National Human Genome Research Institute
Advancing human health through genomics research

UCSD Genetics Training Program