**April 1, 2025**

1. Public Project – new data upload

2. Reupload old data – methods / accessions codes (Array expression detailed sample information and links for raw fastq files)

* [**Characterizing the RNA targets and position-dependent splicing regulation by TDP-43 - PubMed**](https://pubmed.ncbi.nlm.nih.gov/21358640/)
* [**TDP-43 condensation properties specify its RNA-binding and regulatory repertoire - PubMed**](https://pubmed.ncbi.nlm.nih.gov/34380047/)
* [**Heteromeric RNP Assembly at LINEs Controls Lineage-Specific RNA Processing - PubMed**](https://pubmed.ncbi.nlm.nih.gov/30078707/)

3. Flow development and feedback (e.g. Sample presentation), documentation

4. Selected annotation files

5. CLIP variant files may need adjustment

6. Intronic conservation across cell types and species, unannotated splice junctions (splice vault)

7. Cryptic exon repression via microsatellites, transposons, repetitive nucleotides recruit repressors but evolve rapidly e.g. DNA polymerase slippage, TDP-43 primary exonic repressors, predict splice inclusion rating

Per Charlotte: cell or tissue should be the cell or tissue e.g. brain or ESC or SH-SY5Y

* purification agent should be the tdp43 antibody used in the paper
* jernej is the PI, scientist is who did the experiment, in this case i guess james tollervy - if these people have flow you can put their username i think (might be in docs?)
* organisation would be MRC LMB Cambridge i think
* sample name should be something meaningful - this name will be propagated throughout analysis to all downstream files - you might consider concatenating protein-name\_species\_cellOrTissue\_condition\_replicateNumber - something like that
* the 3'barcode is correct - this is a quirk of the old data, it means that when you run the clipseq pipeline you will need to set it to trim two nucleotides at the 3' end of reads once illumina adapters have been removed - will have to look carefully at trimgalore/cutadapt docs to see if its possible to do that in one step

**April 2, 2025**

1. Used ENA script /home/mikej10/advbfx/tdpcondens8/ena-file-download-read\_run-PRJEB51554-fastq\_ftp-20250402-1126.sh to download all TDPcondensate files, convert table to Flow CLIP template. Concatenated file names using ‘awk -F'\t' '{print $1 ".fastq.gz"}' \*.tsv > Filenames.txt’

2. Run TDP43 Hs with extract UMI to Header True, UMI Header format NNNNN, and UMI Collapse separator \_

trimgalore --fastqc --length 10 -q 20 --three\_prime\_clip\_R1 2

<https://github.com/FelixKrueger/TrimGalore/blob/master/Docs/Trim_Galore_User_Guide.md#step-3-removing-short-sequences>

Errors with replicate must start with 1, accidentally uploaded twice, cannot removeA screenshot of a computer

AI-generated content may be incorrect.

**April 3, 2025**

1. Finish CLIP template for TDP condensate project

* Assume sample 1 and 2 from 2020 are in order, eg. 3 and 4
* TARDBP:Q331K-eGFP, no UV – assume control data
* Assume HD on bead is replicate 3, confirmed from ENA <https://www.ebi.ac.uk/ena/browser/view/ERR9192699>
* Unable to match low/med RNAse, fixed
* Hnrpa1/2 from 23/10/2015
* EndoTDP-43 from 29/11/2018
* Missing WT\_med RNase\_1 from 10/10/2017, headers row

3. Bioinformatics assignment, finish paper

4. Remove duplicates, done, rerun pipelines x2

**April 4, 2025**

1. Reupload all files with dated sample names

2. Uploaded files to DRI dropbox

3. Created LINEs repo and downloaded fastq files for MATR3 and PTBP1, mouse and RNAs

**April 7, 2025**

1. Create LINE RNA flow template

2. Run condensation api script, added genome, scraped links off of flow:

const links = Array.from(document.querySelectorAll('a[href]')).map(a => a.href);

copy(links.join('\n'));

console.log(`${links.length}`);

in bash - uniq Filenames.txt > filelinks.txt

awk '{gsub(/[^0-9]/,"")}1' filelinks.txt > samplelist.txt

tac samplelist.txt > revsamplelist.txt

paste revsample.txt <(cut -f2- fileinput.tsv) > updated.tsv

Condensation pipeline plan:

1. Run samples WT, 274del, 316del, 320del, 367del, 320del (CTD) (30 samples)
2. HNRNP (10 samples)
3. Hexandiol group (18 samples)
4. RNAse (8 samples)

**April 8, 2025**

1. Run LINE CLIP data in human and mouse batches with UMI header format NNNNNNNNN, separator \_, trim galore --fastqc --length 10 -q 20 - DONE
2. Upload all LINE RNA data, adjust metadata as needed – DONE and PUBLIC
3. Run Pipeline for condensation data RNAse and HNRPA batches, extract UMI to header FALSE, trim galore --fastqc --length 10 -q 20 – FAILED
4. Ran first 12 RNA-Seq samples with GRCh38

HNRPA UMI collapse fail:

Arguments [bam, -i, GFP-HNRPA2B1-TDP43\_TDPCTD\_1\_20200812\_R1.bam, -o, GFP-HNRPA2B1-TDP43\_TDPCTD\_1\_20200812\_R1.smrna\_withk1.dedup.bam, --umi-sep, rbc:]

Exception in thread "main" java.lang.IllegalStateException: No match found

at java.base/java.util.regex.Matcher.group(Matcher.java:644)

at umicollapse.util.SAMRead.getUMI(SAMRead.java:36)

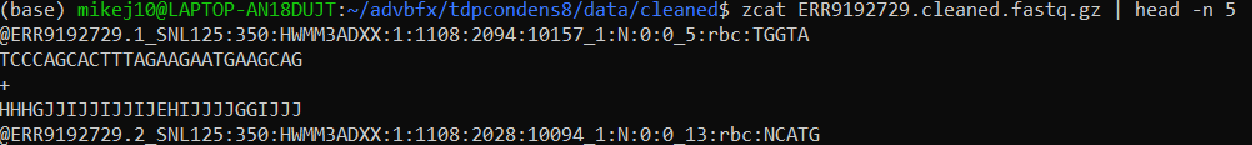
at umicollapse.main.DeduplicateSAM.deduplicateAndMerge(DeduplicateSAM.java:109)

at umicollapse.main.Main.main(Main.java:221)

1. Set up removespace.py script to run with run\_removespace.sh for all 66 condensation files

A black screen with white text

AI-generated content may be incorrect.



1. Need to download ~250 files from irCLIP data set from SRA Run Selector with sra-tools – DONE

prefetch -o ./sra --option-file accessionp1.txt

cat sraacclist.txt | xargs -n 1 -P 4 fasterq-dump --outdir ./fastq

Files in individual folders – move to parent directory and remove empty files,

find . -mindepth 2 -type f -name "\*.sra" -exec mv {} . \;

find . -type d -empty -delete

**April 9, 2025**

1. Finished uploading condensation files. API example script received. Removed duplicate files
   1. Created new uniq id list for condensation files
   2. Ran pipeline with 66 inputs
2. Plan to run 24 remaining LINE RNA-seq files in pipeline - DONE
3. Proposal – deep intronic splicing in brains.
4. irCLIP SRR files converting to fastq – doing 260 - 379
5. irCLIP flow template needs purification agent and barcodes - DONE

**April 10, 2025**

1. Change my fork branch for pull request, bedgraph split into positive negative strands and make two separate bigwigs for each
2. Input source based parameter validation – two possible sample sheets with rnasplice, check usage fastq’s or aligned bams
   1. Source configuration
   2. Dash’s code, input called fasta for genome rather than sample
   3. Line 276 on clipseq.nf, input check line 102
   4. Input\_check.py, samplesheet\_check.py, samplesheet\_bam\_check.py line 52 shows what samplesheet should look like
3. Merge all the Ctrl and all FTLD together, and another showing two best samples of each (i.e., excluding Ctrl\_25 and FTLD\_20).
   1. Generate one image that zooms into the cluster in intron5, and another that zooms further out (let’s say, showing 3-4 nearest exons, but not the whole gene), merge them you do something like the below, so concatenate, then sort, then use bedtools groupby to add up crosslinks at the same positions:
      1. cat \*bed | sort -k1,1 -k2,2n -k6,6 | \

bedtools groupby -i stdin -g 1,2,3,6 -c 5 -o sum > "$OUTPUT"

* 1. fix bed files for file in \*.bed; do

awk 'BEGIN{OFS="\t"} {tmp=$5; $5=$6; $6=tmp; print}' "$file" > "${file%.bed}\_fixed.bed"

done

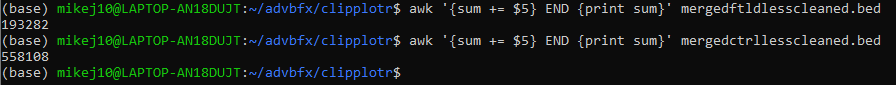
* 1. ./clipplotr --xlinks 'merged\_grouped\_fixed.bed,2merged\_grouped\_fixed.bed,ctrlmerged\_grouped\_fixed.bed,ctrl2merged\_grouped\_fixed.bed' -g gencode.v38.annotation.gtf --region 'chr7:154040000:154900000:+' --output plot.pdf

1. Download SRR files again, prefetch --option-file SRR3079.txt for files 330 – 379

**April 14, 2025**

1. Troubleshoot **condensation pipeline** run.
2. Troubleshoot **clipplotr run**. Confirm bed files format is appropriate and successfully run test.
3. Read through **proposal** and brainstorm on ideal path forward, finalize by 16/4 for Jernej review.
4. Secondary tasks – fix **bigwig** code, upload **irCLIP** files, review flow **LINE study data**

**April 15, 2025**



cDNA totals for FTLD and CTRL samples in medium temporal gyrus brain data

**April 18, 2025**

1. Finish a rough draft of research proposal and continue reading around methodologies
2. Flowbio API upload for irCLIP data, queries.py ‘type’ removed from SAMPLE block

**April 22, 2025**

1. Research Project Proposal – meeting w/ Jernej Wednesday 2PM
2. Finish irCLIP upload and begin FLASH downloads?
3. Troubleshoot **condensation pipeline** run:

Traceback (most recent call last):

File "/media/storage/production/executions/152292164832806495/work/09/5b05de66e2c9af43457933c4858885/.command.sh", line 297, in <module>

main(args.process\_name)

File "/media/storage/production/executions/152292164832806495/work/09/5b05de66e2c9af43457933c4858885/.command.sh", line 264, in main

peaks\_metrics = [get\_peaks\_metrics(pc) for pc in peakcallers]

File "/media/storage/production/executions/152292164832806495/work/09/5b05de66e2c9af43457933c4858885/.command.sh", line 264, in <listcomp>

peaks\_metrics = [get\_peaks\_metrics(pc) for pc in peakcallers]

File "/media/storage/production/executions/152292164832806495/work/09/5b05de66e2c9af43457933c4858885/.command.sh", line 233, in get\_peaks\_metrics

expanded\_xlinks\_bed = pbt.BedTool.from\_dataframe(expanded\_xlinks\_df)

File "/usr/local/lib/python3.10/site-packages/pybedtools/bedtool.py", line 633, in from\_dataframe

df.to\_csv(outfile, \*\*default\_kwargs)

File "/usr/local/lib/python3.10/site-packages/pandas/util/\_decorators.py", line 211, in wrapper

return func(\*args, \*\*kwargs)

File "/usr/local/lib/python3.10/site-packages/pandas/core/generic.py", line 3721, in to\_csv

return DataFrameRenderer(formatter).to\_csv(

File "/usr/local/lib/python3.10/site-packages/pandas/util/\_decorators.py", line 211, in wrapper

return func(\*args, \*\*kwargs)

File "/usr/local/lib/python3.10/site-packages/pandas/io/formats/format.py", line 1189, in to\_csv

csv\_formatter.save()

File "/usr/local/lib/python3.10/site-packages/pandas/io/formats/csvs.py", line 261, in save

self.\_save()

File "/usr/local/lib/python3.10/site-packages/pandas/io/formats/csvs.py", line 266, in \_save

self.\_save\_body()

File "/usr/local/lib/python3.10/site-packages/pandas/io/formats/csvs.py", line 304, in \_save\_body

self.\_save\_chunk(start\_i, end\_i)

File "/usr/local/lib/python3.10/site-packages/pandas/io/formats/csvs.py", line 315, in \_save\_chunk

libwriters.write\_csv\_rows(

File "pandas/\_libs/writers.pyx", line 72, in pandas.\_libs.writers.write\_csv\_rows

OSError: [Errno 28] No space left on device

**April 23, 2025**

1. Revised RPP for Jernej meeting at 2PM
2. Analyzing VastDB data for intron retention statistics
3. Downloaded flash data set 1

head -n 1 PSI\_TABLE-hg38.tab | tr '\t' '\n'

head -n 1 EVENT\_INFO-hg38.tab | tr '\t' '\n' # check column names

cut -f2 EVENT\_INFO-hg38.tab | tail -n +2 | cut -c1-5 | sort | uniq # confirm HsaIN exists

awk 'NR==1 || $2 ~ /^HsaIN/' EVENT\_INFO-hg38.tab > IN\_EVENT-hg38.tab # pipe HsaIN only to new file

awk 'NR>1 {print $2}' IN\_EVENT-hg38.tab > intron\_ids.txt # Match PSI values to EVENT INFO

grep -Ff intron\_ids.txt PSI\_TABLE-hg38.tab > PSI\_IN\_ONLY.tmp

head -n 1 PSI\_TABLE-hg38.tab > header.tmp

cat header.tmp PSI\_IN\_ONLY.tmp > PSI\_IN\_EVENT-hg38.tab

rm intron\_ids.txt PSI\_IN\_ONLY.tmp header.tmp

Column Description

GENE Gene symbol associated with the splicing event

EVENT Unique ID for the event, e.g. HsaIN001234 for intron retention

COORD\_o Original genomic coordinates (hg38)

LE\_o Length of exon/intron event (original)

FULL\_CO Full coordinates of the event

COMPLEX Indicates if the event involves more than one type of splice change

REF\_CO Reference (canonical) transcript coordinates

LE\_n Updated/normalized length (post-ref alignment)

CO\_C1 Coordinates for condition 1 splicing isoform

CO\_A Coordinates for alternative isoform

CO\_C2 Coordinates for condition 2 splicing isoform

Seq\_C1 Sequence corresponding to CO\_C1

Seq\_A Sequence corresponding to CO\_A

Seq\_C2 Sequence corresponding to CO\_C2

**April 24, 2025**

1. Flow template for FLASH data
2. Reanalyze VastDB data starting from expression
3. **Condensation pipeline run**
4. Fix irCLIP data to make public

**April 25, 2025**

Same as yesterday. RPP submitted. Starting git repo, made directories and an rMarkdown file for VastDB exploration. Working to transpose [GSE118265\_series\_matrix.txt.gz](https://ftp.ncbi.nlm.nih.gov/geo/series/GSE118nnn/GSE118265/matrix/GSE118265_series_matrix.txt.gz)

sed -E ':a; s/"([^"]\*) ([^"]\*)"/"\1\_\2"/g; ta' GSE118265\_series.txt > GSE118265\_cleaned.txt # replace spaces in sample title with ‘\_’

awk ' # transpose all metadata

{

for (i=1; i<=NF; i++) {

a[NR,i] = $i

}

}

NF>p { p = NF }

END {

for(j=1; j<=p; j++) {

str = a[1,j]

for(i=2; i<=NR; i++){

str = str "\t" a[i,j]

}

print str

}

}' GSE118265\_cleaned.txt > GSE118265\_transposed.tsv

**April 28, 2025**

1. VastDB analysis
2. Upload FLASH data – remove 3’ barcode with umitools
3. **Reanalyze condensation data – troubleshoot why the input file is empty**
4. Begin irCLIP + Re-CLIP analysis
5. Fix bigwig to bedgraph code

**April 29, 2025**

1. ~~ FLASH
   1. Run a `srun` command with –pty to use pseudo terminal mode then run scripts from the command line on the allocated compute node, then run a `sbatch` command
      1. srun -p cpu --time 60 --pty /bin/bash -l
      2. sbatch -p cpu fasterq-dump.sh
   2. Add `-l` to the shebang on a command allocates a compute node
   3. Allocate cpus and time limit, set your partition

**April 30, 2025**

1. Fixed Rstudio and conda in CREATE
2. Set up VastDB project and worked through some expression data, looking at grouping samples and plotted first raw histogram and a PCA of the PSI data for events w PSI Avg over 20

**May 1, 2025**

1. Started working with PSI and EVENT METRICS dataset to check All PSI values, Max PSI values, tried grouping by sample, and binarized with values of 10, 15, 20 or 25 avg PSI.

**May 2, 2025**

1. Cleaned up Rmd file, began by filtering data for Max < 1. For validation that length doesn’t affect retention bias in the data, plotted hexbin and percentile box plot of intron length versus Max PSI. Finally grouped introns as never-retained, and stable before clustering with k-means = 6 and creating a heatmap.

**May 6, 2025**

1. Downloaded and zipped all FLASH files in CREATE. Working on UMI-extract tools script.

Goals for the week ~

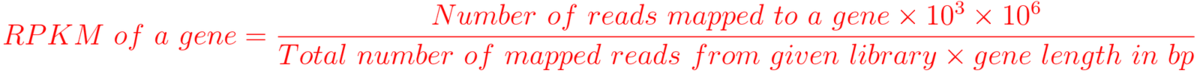
* + 1. Checked reads are all in order with repair.sh, extracted headers with umi-tools extract, copied headers to read \_1, last step is to merge reads with NGmerge, separate all unmatched reads and match
    2. Run heatmap again with colcluster = TRUE
    3. PDF of Rmd file, scripts folder, new markdown for TPM and IR maybe use a quarto notebook.

**May 7, 2025**

* + - 1. Find program to check all unmapped reads, and use NGmerge
      2. Removed stable and never retained groups and scaling for matrix.
      3. look at the cell/tissue clustering, summarise what kinds of cells/tissues are associated with each cluster

**May, 13, 2025**1. Finished first Rmd file, still need to add tissue group labels and determine the identity of the sparse columns

1. Reanalyze condensation data to reproduce error. Redownload and check header, NNNNNNN
   1. Looks good
2. Download RBPNet
3. Upload SM data
4. Upload FLASH data
5. Get cRPKM PSI correlation for each cluster



**May 14, 2025**

1. Rmd File Pt 1 uploaded with fixed colors
2. Uploading Spliceosome files now, done and analyzing
3. Reanalyzing condensation data with flow.py, fixed API script
4. Fixing colors on heatmap, done and uploaded
5. Continue cRPKM analysis, plots are functional, faceted and cluster overlays
   1. Calculate RPKM / PSI for each cluster
6. Work on RBPNet
7. Upload FLASH data manually

**May 15, 2025**

1. RBPNet day
   1. Gather intron sequences from metaPsi table
      1. One cluster or all four?
   2. Create a dataspec.yml file which defines tasks and assigns signal bigWig and input regions
   3. TFRecords training set generated
   4. Train with rbpnet train
2. Braunschweig characteristics
   1. %C/G (Intron center, end, 3’ junction and 5’ exon)
   2. Length
   3. 5’ splice site
   4. 3’ splice site
   5. Within UTR
   6. Length relative to 3’ exon
   7. Position 5’ 🡪 3
   8. Last intron highly retained
3. Plotting hexbins of cRPKM / PSI for each cluster
4. Executions failed
5. FLASH must be uploaded from CREATE

**May 17, 2025**

1. RBPnet environment created, tfrecord created, training sbatched
2. Fixing expression filtered heatmap

**May 18, 2025**

1. Train RBPnet w gpu
2. Condensation analysis runs
   1. --smrna\_fasta Homo\_sapiens.GRCh38.smrna.fasta
   2. --smrna\_fasta\_fai Homo\_sapiens.GRCh38.smrna.fasta.fai
   3. --smrna\_chrom\_sizes Homo\_sapiens.GRCh38.smrna.fasta.sizes
3. Spliceosome run started – needed to remove 2 SMB FLAG samples from 2012 due to barcode extracted but not in header
4. irCLIP data need to be processed in two runs, with barcode extraction

**May 21, 2025**

1. fixed TDP43 condensation data and reuploaded, ran analysis again DONE
2. Attempting rbpnet on local
   1. rbpnet tfrecord -o train.tfrecord dataspec.yml
   2. rbpnet train -d dataspec.yml -c config.no-control.gin -o ~/advbfx/rbpnet/ train.tfrecord
   3. rbpnet predict -m PRPF8\_HepG2.model.h5 -o HNRNPH1test.tsv --format fasta HNRNPH1test1.fa
   4. rbpnet predict -m models/PRPF8\_HepG2.model.h5 -o data/introns50seq.tsv --format fasta data/sequencescln.fa
3. Need to try irCLIP API with ?page2 – done, python scripts created

**May 22, 2025**

1. RBPnet first priority connect with Klara – issues due to ?
2. irCLIP running with pagination on 32 samples with barcode length 10 – *dreamy cajal*
   1. 214 remaining, batch as follows
      1. AGTCA 36 records – *goofy solvay*
      2. AAGTC/ACTGA 26 records – *hopeful goldberg*
      3. CAGTC 36 records – *irreverent torvalds*
      4. CCAGT/CTGAC 36 records -
      5. GACTG/GGTCA 32 records -
      6. GTCAG/TCAGT 20 records -
      7. TGACT/TTCAG 28 records -
3. Need to fix upload script for FLASH, download w cyberduck instead, fixed xlsx file

RBPnet

Create introns\_flank50.bed

awk '{

start = ($2 - 50 >= 0) ? $2 - 50 : 0; # avoid negative start

end = $3 + 50;

print $1 "\t" start "\t" end "\t" $4 "\t" $5 "\t" $6;

}' introns.bed > introns\_flank50.bed

Create sequences.fa

bedtools getfasta \

-fi /scratch/prj/ppn\_rnp\_networks/shared/references/genomes/homo\_sapiens/GRCh38-EnsembleRelease109/Homo\_sapiens.GRCh38.fasta \

-bed introns50ens.bed \

-s -name \

-fo sequences.fa

>HNRNPH1::chr5:179621347-179623086(-)

>HNRNPH1::chr5:179620985-179621291(-)

>HNRNPH1::chr5:179619357-179620941(-)

>HNRNPH1::chr5:179618273-179619318(-)

>HNRNPH1::chr5:179618010-179618194(-)

>HNRNPH1::chr5:179617882-179618038(-)

>HNRNPH1::chr5:179617599-179617848(-)

>HNRNPH1::chr5:179616908-179617100(-)

>HNRNPH1::chr5:179616168-179616918(-)

>HNRNPH1::chr5:179615545-179616175(-)

>HNRNPH1::chr5:179614909-179615595(-)

(base) mikej10@LAPTOP-AN18DUJT:~/advbfx/rbpnet$ grep HNRNPH1::chr5:179616908-179617100 -A 1 sequences.fa

>HNRNPH1::chr5:179616908-179617100(-)

TGTAGAACTCTTCTTGAATTCTACAGCAGGAGCAAGCGGTGGTGCTTACGGTAGCCAAATGCTAGGAGGCATGGGTTTGTGTAAATATCACTTTAGTGTCTTTTTTTTAAGCTAACCTTGTATGCCTTTTCTCTCATTTCAGAACACAGATATGTAGAACTCTTCTTGAATTCTACAGCAGGAGCAAGCGGT

**May 23, 2025**

1. Review RBPnet paper, downloaded pretrained models, ran HNRNPH1 test with PRPF8\_HepG2.model.h5
2. Uploading FLASH data
3. SpliceAI research

**May 27, 2025**

1. Retry training with PRPF8\_siControl.norm.genome.cmd.beds, merged and split into positive and negative strands.

cat \*.bed | \

sort -k1,1 -k2,2n -k3,3 -k6,6 | \

bedtools groupby -i stdin -g 1,2,3,6 -c 5 -o sum | \

awk '{OFS="\t"}{print $1, $2, $3, $4$5}' > merged.bedgraph

then separate bedgraph into plus and minus strands by:  
grep - merged.bedgraph > merged.neg.bedgraph  
grep -v - merged.bedgraph > merged.pos.bedgraph

then ucsc bedgraphtobigwig on both

bedGraphToBigWig mergedsiEIF4A3.neg.bedgraph ~/advbfx/rbpnet/ref/genome.txt mergedsiEIF4A3.neg.bw

created new dataspec.yml file and new train.tfrecord

1. Use pretrained model output to plot and check for expected peaks.
2. Check for length sed -n '2p' HNRNPH1test1.fa | tr -d '\n' | wc -c

Model Inputs:

['sequence']

Model Outputs:

['PRPF8\_mixing\_coefficient', 'PRPF8\_profile', 'PRPF8\_profile\_control', 'PRPF8\_profile\_target']

Train Dataset:

({'sequence': TensorSpec(shape=<unknown>, dtype=tf.float32, name=None)}, {'PRPF8\_profile': TensorSpec(shape=<unknown>, dtype=tf.float32, name=None), 'PRPF8\_profile\_control': TensorSpec(shape=<unknown>, dtype=tf.float32, name=None)})

Validation Dataset:

NONE

Callbacks:

[<rbpnet.callbacks.LRAdjustFactor object at 0x7f9dd5f7e140>, <keras.src.callbacks.EarlyStopping object at 0x7f9dd5fb0130>, <keras.src.callbacks.TensorBoard object at 0x7f9dd4fbaec0>, <keras.src.callbacks.CSVLogger object at 0x7f9dd4fba020>]

Traceback (most recent call last):

File "/home/mikej10/miniconda3/envs/rbpnet-env/bin/rbpnet", line 8, in <module>

sys.exit(main())

File "/home/mikej10/miniconda3/envs/rbpnet-env/lib/python3.10/site-packages/click/core.py", line 1442, in \_\_call\_\_

return self.main(\*args, \*\*kwargs)

File "/home/mikej10/miniconda3/envs/rbpnet-env/lib/python3.10/site-packages/click/core.py", line 1363, in main

rv = self.invoke(ctx)

File "/home/mikej10/miniconda3/envs/rbpnet-env/lib/python3.10/site-packages/click/core.py", line 1830, in invoke

return \_process\_result(sub\_ctx.command.invoke(sub\_ctx))

File "/home/mikej10/miniconda3/envs/rbpnet-env/lib/python3.10/site-packages/click/core.py", line 1226, in invoke

return ctx.invoke(self.callback, \*\*ctx.params)

File "/home/mikej10/miniconda3/envs/rbpnet-env/lib/python3.10/site-packages/click/core.py", line 794, in invoke

return callback(\*args, \*\*kwargs)

File "/home/mikej10/miniconda3/envs/rbpnet-env/lib/python3.10/site-packages/rbpnet/bin/train.py", line 23, in main

train(list(train\_data), dataspec, config, output, val\_data=([validation\_data] if validation\_data is not None else None))

File "/home/mikej10/miniconda3/envs/rbpnet-env/lib/python3.10/site-packages/gin/config.py", line 1605, in gin\_wrapper

utils.augment\_exception\_message\_and\_reraise(e, err\_str)

File "/home/mikej10/miniconda3/envs/rbpnet-env/lib/python3.10/site-packages/gin/utils.py", line 41, in augment\_exception\_message\_and\_reraise

raise proxy.with\_traceback(exception.\_\_traceback\_\_) from None

File "/home/mikej10/miniconda3/envs/rbpnet-env/lib/python3.10/site-packages/gin/config.py", line 1582, in gin\_wrapper

return fn(\*new\_args, \*\*new\_kwargs)

File "/home/mikej10/miniconda3/envs/rbpnet-env/lib/python3.10/site-packages/rbpnet/train.py", line 136, in train

history = model.fit(train\_dataset, validation\_data=val\_dataset, epochs=epochs, verbose=2, callbacks=callbacks)

File "/home/mikej10/miniconda3/envs/rbpnet-env/lib/python3.10/site-packages/keras/src/utils/traceback\_utils.py", line 70, in error\_handler

raise e.with\_traceback(filtered\_tb) from None

File "/home/mikej10/miniconda3/envs/rbpnet-env/lib/python3.10/site-packages/rbpnet/callbacks.py", line 29, in on\_epoch\_end

A screenshot of a computer screen

AI-generated content may be incorrect. raise ValueError("No validation loss ('val\_loss') found in logs.")

ValueError: No validation loss ('val\_loss') found in logs.

In call to configurable 'train' (<function train at 0x7f9dd9eb6950>)

Prioritization: **Jupyter Notebook** > **Splice AI** > **Proteome data added to heatmap**

**~3 slides**

**May 28, 2025**

1. Split sequences\_cln.fa into 64 .fa files and ran through slurm array to produce 64 batch.tsv files, concatenated to pred.tsv (1,157,854 lines)

Pred.tsv

>RHOC\_\_chr1\_112705118-112707127 PRPF8\_HepG2\_profile,PRPF8\_HepG2\_profile\_control,PRPF8\_HepG2\_profile\_target,PRPF8\_HepG2\_mixing\_coeff

SEQ

Profile

Profile\_Control

Profile\_target

Mixing\_coefficient

Count of introns - wc -l introns.tsv 192950 introns

Sequences\_cld.fa 385930 lines /2 = 192965 introns

Count of predictions grep -c "^>" pred/pred.tsv 192965

Confirmed PSI Table contains 192965 introns after filtering to HsaIN events

cat slurm-26493882.out

Total blocks: 192976

Duplicate blocks: 0

Unique blocks: 192976

wc -l predtrim.tsv 1157790 predtrim.tsv

1157790 / 6 = 192,965

**May 30, 2025**

1. Run spliceAI with 10kb max sequence
2. Email Marc Horlacher
3. pip install ipykernel
4. python -m ipykernel install --user --name=spliceai-env --display-name "Python (spliceai-env)"
5. restart VS Code

**June 2, 2025**

1. regenerated flank 50 and 400 fa files with bedtools slop
2. Fixed ipynb to run spliceai env
3. Troubleshooting context trimming, next step run all inference through slurm

**June 3, 2025**

1. Check RBPnet predictions, concatenate and plot sum against PSI
   1. Done concat, redo sum, plotted all tissue groups
   2. sbatch -p cpu --mem 30G --cpus-per-task 4 --time 240 --wrap="python3 /scratch/prj/ppn\_rnp\_networks/users/mike.jones/software/rbpnet/sumpredictions.py"
2. SpliceAI checklist
   1. Check paths are accurate
   2. Test on srun -p interruptible\_gpu --gres gpu --constraint a40 --pty /bin/bash -l
   3. Created batches.tsv
   4. Adjusted to matrix file output

**June 4, 2025**

1. Running test again sbatch -p gpu --gres gpu --time 120 --wrap="testsplice.py"
2. Set up Wohl access and Fire safety training
3. Successfully created splicemini\_triplets.tsv in local notebook
4. Waiting for gpu to run all spliceai inferences
5. Pull clusters into rbp pred merged dataset
6. Created merged dataset and fixed heatmap inconsistencies, need to look into duplicates in the expression gene list
7. Sbatch -p gpu runsplice.sh

**June 6, 2025**

1. Continued RBPnet prediction analysis
2. Merged rbpnet data, fixing heatmap code DONE
3. Merging transposons, intron classifications, liftOver done, minor introns
4. NEED TO EMAIL Marc and Urlich

**June 9, 2025**

1. Email Marc and Ulrich
2. Fuzzy merge Braunschweig classes 160,000/202553, 883 introns unmapped in liftOver, 12,600 mismatch genes, adjust to 25 tolerance, ~1600 duplicate lines
   1. Check gene mismatches
   2. Remove duplicate lines
3. Merge minor introns annotations, check duplicates, no intron class divergence in dupes, removed and exported, from 77,748 to 37,088
4. Inspect and merge transposable element data
5. Create RBP + SpliceAI combined bed
   1. Process RBPnet prediction file
   2. Replace header
   3. Convert to bedfile then bedgraph

Chat input:  
I want to convert my rbpnet output into bed6 format for visualization. The file called pred.tsv has 192965 introns represented, with six lines per intron.

this is the format, with many predictions on each line covering the region in the fasta header.

i. FASTA header

ii. Sequence

iii. Profile

iv. Profile\_control

v. Profile\_target – understood to be mixing of profile and control

vi. Mixing coefficients – higher indicates profile dominance over control signal

Give a bash or python script to:

extract chromosome from the fasta header for column 1, it has :: in front and : after. extract start coordinate from fasta header immediately after chromosome to column 2. Add 1 to start coordinate for column 3. extract gene name after > in fasta header for column 4 "name". extract profile target from the 5th line of each intron entry (5, 11, 17, etc.) for signal in column 5. Strand goes in column 6 from the header. Each line of the bed file will have one prediction at single nucleotide resolution, so each intron be represented by lines equal to the number of predictions in line 5 of each entry in the final bed file.   
header format: >PSMC6::14:52718988-52720910(+)  
BE VERY SPECIFIC IN MAKING ONE LINE OF THE BED FILE PER PREDICTION IN INTRON LINE 5. YOU WILL MAKE A NEW LINE FOR EACH PREDICTION BY using the end coordinate from the previous line.  
summary:  
script that converts a string of prediction data into bed format. First entry for an intron uses start coordinate and then adds one for each following line to get next start coordinate. One prediction per row. Once you have made a new row for each prediction, then you can move to the next intron prediction block of six lines.

**June 10, 2026**

1. Converting rbpnet predictions bed to bedgraph - sbatch -p cpu --time=120 --mem=12G --cpus-per-task=4 --wrap="bedtools genomecov -i predictions.bed -g /scratch/prj/ppn\_rnp\_networks/shared/references/genomes/homo\_sapiens/GRCh38.p14-GencodeRelease44/hg38.genome -bg > predictions1.bedgraph"
   1. Filtered all values less than 0.000001
      1. 429520339 removed
      2. awk '$4 < 0.000001' PRPF8\_eCLIP\_RBPbinding\_Prediction.bedgraph > PRPF8\_eCLIP\_RBPbinding\_Predictionfilter1e-5.bedgraph
2. Convert spliceAI inferences to bedgraph, concatenation w concat.sh
   1. Format = ID [Acceptor,Donor,Position]...
      1. Remove headers from each file
         1. grep -P '^ID\s+\[Acceptor,Donor,Position\]' inf.tsv | wc -l (Only 1 line matched, headers were sticking to last row of previous file)
      2. converted to bed format, cut to bedgraph, awk to remove +1 billion 0’s
         1. cut -f1-3,5 inf.bed > inf.bedgraph
         2. awk '$5 != 0' CrypticSS\_Donor.bed > CrypticSS\_Donor\_Filtered.bed
         3. sort -k1,1 -k2,2n pred.bedgraph > pred.sorted.bedgraph
      3. repeat bed generation for Donor
3. Remove duplicate genes from Braunschweig file
   1. Done
4. Merge with minor intron classes
   1. Merged 36592 / 37088 98.7% matched with spliceosome class

**June 11, 2025**

* Convert bedgraphs to bigwigs x3
  + bedGraphToBigWig CrypticSS\_Donor\_InferenceSort.bedgraph /scratch/prj/ppn\_rnp\_networks/shared/references/genomes/homo\_sapiens/GRCh38.p14-GencodeRelease44/hg38.genome CrypticSS\_Donor\_Inference.bw
  + Possible error in bedgraph creation led to duplicates in bedgraph files and error in bedgraphtobigwig
* Test unique intron\_start and end in minor spliceosome file
  + Need to start new R Granges markdown file
* Merge Transposons file
* Regenerate PRPF8 prediction bed (confirmed+ and – present in bed, filter, then separate positive negative strands, convert to bedgraph and get two resulting bws
  + awk '$5 < 0.000001' PRPF8\_eCLIP\_RBPbinding\_Prediction.bed > PRPF8\_eCLIP\_RBPbinding\_Predictionfilter1e-5.bed
* Go back and confirm acceptor and donor output from SpliceAI predictions
* Parse one gene from PRPF8 predictions, test bed file conversion and visualize
* Check strandedness on getfasta calls
  + Checked out, needed to adjust strandedness in bed generation
    - 1. Start by regenerating flank 50 fasta for PRPF8 predictions /scratch/prj/ppn\_rnp\_networks/shared/references/genomes/homo\_sapiens/GRCh38.p14-GencodeRelease44/GRCh38.primary\_assembly.genome.fa
         1. wc -l RBPnet\_Prediction\_Input.fa 385930
      2. Repeat with full header for SpliceAI 400 flank bed input wc -l SpliceAI\_Prediction\_Input.fa 385930
      3. Fasta files look ok, red herring
      4. Regenerating bed files with stranded logic
      5. Filter again
         1. awk '$5 < 0.000001' PRPF8\_eCLIP\_RBPbinding\_Prediction.bed > PRPF8\_eCLIP\_RBPbinding\_Predictionfilter1e-5.bed

**June 12, 2025**

1. No conflict after separating into positive and negative bedgraphs

awk '{OFS="\t"}{print $1, $2, $3, $6$5}' PRPF8\_eCLIP\_RBPbinding\_PredictionSorted.bed > PRPF8\_eCLIP\_RBPbinding\_Prediction.bedgraph

**grep - PRPF8\_eCLIP\_RBPbinding\_Prediction.bedgraph > PRPF8\_eCLIP\_RBPbinding\_Prediction.neg.bedgraph**

**grep + PRPF8\_eCLIP\_RBPbinding\_Prediction.bedgraph > PRPF8\_eCLIP\_RBPbinding\_Prediction.pos.bedgraph**

awk '

{

key = $1 FS $2 FS $3

sum[key] += $4

count[key] += 1

}

END {

for (k in sum) {

split(k, a, FS)

avg = sum[k] / count[k]

print a[1], a[2], a[3], avg

}

}' PRPF8\_eCLIP\_RBPbinding\_Prediction.neg.bedgraph > PRPF8\_eCLIP\_RBPbinding\_Prediction.neg.collapsed.bedgraph

1. Convert bedGraphToBigWig

bedGraphToBigWig PRPF8\_eCLIP\_RBPbinding\_Prediction.collapsed.pos.bedgraph /scratch/prj/ppn\_rnp\_networks/shared/references/genomes/homo\_sapiens/GRCh38.p14-GencodeRelease44/hg38.genome PRPF8\_eCLIP\_RBPbinding\_Prediction.pos.bw

bedGraphToBigWig PRPF8\_eCLIP\_RBPbinding\_Prediction.collapsed.neg.bedgraph /scratch/prj/ppn\_rnp\_networks/shared/references/genomes/homo\_sapiens/GRCh38.p14-GencodeRelease44/hg38.genome PRPF8\_eCLIP\_RBPbinding\_Prediction.neg.bw

Visualize in IGV

1. Start minor intron conversion R Workbook
   1. Check duplicate intron\_names
   2. 1264 = number of conflicted EVENT IDs
   3. 3460 = number of full duplicated rows involved
2. Sort donor bed, split into pos/neg and convert

sort -k1,1 -k2,2n -k3,3n -k6,6 CrypticSS\_Donor\_Filtered.bed > CrypticSS\_Donor\_Sorted.bed

awk '{OFS="\t"}{print $1, $2, $3, $6$5}' CrypticSS\_Donor\_Sorted.bed > Cryptic\_Donor\_Inferences.bedgraph

grep - Cryptic\_Donor\_Inferences.bedgraph > Cryptic\_Donor\_Inferences.neg.bedgraph

grep + Cryptic\_Donor\_Inferences.bedgraph > Cryptic\_Donor\_Inferences.pos.bedgraph

awk '

{

key = $1 FS $2 FS $3

sum[key] += $4

count[key] += 1

}

END {

for (k in sum) {

split(k, a, FS)

avg = sum[k] / count[k]

print a[1], a[2], a[3], avg

}

}' Cryptic\_Donor\_Inferences.pos.bedgraph > CrypticSS\_Donor\_Inferences.pos.collapsed.bedgraph

bedGraphToBigWig CrypticSS\_Donor\_Inferences.pos.bedgraph /scratch/prj/ppn\_rnp\_networks/shared/references/genomes/homo\_sapiens/GRCh38.p14-GencodeRelease44/hg38.genome CrypticSS\_Donor\_Inferences.pos.bw

bedGraphToBigWig CrypticSS\_Donor\_Inferences.neg.bedgraph /scratch/prj/ppn\_rnp\_networks/shared/references/genomes/homo\_sapiens/GRCh38.p14-GencodeRelease44/hg38.genome CrypticSS\_Donor\_Inferences.neg.bw

**June 13, 2025**

* Pulled HNRNPH1 predictions, converted to bed and bedgraph

grep 'HNRNPH1' -A 5 PRPF8\_eCLIP\_RBPbinding\_Prediction.tsv > HNRNPH1\_PRPF8\_Prediction.tsv

* Braunschweig intron types missing from 18723 events

sum(mergedPSI50$braun\_name == "" | is.na(mergedPSI50$braun\_name))

**June 16, 2025**

* PRPF8 BED conversion debug
  + Extracted HNRNPH1 prediction from large tsv file and plot matches test prediction from May
  + Aligned sequence with predictions and compared to bedgraph, which appears to have positions reversed
  + Adjust python script accordingly

sbatch -p cpu --time=120 --mem=48G --cpus-per-task=30 --wrap="python3 Convert\_PRPF8TSV\_2bed.py"

* Need to run in rbpnet environment created run\_convert2bed\_rbpnet.sh
* batch job 2693009
* convert to bed graph 🡪 srun -p cpu --time 240 --mem=48G --cpus-per-task=20 --pty /bin/bash

awk -F'\t' '{OFS="\t"; sign=($6=="+"?1:-1); print $1, $2, $3, sign \* $5}' PRPF8\_eCLIP\_RBPbinding\_Prediction.filtered.bed > PRPF8\_eCLIP\_RBPbinding\_Prediction.bedgraph

* Then separate neg and pos bedgraphs
* Collapse and convert to bigwig with Collapse\_Predictions\_Bed.sh and ucsc bedGraphToBigWig

Test SpliceAI prediction bed conversion with HNRNPH1::chr5:179616908-179617100

grep HNRNPH1::chr5:179616908-179617100 CrypticSS\_All\_Inferences.tsv > CrypticSS\_HNRNPH1\_Inference.tsv

created run\_convert2bed\_splice.sh and Convert\_CrypticSS\_Test\_2bed.py

**June 17, 2025**

* Creating PRPF8 bedgraph, next separate pos/neg, collapse and convert to bw
  + awk '$4 <= 0 { printf "%s\t%s\t%s\t%.6f\n", $1, $2, $3, $4 }' PRPF8\_eCLIP\_RBPbinding\_Prediction.bedgraph > PRPF8\_eCLIP\_RBPbinding\_Prediction.neg.bedgraph
    - neg strand has 300 million too many lines, create again with awk
  + grep - PRPF8\_eCLIP\_RBPbinding\_Prediction.bedgraph > PRPF8\_eCLIP\_RBPbinding\_Prediction.neg.bedgraph
  + bash Collapse\_Predictions\_Bed.sh
  + sort -k1,1 -k2,2n -k3,3n /scratch/prj/ppn\_rnp\_networks/users/mike.jones/data/rbpnet/PRPF8\_eCLIP\_RBPbinding\_Prediction.ne6.bedgraph > /scratch/prj/ppn\_rnp\_networks/users/mike.jones/data/rbpnet/sorted2.bedgraph
  + sbatch -p cpu --time=120 --cpus-per-task=20 --mem=30G --wrap="bedGraphToBigWig PRPF8\_eCLIP\_RBPbinding\_Prediction.collapsed.pos.bedgraph /scratch/prj/ppn\_rnp\_networks/shared/references/genomes/homo\_sapiens/GRCh38.p14-GencodeRelease44/hg38.genome PRPF8\_eCLIP\_RBPbinding\_Prediction.pos.bw"
* Run SpliceAI test
  + Create HNRNPH1 and STMN2 test tsv, convert to bed
  + Sbatch run\_convert2bed\_splice.sh
  + *SF3B1*cryptic splice site present in intron 4 <https://pmc.ncbi.nlm.nih.gov/articles/PMC6097662/>
  + Check for ARGLU1 ultra conserved element in intron 2 and OGT intron 4 <https://pmc.ncbi.nlm.nih.gov/articles/PMC5389617/> <https://pubmed.ncbi.nlm.nih.gov/40265571/>
* Combine Transposon annotations
  + AnnotationsPart2.Rmd
  + Look for SVA\_E/F, LTR5\_Hs, L1HS
  + Matched 1331 out of 4070
* Explore combined dataset, summarize variables against Cluster1-4 identity

**June 18, 2025**

* Redid prediction screenshots with seq alignment and cassette exons highlighted
* Installed clippy and dependencies
* Regenerating bed files
* Fix transposon annotations merge
* To do: fix clippy install, read Varcarcel sup table 5, transposons merge w ID

**June 23, 2025**

* Running rbpnet predictions through clippy split by chrom – verify if output is OK
* Concatenate rbpnet peaks – values too low
* Bedtools intersect

awk '{print $5}' PRPF8\_eCLIP\_RBPbinding\_Prediction.filtered.bed | awk '

{ sum += $1; vals[NR] = $1 }

END {

n = NR

mean = sum / n

asort(vals)

if (n % 2) {

median = vals[int(n/2)+1]

} else {

median = (vals[n/2] + vals[n/2+1]) / 2}

print "Mean:", mean

print "Median:", median}'

Mean: 0.000237747 Median: 1.185e-05

wc -l PRPF8\_eCLIP\_RBPbinding\_Prediction.filtered.bed 811125603

Sum = 192842.678736441

Max 0.79258609

awk '{print $5}' grouped\_Prpf8.bed | awk '

{ sum += $1; vals[NR] = $1 }

END {

n = NR

mean = sum / n

asort(vals)

if (n % 2) {

median = vals[int(n/2)+1]

} else {

median = (vals[n/2] + vals[n/2+1]) / 2}

print "Mean:", mean

print "Median:", median}'

Mean: 1.56534 Median: 1

Wc -l 3108179

Sum = 4,865,356.655

Max 3800

Try factor 4795 to match max – not over 5 in test file

New approach, normalize number of lines then compare sums

261 x more lines in prpf8 predictions bed. Sum is 25 times smaller than grouped iCLIP data, new factor 6584

**June 24, 2025**

* Convert bedgraphs from iCLIP and eCLIP to bed, merge and compare signal with PRPF8 predictions
  + Done installed ucsc-bigBedtoBed in ucsc-env
    - cat \*bed | sort -k1,1 -k2,2n -k6,6 | \
  + cat merged\_Prpf8.bed | bedtools groupby -g 1,2,3,6 -c 5 -o sum > grouped\_Prpf8.bed
  + fix by adding column 4 and moving 4 to 6 for bed6 format

awk 'BEGIN{OFS="\t"} {print $1, $2, $3, ".", $5, $4}' grouped\_Prpf8.bed > grouped\_eCLIP\_HepG2.bed

sbatch -p cpu --time=120 --cpus-per-task=24 --mem=24G --wrap="mkdir -p f50; for f in chr\*.bed; do awk -v OFS='\t' '{\$5 = \$5 \* 50; print}' \$f > f50/\$f; done"

awk 'BEGIN{max=0} {if($5>max) max=$5} END{print max}' your\_file.bed

awk '$5 > 5' chr10.bed | wc -l

TEST clippy

clippy -i /scratch/prj/ppn\_rnp\_networks/users/mike.jones/data/rbpnet/split\_beds/f4795/chr5.bed -o Prpf8\_chr5\_f4795 -a /scratch/prj/ppn\_rnp\_networks/shared/references/genomes/homo\_sapiens/GRCh38.p14-GencodeRelease44/gencode.v44.primary\_assembly.annotation.gtf -g /scratch/prj/ppn\_rnp\_networks/shared/references/genomes/homo\_sapiens/GRCh38.p14-GencodeRelease44/GRCh38.primary\_assembly.genome.fa.fai

awk 'BEGIN{OFS="\t"} {print $1, $2, $3, ".", $4, "+"}' PRPF8\_eCLIP\_HepG2\_R2\_pos.bedgraph > PRPF8\_eCLIP\_HepG2\_R2\_pos.bed

rds: [k24121579@erc-hpc-login2:/rds/prj/ppn\_comp\_analysis\_rnp/users/mike.jones/rbpnet$](mailto:k24121579@erc-hpc-login2:/rds/prj/ppn_comp_analysis_rnp/users/mike.jones/rbpnet$)

clippy -i PRPF8\_eCLIP\_HepG2\_xlinks.bed -o PRPF8\_eCLIP\_HepG2 -a /scratch/prj/ppn\_rnp\_networks/shared/references/genomes/homo\_sapiens/GRCh38.p14-GencodeRelease44/gencode.v44.primary\_assembly.annotation.gtf -g /scratch/prj/ppn\_rnp\_networks/shared/references/genomes/homo\_sapiens/GRCh38.p14-GencodeRelease44/GRCh38.primary\_assembly.genome.fa.fai

**June 25, 2025**

* All factors above 5000+ appear identical, repeat from 500, 1000, 2000, 3000, 4000

for factor in 100 150 200 250 300 350 400 450; do mkdir -p f$factor; awk -v OFS='\t' -v m=$factor '{$5 = $5 \* m; print}' chr5.bed > f$factor/chr5.bed; done

1. Generate full PRPF8 prediction beds for f50, f100, and f250, concatenate and sort - **done**
2. Create three crosslink files for global signal, eclip signal (PRPF8\_eCLIP\_HepG2\_xlinks.bed), and iclip signal (PRPF8\_iCLIP\_HepG2\_xlinks.bed)
   1. eCLIP, merge pos and neg bedgraphs, convert to bed, merge, sort, groupby
3. Create canonical splice site bed - **done**
   1. Strand aware bash script takes start for pos and end for neg as splice site, subtracts 5 for start and add 5 for end
4. Intersect against splice site predictions, 3 sets of real data, and canonical splice sites

**True pos** - PRPF8 prediction agrees with iCLIP, eCLIP, canonical splice **AND/OR** splice prediction  
**True neg** - Pick a protein that binds somewhere else like 3'UTRs and use its binding sites as "true negatives"  
**False pos** - PRPF8 binding prediction unsupported by ANY of combined e/iCLIP xlinks, canonical splice, predicted splice. *We would expect that predicted splice would cover most new Prpf8 predictions. Extend the splice sites (both canon and predicted) by +-50nt because we know Prpf8 binding can be offset. Later we could add branch points too, but for now this can just be a caveat of the analysis that is commented on - along with perhaps there are genuine Prpf8 binding sites that do not overlap in any way with 5'SS - tbh you can empirically assess this by look at iclip/eclip peak overlap with canon/predicted splice***.**  
**False neg** - No PRPF8 prediction where signal exists in eCLIP or iCLIP xlinks, and signal in canonical splice or predicted splice. 

PLOT **PR Curve** for determining the optimal PRPF8 binding prediction model (f50, f100, f250)

Y axis = precision TP / TP + FP, X axis = sensitivity or TPR (TP / TP + FN)

**June 26, 2025**

Create dissertation outline

Potentially filter eCLIP file before merging with iCLIP data – filtered out < 5

Check bedgraphs visually

cat \*neg.bedgraph | sort sort -k1,1 -k2,2n > merged\_eCLIP\_HepG2\_neg.bedgraph

Bedtools intersect

Intersect Plan

1. Splice Site Predictions
   1. Filter splice site predictions by 2e-4
2. Canonical Splice sites
   1. Create larger flanks for False Positive detection
3. eCLIP xlinks
4. iCLIP xlinks

|  |  |  |  |
| --- | --- | --- | --- |
|  | F50 | F100 | F250 |
| TP | 191,108 | 292,065 | 470,166 |
| TN | - | - | - |
| FP | 32,016 | 110,420 | 379,121 |
| FN | 43,701 | 36,235 | 30,433 |
| TP - iCLIP | 136155 | 213953 | 358114 |
| Novel predictions % |  |  |  |

**FP** = intersect -s -v A, B, C, and D (extend B by +- 50 nt)

bedtools intersect \

-a ../rbpnet/peaks/f250/rbpnet\_clippy\_f250\_rollmean10\_minHeightAdjust1.0\_minPromAdjust1.0\_minGeneCount5\_Peaks.bed \

-b ../eCLIP/PRPF8\_eCLIP\_HepG2\_rollmean10\_minHeightAdjust1.0\_minPromAdjust1.0\_minGeneCount5\_Peaks.bed \

../prpf8/PRPF8\_iCLIP\_HepG2\_xlinks.bed \

-v \

-s \

> Intersect\_f250\_Clip\_collapsed\_overlaps.bed

awk '$7 != 0' Intersect\_f50\_TPR\_collapsed\_overlaps.bed > Intersect\_f50\_TPR\_filtered.bed

**TPR Rate (sensitivity)** - Intersect all four files, total count of rows after removing coordinates with 0 sum in column 7, True positives divided by TP + FN

**Precision –** TP / TP + FP

988035 Intersect\_f250\_TPR\_collapsed\_overlaps.bed

470166 Intersect\_f250\_TPR\_filtered.bed

458427 Intersect\_f100\_TPR\_collapsed\_overlaps.bed

292065 Intersect\_f100\_TPR\_filtered.bed

242946 Intersect\_f50\_TPR\_collapsed\_overlaps.bed

191108 Intersect\_f50\_TPR\_filtered.bed

**FPR** – intersect all four files with -v instead of -c and wider splice site bed

379121 Intersect\_f250\_FPR\_collapsed\_overlaps.bed

110420 Intersect\_f100\_FPR\_collapsed\_overlaps.bed

32016 Intersect\_f50\_FPR\_collapsed\_overlaps.bed

**FNR** – combine and bedtools merge eCLIP/iCLIP files, same for splice site files

30433 Intersect\_f250\_FN.bed

36235 Intersect\_f100\_FN.bed

43701 Intersect\_f50\_FN.bed

286638 Intersect\_f100\_FNwide.bed

259723 Intersect\_f250\_FNwide.bed

310397 Intersect\_f50\_FNwide.bed

bedtools intersect \

-a merged\_clip\_widesplice\_overlap.bed \

-b ../rbpnet/peaks/f100/rbpnet\_clippy\_f100\_rollmean10\_minHeightAdjust1.0\_minPromAdjust1.0\_minGeneCount5\_Peaks.bed \

-v \

-s \

> Intersect\_f100\_FNwide.bed

**July 6, 2025** – Pearson’s Chi squared test: independence between two categorical variables (Is there association between a and b) eg. Count and category

| **CLUSTER**  **<chr>** | **Simple\_repeat**  **<dbl>** | **Low\_complexity**  **<dbl>** | **Retroposon/SVA**  **<dbl>** | **dna**  **<dbl>** | **line**  **<dbl>** | **ltr**  **<dbl>** | **sine**  **<dbl>** | **satellite**  **<dbl>** | **rna**  **<dbl>** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Cluster\_1** | **47057** | **14895** | **1357** | **40162** | **54546** | **24967** | **75453** | **158** | **3001** |
| **Cluster\_2** | **11089** | **3636** | **300** | **9109** | **12398** | **5853** | **16904** | **57** | **632** |
| **Cluster\_3** | **7865** | **3035** | **240** | **6172** | **8294** | **5603** | **10359** | **66** | **499** |
| **Cluster\_4** | **3463** | **1217** | **85** | **2656** | **3581** | **1985** | **4877** | **15** | **221** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Transposon Type** | **Chi Squared** | **df** | **P-value** |
| **Simple Repeat** | **429.05** | **3** | **< 2.2e-16** |
| **Low Complexity Region** | **460.42** | **3** | **< 2.2e-16** |
| **Retrotransposon** | **17.94** | **3** | **0.0004533** |
| **DNA** | **159.84** | **3** | **< 2.2e-16** |
| **LINE** | **230.90** | **3** | **< 2.2e-16** |
| **LTR** | **1479.20** | **3** | **< 2.2e-16** |
| **SINE** | **200.50** | **3** | **< 2.2e-16** |
| **Satellite** | **63.24** | **3** | **1.196e-13** |
| **RNA** | **26.60** | **3** | **7.138e-06** |

**LTR-containing introns are much more common in some clusters, transposon presence is not evenly distributed across clusters — some clusters are enriched or depleted**

**Call:**

**lm(formula = median\_PSI ~ sine + line + dna + ltr + Retroposon\_SVA +**

**satellite + Low\_complexity + Simple\_repeat + rna, data = merged\_for\_lm)**

**Residuals:**

**Min 1Q Median 3Q Max**

**-33.264 -13.843 -10.969 -5.681 91.344**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Estimate** | **Std. Error** | **T value** | **Pr(>|t|)** | **Significance** |
| **Intercept** | **14.1505** | **0.1113** | **127.103** | **< 2e-16** | **\*\*\*** |
| **SINE** | **-2.8717** | **0.1712** | **-16.772** | **< 2e-16** | **\*\*\*** |
| **LINE** | **-0.5165** | **0.1770** | **-2.918** | **0.00352** | **\*\*** |
| **DNA** | **-1.0497** | **0.1791** | **-5.860** | **4.64e-09** | **\*\*\*** |
| **LTR** | **6.5454** | **0.2013** | **32.509** | **< 2e-16** | **\*\*\*** |
| **Retrotransposon/SVA** | **-0.8651** | **0.6697** | **-1.292** | **0.19645** |  |
| **Satellite** | **11.9817** | **1.7017** | **7.041** | **1.92e-12** | **\*\*\*** |
| **Low Complexity Region** | **2.5902** | **0.2255** | **11.486** | **< 2e-16** | **\*\*\*** |
| **Simple Repeat** | **1.8291** | **0.1685** | **10.854** | **< 2e-16** | **\*\*\*** |
| **RNA** | **-1.0565** | **0.4610** | **-2.292** | **0.02192** | **\*** |

**Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1**

**Residual standard error: 29.25 on 181328 degrees of freedom**

**Multiple R-squared: 0.01049, Adjusted R-squared: 0.01044**

**F-statistic: 213.5 on 9 and 181328 DF, p-value: < 2.2e-16**

**Median PSI analysis**

* **sine, line, dna, Retroposon\_SVA, and rna are negatively associated with mean PSI—introns with these elements tend to have lower retention**
* **ltr, satellite, Low\_complexity, and Simple\_repeat are positively associated with PSI—introns with these elements are more likely to be retained.**
* **R² = 0.01049 → only ~1% of PSI variation is explained by transposon presence alone**
* **With ~180k introns, that’s still significant (p < 2e-16), but biological variation is likely driven by more than transposons (e.g. RBP binding, splice site strength, secondary structure, or chromatin).**

**Intersect to remove canonical splice sites then intersect with prpf8 predictions**

bedtools intersect \

-a ../splice/Splice\_All.filtered.2e-4.bed \

-b ../rbpnet/Canonical\_splice\_sites.bed \

-v \

-s \

> Intersect\_SpliceSites\_CrypticOnly.bed

bedtools intersect \

-a ../rbpnet/peaks/f100/rbpnet\_clippy\_f100\_rollmean10\_minHeightAdjust1.0\_minPromAdjust1.0\_minGeneCount5\_Peaks.bed \

-b Intersect\_SpliceSites\_CrypticOnly.bed \

-c \

-s \

> Intersect\_f100\_CrypticPRPF8.bed

**Intersect to find true positive that are not seen in iCLIP xlinks**

bedtools intersect \

-a Intersect\_f100\_TPR\_filtered.bed \

-b ../prpf8/PRPF8\_iCLIP\_HepG2\_xlinks.bed \

-v \

-s \

> Intersect\_f100\_Clip\_collapsed\_overlaps.bed

wc -l 358114 Intersect\_f250\_Clip\_collapsed\_overlaps.bed

213953 Intersect\_f100\_Clip\_collapsed\_overlaps.bed

136155 Intersect\_f50\_Clip\_collapsed\_overlaps.bed

72576 Prpf8\_grouped\_rollmean10\_minHeightAdjust1.0\_minPromAdjust1.0\_minGeneCount5\_Peaks.bed

188 - 493% increase in PRPF8 peaks

Call:

polr(formula = CLUSTER ~ predict\_overlap\_sum, data = merged,

Hess = TRUE, method = "logistic")

Coefficients:

Value Std. Error t value

predict\_overlap\_sum 0.0858 0.002206 38.89

Intercepts:

Value Std. Error t value

1|2 0.9631 0.0066 147.0033

2|4 1.9351 0.0080 242.7193

4|3 2.4157 0.0092 263.5696

Residual Deviance: 333537.03

AIC: 333545.03

Call:

polr(formula = CLUSTER ~ log\_LENGTH, data = merged, Hess = TRUE,

method = "logistic")

Coefficients:

Value Std. Error t value

log\_LENGTH 0.09023 0.007163 12.6

Intercepts:

Value Std. Error t value

1|2 1.0988 0.0236 46.5874

2|4 2.0641 0.0240 85.9510

4|3 2.5424 0.0245 103.9614

Residual Deviance: 334852.97

AIC: 334860.97

**August 8th 2025**

Check PSI prediction capability of combined PRPF8 + cryptic splice site predictions against cryptic splice site predictions that do not overlap PRPF8 predictions

bedtools intersect \

-a ../splice/Splice\_All.filtered.2e-4.bed \

-b Intersect\_f250\_CrypticPRPF8.bed \

-v \

-s \

> Intersect\_f250\_CrypticsNoPRPF8Overlap.bed

Check for agreement by intersecting again Only cryptic sites that don’t overlap f250 PRPF8 predictions, selecting second analysis as best option, first version removes all combined predictions from all splice AI predictions but leaves some canonical splice sites that did not overlap with PRPF8 predictions. Second script starts with only cryptics and keep only non-overlapping from source PRPF8 predictions of low precision and high sensitivity.

bedtools intersect \

-a Intersect\_SpliceSites\_CrypticOnly.bed \

-b ../rbpnet/peaks/f250/rbpnet\_clippy\_f250\_rollmean10\_minHeightAdjust1.0\_minPromAdjust1.0\_minGeneCount5\_Peaks.bed\

-v \

-s \

> Intersect\_f250\_CrypticsNoPRPF8OverlapRepeatTest.bed