**Fig2A**

For all RIP fastq files, run STAR alignments to mm10 genome with codes: run\_star.sh and wrapper\_run\_star.sh.

Convert STAR output .sam to .bam format with code wrapper\_convert.sh and convert\_sam\_to\_bam.sh (we have other use of the sam format, so STAR is not coded to output bam)

Sort .bam files using wrapper\_sort.sh and sort\_bam.sh

Index the sorted .bam files, using wrapper\_index.sh and index\_bam.sh

Therefore, for each RBP we have its sorted and indexed bam files.

Run fastq\_totalcount.ipynb to count each fastq file total reads, saved in fastq\_read\_counts.csv

Manually curated fastq\_samples.csv served as metadata for which RBP each fastq file corresponds to. Total 28 RBP including IGG.

Use all transcripts in the table: TSC-exp\_ESC-exp\_Chrom-assoc\_igg\_rpm\_4\_10\_2024\_filtered\_07222024.csv as the full list. In this list, we have both spliced and unspliced version of the same gene. For unspliced transcript, cut it into 25bp window size with no overlap, include strand info (6 col) in the BED file; for spliced transcripts, we need to convert into the genomic coordinates using gencode.vM25.basic.annotation.gtf and cut into 25bp based on that.

Codes: gen\_chunk\_bedfile.R

Output are individual bedfile for each transcripts in the destined folder.

Run multicov\_counts\_allgenes\_08092024.sh to get multicov counts for all sorted and indexed RBP bam files cross all transcripts, using bedfiles generated above. Also save all input sorted and indexed bam file names into multicov\_files\_list\_allgenes\_08092024.txt

Output are individual multicov count file for each transcripts, which contains all RBP counts in the columns. The order of these RBPs are the same as the files in multicov\_files\_list\_allgenes\_08092024.txt.

Analyze the multicov count files using multicov\_analysis\_allgenes.R:

* normalize the counts into rpm by dividing the total reads in fastq files saved in fastq\_read\_counts.csv
* combine replicates of each RBP and save the mean, using metadata saved in fastq\_samples.csv
* subtract igg from each RBP normalized mean counts, if negative after subtraction, set to 0. the drop the IGG column. Now we have 27 RBPs
* calculate Pearson’s correlation for all possible pairs of RBPs (27x26/2=351) for each transcripts. If a network is constructed for a specific transcripts with 27 RBP as nodes, the r value between two RBPs can be viewed as the edge weight. This network would reflect how the 27 RBP interact with the transcript. If two RBPs binds to the transcripts in a similar fashion, they would be very close to each other (high r value). Two components within the same complex would be a good example.
* Save the r values for 351 pairs of RBP as the edges for each transcript, which can be used for plotting network in Gephi, with other files generated in later steps. Here we could have NA values between two RBP if at least one of the RBP has 0 counts throughout the whole transcript, which then has 0 sd and therefore NA for pearson r value.
* Generate weighted adjacency matrix for Leiden clustering by setting all negative and NA r values (edge weights) into 0. Save this as ld matrix file for each transcript.
* (Leiden clustering is performed in Python in a later step, as the Leiden package in R does not work properly with this set of data)
* Integrate all 315 RBP pairs name, and r value across all transcripts into a file that can be used in the later steps for other analysis. comb\_allgenes\_cor\_08092024\_long.csv
* After run Leiden clustering in Python, combine nodes community assignment for all transcripts together for later use. ldcommunity\_allgenes\_08092024.csv. Also save the filtered transcript list for later use. genelist13831.csv
* iterate through all possible RBP pair combinations and calculate the count of each RBP pair being in the same community across all transcript networks, and the percentage of that count. alldoublepro\_ld\_percent\_08092024.csv

All these step are done with code multicov\_analysis\_allgenes.R

In Python perform Leiden clustering of the 27 RBPs across all transcripts, using code multicov\_leiden.ipynb. The results are organized with multicov\_analysis\_allgenes.R as mentioned above.

For XAK, generate the Gephi network in **Fig2A** based on the edge file and Leiden community assignment files generated above. For edge files, delete all negative edges and only show positive edges. Generate network plot with Gephi with Force Atlas 2 layout: scale 5, edge weight influence 3, prevent overlap checked.

Edges (ends with edges\_pos.csv)and nodes (ends with nodes\_ld.csv) files for XAK to generate the networks are in the folder as well.

**Fig2B**

For all the edge files generated above, in order to include transcripts such as Neat1, which has some edge weight = NA, we replaced all NAs in the edge weights by 0. Then we compare the network of 27 RBPs (all paired edge weights of 27 RBPs- 351 edges) of 19295 transcripts against XAK using pearson correlation.

input: comb\_allgenes\_cor\_08092024\_long.csv

code: XAK\_network\_pearson.R

file: XAK\_vs\_all\_pearson\_r\_network\_NA0\_09112024.csv and Fig2B plot

**Fig2C**

i)

For each of XAK networks, across all edges, calculate the distribution of edge weights (pearson r values) within community and inter community. compare the intra and inter community r value distribution to see if there are significant changes.

input: XAK edges and nodes files.

code: XAK\_percentage.R

plot: Fig2C i

ii)

Modularity calculates a global strength of division into communities, compared with a random-graph null model (which preserves each node’s degree but shuffles edges randomly). It gives a single measurement for each network, with value ranging from -1/2 to 1, the closer to 1, the better the community is: network splits cleanly into well-defined, tightly knit “modules” (or communities) with very sparse inter-module connections. Modularity is calculated for each transcript.

input: comb\_allgenes\_cor\_08092024\_long.csv and ldcommunity\_allgenes\_08092024.csv

code: modularity.R

plot: Fig2C ii

iii)

Silhouette width quantifies how well nodes are clustered within communities compared to nodes in other communities based on the distances between them. Calculated the silhouette width for each node in the network across all transcripts.

For each node, the Silhouette width ranges from -1 to 1. A value close to 1 indicates the node is well-matched to its own community and poorly matched to neighboring communities. A value close to -1 indicates the opposite.

Then get the average Silhouette width for a specific transcript.

input: comb\_allgenes\_cor\_08092024\_long.csv and ldcommunity\_allgenes\_08092024.csv

code: silhouette.R

plot: Fig2C iii

**Fig2D**

Across all transcripts (13831), we calculated p values that describe the likelihood that each possible pair of RBPs (351) would be detected in the same community relative to randomized controls, using Poisson Binomial Distribution.

For each transcript network, it can be calculated: what is the probability that any two RBPs are within the same community by dividing the total number of RBP pairs within the same community in that network by 351. This is the weight – the chance of any two RBPs that are in the same community for this transcript. alldbl\_rate\_08092024.csv

Then it is a problem that if we flip 13831 coins each with a certain weight to see Head (same community), what is the chance that we see x% of the coins landed on Head. This is the p val for seeing x% out of the 13831 coins being Head. This is a binomial trial with different weight for each coin flip event, which then follows Poisson Binomial Distribution.

We calculated the probability of x or fewer successes: the pval for seeing a protein doublet reside is the same community in x or fewer transcripts, which corresponds to significantly rare edges.

We calculated the probability of x successes or more. This corresponds to significantly prevalent edges.

alldoublepro\_ld\_pval\_adj\_06182025\_raw.csv

For p value < 1.0E-12 we set to < 1.0E-12 instead of saving the exact number, as we observed numerical instability or rounding errors in the computation of the Poisson binomial cumulative distribution function. Saving small p value as < 1.0E-12 would make the multiple comparison adjustment results more consistent and stable and they won’t affect the actual significance level for these genes or others.

Then use BH for multi-comparison correction.

input: ldcommunity\_allgenes\_08092024.csv and alldoublepro\_ld\_percent\_08092024.csv

code: calculate\_pval\_06182025.R

file: alldoublepro\_ld\_pval\_adj\_06182025.csv

For each 13831 transcript, calculate whether each RBP doublet pairs is inter-0 or intra-1 community from the file: ldcommunity\_allgenes\_08092024.csv

Add the status of whether each edge is inter/intra community of X,A and K to the alldoublepro\_ld\_pval\_adj\_06182025\_raw.csv table.

For the **intra community** protein double pair check the adjpval and see if it is sig prevalent (prev\_adjp<0.05) or sig rare (rare\_adjp<0.05). Take the ratio of prev and rare pairs for each transcript. Check where XAK ratio stand among all transcripts.

input: Leiden node files, alldoublepro\_ld\_pval\_adj\_06182025\_raw.csv

code: XAK\_intra\_rare\_ratio\_06192025.R

file: allgenes\_double\_intra\_rare\_ratio\_06192025.csv and Fig2D