Fig5

unique\_chunk500nt\_k4\_background.py cut unique genecode seqs into chunks. Calculate all gencode chunks vs all gencode chunks r scores and get the threshold value at 97.5% of all the r scores (q975).

seqcompare\_XAK\_rscore\_qlist.py cut Xist intervals and A and K into chunks.Manually assemble Xist by concatenate Xist repeat features and interval chunks. Compare XAK chunks pairwise and save the r scores into a file.

seq\_compare\_with\_density\_plot.py use the quantile info (97.5% r score) and all gencode vs gencode r scores from unique\_chunk500nt\_k4\_background.py and XAK pairwise comparison r scores from seqcompare\_XAK\_rscore\_qlist.py, perform the ks test and plot the density plots of Fig 5B.

seq\_compare\_with\_linear\_plot\_newlegend\_2.py plot transcripts as linear parallel tracks, and connect q975 chunks with colored blocks. q975 chunks are chunks that has a higher similarity than the threshold value at 97.5% of all gencode chunk r scores. This generates the plots for Fig 5CD upper panels and Fig S3 upper panels.

seq\_compare\_with\_hist\_plot\_newlegend\_2.py plot one transcript chunks as x axis and plot sig chunk pairs as color coded histogram, stacked, as shown in Fig 5CD lower panels and Fig S3 lower panels. Also in this code legends of XAK were plotted as colored rectangles according to their length, color code the line based on the feature and label coordinates of the feature for Xist. For A and K just color code the blocked based on coordinates. These are used in Fig 5E.

chunk\_num\_sig\_simulation.py tests whether the num of chunk pairs pass q975 threhold of X vs A is significant. Get the seekr pearson correlation matrix between X chunks and A chunks. And then shuffle the values of this matrix. For each A chunk (rows), count how many X chunks have r score more than 97.5% quantile threshold (0.27088). And then filter with the kmask (for each A chunk whether there is also at least one sig K chunk). This gives a vector of X sig chunk number but only at those A chunk locations that also has at least one sig K. p\_total: count the number that is > 0 in the final vector. This gives the simulated p value of 0.002.

XAK\_chunkrscore\_organization.R organize data of chunk seekr comparisons between XAK (from seqcompare\_XAK\_rscore\_qlist.py): Pin on X and assign all related q975 chunks (chunks that has a higher similarity than the threshold value at 97.5% of all gencode chunk r scores) of A and K to each chunk of X (target seq), saving the chunkname, r value, and the sequences. Do the same for A and K. For each chunk of XAK, compile target seq and all related seqs together into one fasta file and use that as input for MEME motif analysis. This prepares the data to be used for MEME analysis in Fig 5E.

unique\_lncRNA\_500\_withfullairn\_markov is the bfile (Markov background model) for MEME analysis. It is generated by generate\_Markov\_model.sh

use wrapper\_meme\_classic\_allw.sh which calls meme\_classic\_allw.sh to run MEME analysis on the results from XAK\_chunkrscore\_organization.R.

parse\_MEME\_txt.py parse MEME restults .txt file to extract all motifs and their stats.