From what I remember, you were interested in finding ROHs in the krat genomes and then wanted to relate the ROHs to DMRs?

1. Easley\_ms\_scripts
   1. This uses nextflow to run the nf-core-methylseq pipeline and align to the reference genome. The nfcore-methylseq should pre-process raw data from FastQ inputs, align the reads and perform quality-control on the results.
   2. Questions:
      1. why are most things in the .sh scripts commented out?
      2. It looks like you left off at the snp calling part (step 03), right?
2. Easley\_plink\_rounds
   1. Plink is used to analyze genotype/phenotype data. Here, it is used to find ROHs in the sequences (and set size and window limits for ROHs of interest).
   2. Questions:
      1. It looks like there are 6 rounds of plink with the parameters changed in each round. Is there a “default” parameter set that is more well known?
      2. Do you remember much variation in results from changing the parameters?
      3. How do you decide which makes the most sense with the data?
3. Easley\_r\_scripts
   1. Intrapair\_gendist\_calcs.R
      1. This finds indv with and without offspring and looks at per-locus genotype similarities and distances within pairs. Output includes: intrapair\_gen\_distances.txt ; intrapair\_gen\_distances\_with\_shared\_windowIDs.txt ; pcaformatted\_gendists\_shared\_windowIDs.txt ;   
         with column names: c('pair.id','surv.or.no','c.index','wind.id','start','end','n.snps','n.diffs'). i.e., pair ID, fitness status, c.index, window ID, window start and end positions, # of SNPs in the window, and total # of allelic differences.
      2. Contigs are the DNA segments resulting from sequencing; these overlap to provide the whole genome. Because contig lengths and locations vary across individuals, this step is used to find overlapping contigs in pairs and then uses the variation in contigs to determine genetic distances (and therefore, relatedness).
      3. Questions:
         1. Is this doing pairwise comparisons between mated pairs or all sequenced individuals?
         2. How is this different from kinship coefficients, if at all?
            1. Note: doi: 10.5483/BMBRep.2013.46.6.177 suggests genetic distances and kinship coefficients are proportional
   2. Lo\_and\_hi\_roh\_region\_ID.R
      1. This \*randomly\* finds any genes that overlap the ROHs found by keeping track of the ROH region, the gene length(s), length(s) of overlap, and number of genes that overlapped. This includes genes that began outside of the ROH but ended inside, genes that completely overlap the ROH, genes that are completely inside the ROH, and genes that began inside and ended outside the ROH. Output is high\_roh\_randomization\_stats\_n.csv with column names c('iteration','total.gene.len','num.genes') and a pdf histogram titled “gene overlap with randomly selected regions (hi-ROH)” with “total length of overlapped genes” on the xaxis.
      2. This also looks at ROHs with no genes that overlap, but uses Easley because there are more regions to test here. Output includes: no\_roh\_randomregion\_geneoverlaps\_n.csv ; no\_roh\_randomization\_stats\_n.csv and a pdf histogram titled “gene overlap with no-ROH regions” with “total gene length” on the xaxis.
      3. Questions:
         1. Why the random regions rather than iterate over all contigs?
         2. Why run the “no gene overlap” section if no genes overlap?
4. Easley\_wgs\_scripts
   1. 01a\_read\_qc\_trimming.sh grabs the zipped fastq files, runs FastQC on the 4 read files per sample, runs TrimGalore to clean and trim adapters from the reads, then runs FastQC on the trimmed files.
      1. Questions:
         1. why are there 4 files per sample (mentioned in the script)?
         2. Why is there a 01b\_read\_qc\_trimming.sh and what is the difference between a and b (I don’t see any when skimming them)?
   2. 02a\_read\_mapping.sh maps the trimmed reads to the default (smaller) reference genome. Output are sorted .bam files and \*\_rgroups\_flagstat\_out.txt
      1. Q: difference between a and b here?
   3. 02c\_combining\_doubleseqs.sh grabs samples with multiple .bam files for an indv.
      1. Questions:
         1. I see some samples were from Duke and some from UofI – why?
   4. 03a\_preBQSR\_snp\_calling.sh marks duplicate reads (GATK tools ignores them so they aren’t being removed, just flagged).
   5. 03b\_noBQSR\_genomicsdb\_contig\_batches.sh – not sure what this does? I think it takes the GVCFs and puts genotype info there (cuz GVCFs contain all sites, even without variants called while VCFs only contain info). This also removes SNPs within 5bp of indels with BCFtools then holds only biallelic SNPS with SelectVariants.
   6. 03c\_vcf\_concat\_and\_filter.sh concatenates the final VCF files of SNPs and the output is krat\_final\_final\_allfiltcontigs\_all\_samps.recode.vcf.gz
   7. 04\_bcftoolsROH.sh generates allele frequencies for BCFtools and ROH calling with both genotype and genotype likelihood estimations
      1. More info here: <https://samtools.github.io/bcftools/howtos/roh-calling.html>
   8. There is no 05 step. Step 05\_plink runs plink with 6 different rounds (see above)
   9. 06\_LD\_pruning.sh makes a chromosome map, runs LD-pruning in Plink, identifies SNPs in strong LD, and creates a new .vcf file without SNPs in LD.
      1. Question:
         1. What does it mean that you “added in filtering for est-sfs to allow 0 missingness”?
   10. 07\_LD\_pruned\_plink.sh runs plink to find ROHs on pruned and trimmed sequences.
       1. Question: is this just the “chosen” plink round from Easley\_plink\_rounds 1-6?
   11. 08\_LD\_pruned\_bcftoolsROH.sh used BCFtools to find ROHs on the pruned and trimmed sequences.
       1. Question: steps 7 and 8 seem the same as 4 but before LD pruning. Why run step 4 before pruning at all when we typically don’t want SNPs in LD anyways?
   12. 09a\_R\_intrapair\_gendists.sh runs the EASLEY\_01\_intrapair\_gendist\_calcs.R script and copies the output to another folder and 09b\_R\_roh\_region\_analyses.sh runs the EASLEY\_02\_lo\_and\_hi\_roh\_region\_ID.R script and copies output files.
   13. 10a\_snpeff\_db\_build.sh uses the genome FASTA, GTF annotation, and protein FASTA files to go in SnpEff. [SnpEff](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3679285/) annotates and predicts the effects of SNPs and predicts coding effects.
   14. 10b\_snpeff\_annot.sh copies the full and LD-pruned VCF files and renames chromosomes to match RefSeq names, runs SnpEff, filters results to allow zero missingness, organizes the annotations with SnpSift, and compresses the VCFs (4 files).
   15. 11a\_prep\_gone\_vcf.sh grabs the 200 largest contigs to prepare the use of GONE (output is krat\_gone\_filt.recode.vcf). GONE is used to estimate effective population size in populations with overlapping generations. Output is krat\_gone\_filt\_integerchrnames.vcf.gz
   16. 11b\_run\_gone\_vcf has 4 separate .sh files – what are the differences? Looks like \_contig\_effects.sh iterated over 25, 50, and 100 contig sets but I’m not sure what that means? GONE is run 100 times in all scripts.
   17. 12a\_gerp\_input\_prep.sh – not sure what this does, looks like it just grabs the reference, aligns the sequences to the reference, then converts the .bam file to a .fasta file. 12b\_format\_mfa\_files.sh goes over the species files, adds the contig seq to the appropriate file, and renames the file. Output has 1 file per contig, 1 entry per species with data for that contig. 12c\_combine\_mfas\_run\_gerp.sh runs GERP. GERP identifies constrained loci in multiple sequence alignments by comparing the substitution level as compared to neutral expectation (because these sites are typically highly conserved)
       1. DOI: 10.1371/journal.pgen.1008827 suggests GERP score is related to the strength of purifying selection
       2. Questions:
          1. This seems to be a multi-species calculation, what are the species used here and can it be a within-species calc?
          2. Is the expected substitution rate related to the mutation rate or how is it defined?
          3. What is the significance of this step?
          4. When it says one file per contig, does it mean one file per section of DNA sequenced? Wouldn’t that be an insanely large number?
   18. 13\_plink\_het.sh uses plink to estimate heterozygosity.
       1. Q: over each indv? The whole pop?
   19. qualimap.sh runs Qualimap (in multi-sample BAM QC mode, providing raw BAM files), which evaluates the quality of next-generation seq data.
       1. Q: Is this where they get all the stats to report on the sequence itself? Like how much wasn’t aligned, depth, coverage, etc. and all that?
   20. vcf\_depth\_check.sh runs vcftools to find the mean depth for each indv.
   21. Additional scripts I’m not sure about but I think are just to get things in order?
       1. LDpruned\_vcf\_to\_genotypes.sh
       2. make\_simple\_contig\_name\_ref.sh
       3. vcf\_copy.sh
       4. vcf\_subset.sh
       5. vcfEffOnePerLine.pl
   22. local\_scripts directory looks like analysis and plotting figures, which I have not gone through yet.

Overall questions:

It looks like most of the scripts are commented out- why?

What does the \ at the end of most lines in the scripts signify?

How many of these data files are dangerous to download to my local drive (e.g., can wipe out what’s in storage)?

Is there a reason that I should run all of these scripts as well or would it make more sense to just use the data that you’ve already generated to save time and energy?

What is your overall opinion on the questions you wanted to answer and what you’ve found?