Methods

Phasing

* Used Whatshap 0.16 to phase previously identified *de novo* variants in 10 probands.
  + Input VCF file included complete trio
  + Output VCF file included complete trio, proband’s variants were phased
  + Did not include indels in phasing (did not use –indel flag)
* Programmatically assigned *de novo* variants to parent of origin
  + Generated GTF file from phased VCF using Whatshap in order to get haplotype blocks; this helped establish boundaries for which informative variants we were able to use to assign *de novo* variants to a parent of origin
  + *de novo* variants were assigned to a parent of origin if >=85% of the informative variants were assigned to that parent
* Manually phased indels using IGV
  + Used Illumina data to choose Pacbio reads that we were confident were either Ref or Alt reads
  + Highlighted reads and looked for informative SNPs on the same reads

Replicating Iceland Results

* Used Ref and Alt columns of dataframe to classify SNPs into mutational classes (C > A,

C > T, C > G, T > A, T > C, CpG > TpG) and included indels as their own mutational class

* In order to get CpG > TpG mutations:
  + Used bedtools getfasta with hg38 reference file to get the single nucleotides on either side of the SNP and marked every SNP that was a C adjacent to a G, or a G adjacent to a C as CpG
  + The *de novo* variants that were annotated as being in CpG regions and were C > T mutations classified as CpG > TpG, the *de novo* variants that were C > T but were not in CpG regions were classified as C > T (no overlap between the two)
* Calculated the fraction of phased *de novo* variants that were components of each mutational class and created bar chart to compare those assigned to mother versus father as parent of origin for each class
* Plotted these fractions for each proband against the age of the mother and age of the father to look for associations with parental age

Classifying Indels

* Developed Python package to automate this process
  + Named sorting\_hat
* Three classes: HR, CCC, non-CCC
  + HR: homopolymer run (mutation is in a region where there are 6 or more copies of the nucleotide being inserted or deleted)
  + CCC: change in copy count (the sequence being inserted or deleted has 1 or more repeats in the mutation region)
  + non-CCC: no change in copy count (the sequence being inserted or deleted is not repeated in the mutation region)
* Used bedtools getfasta to get bases surrounding the indel
  + If the indel was a single nucleotide, collected 6 bases on either side of the indel
  + If the indel was a sequence of 2 or more bases, collected 2\*length of sequence bases on either side of the indel
* Compared indel sequence to adjacent sequence of equal length
  + If the indel sequence and adjacent sequence were the same:
    - If the indel sequence was a single nucleotide, checked for 6 adjacent copies to see if it was HR
    - Otherwise, assigned as CCC
  + If the indel sequence and adjacent sequence were not the same, assigned as non-CCC
* Downloaded RepeatMasker from UCSC Genome Browser to obtain repeat name, repeat class, and family if the indel was in a repeat region