fastq.gz files
bash_variables
datatable.txt

00_start-pipeline.py

For each pool_name

sample *n*

trim each fastq file with `fastp`

01_trim-fastq.py

sample *n*

CodeAdapTree
Healthy Code for Future Research

sample *n*

- map each sample to ref.fa using `bwa-mem`
- add read group info while mapping
- reads filtered with `samtools`
- record flagstats with `samtools`
- mapping coordinates recorded with `bedtools`

02_bwa-map_view_sort_index_flagstat.py

- remove duplicates with `picardtools`
- record flagstats with `samtools`

03_mark_build.py

sample *n*

- use `GATK3`'s `RealignerTargetCreator` and `IndelRealigner` to realign around indels

04_realignTargetCreator.py
05_indelRealign.py

Once all sample.bam files have been religned …

- Call SNPs across all samples for a given interval.bed using varscan
- Convert each batch.vcf to batch.txt

batch_00.bed

start_varscan.py

batch_01.bed batch_02.bed batch_03.bed batch_04.bed batch_05.bed batch_06.bed batch_07.bed batch_08.bed batch_09.bed batch_10.bed batch_nn.bed

Once all varscan jobs have finished …

- Combine all batch.txt files and filter SNPs for GQ < 20, missing data > 25%, MAF
- Filter futher into files for INDELs, SNPs, SNPs in repeat regions, SNPs at putative paralog regions,

combine_varscan.py (filter_VariantsToTable.py)

Once combining has finished for each pool_name …

- Get read stats (counts of reads at each stage of pipeline: raw -> mapped)
- Bundle files for transfer to local server (create rsync cmds, mkdir cmds)

(run manually) 98_get_read_stats.py
(run manually) 99_bundle_files_for_transfer.py