Denghui Chen (UCSD, Palmer Lab) March 2021

California Mouse Genotyping Pipeline

Prerequisites

Cluster System

This pipeline is developed and tested on San Diego Supercomputer Center (SDSC) Triton Shared Computing Cluster (TSCC) that uses TORQUE Resource Manager (also known by its historical name Portable Batch System, or PBS) with the Maui Cluster Scheduler to define and manage job queues.

Compute Node Specifications

2 x 12-core Intel Haswell processors with 128GB of main memory.

Sequencing Data

96 California mice paired-end sequencing data

- iGenomX Riptide High-Throughput Rapid Library Prep Kit
- Illumina
- forward sequence length 131bp
- Reverse sequence length: 143bp
- coverage: ~0.3X

Reference Genome

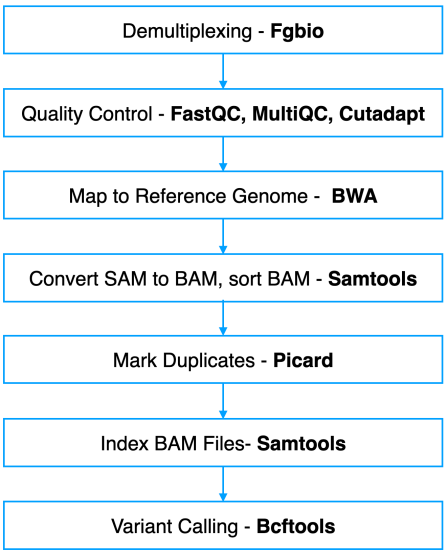
Reference genome used on alignment step: Peromyscus californicus insignis strain:IS Genome sequencing and assembly on NCBI: PRJNA513219

Github

The code for this California Mouse Genotyping Pipeline is fully documented on Github.

Genotyping Pipeline Documentation

This genotyping pipeline is specifically built for California Mice (Peromyscus californicus). The overview flow chart of the pipeline is shown below.



More detailed documentation on each step including command options and software versions used is shown below. [note: the time used shown is the time used to process all 96 samples in series, no parallel computing is used for this pipeline]

Reference Genome Preparation

Generate BWA index (BWA 0.7.12)

```
bwa index ref.fa
```

- BWA index command index database sequences in the FASTA format.

Time used: 00:40:00
Memory used: 5GB

Generate fasta file index (Samtools 1.3)

```
samtools faidx ref.fa
```

- Index reference sequence in the FASTA format.

Time used: 00:02:00
Memory used: 5GB

Generate the sequence dictionary (Picard 1.141)

```
java -jar picard.jar CreateSequenceDictionary \
      REFERENCE=ref.fa \
      OUTPUT=ref.dict
```

- Create a sequence dictionary for a reference sequence.

Time used: 00:02:00
Memory used: 5GB

Demultiplexing (Fgbio 1.1.0)

```
java -jar fgbio-1.1.0.jar DemuxFastqs \
    --inputs R1.fastq.gz R2.fastq.gz \
    --metadata metadata.csv \
    --read-structures 8B12M+T 8M+T \
    --output-type=Fastq \
    --threads 2\
    --output output_dir/ \
    --metrics output_dir/metrics.txt
```

- Perform sample demultiplexing on FASTQs.
- --metadata: a file containing the metadata about the samples (see here for more details).
- --read-structures: the read structure for each of the FASTQs (8B12M+T 8M+T is specifically for Riptide library preparation).
- --threads 2: the number of threads to use while demultiplexing.
- --metrics: the file to which per-barcode metrics are written.

Time used: 00:45:00
Memory used: 40GB

Quality Control

FastQC quality control check (FastQC 0.11.8)

```
fastqc file.fastq.gz --outdir=output_dir/
```

- Quality control check on raw sequence data.

Time used: 00:45:00
Memory used: 5GB

MultiQC quality control check (MultiQC 1.8)

```
Multiqc fastqc_dir -o output_dir
```

- Aggregate FastQC results across multiple samples into a single report.

Time used: 00:10:00
Memory used: 5GB

Quality Trimming (Cutadapt 1.9.1)

```
 cutadapt -q 20 \
      --minimum-length 50 -pair-filter=both \
      -o trimmed_R1.fastq.gz \
      -p trimmed_R2.fastq.gz \
       R1.fastq.gz R2.fastq.gz
```

- Perform quality trimming on FASTQs.
- -q 20: quality cutoff score on 3' end is set to 20.
- --minimum-length 50: discard processed reads that are shorter than 50.
- -pair-filter=both: apply filtering criteria to both reads in order for a read pair to be discarded.

Time used: 00:45:00
Memory used: 5GB

Map to Reference Genome (BWA 0.7.12)

```
bwa mem -aM -t 2\
    -R "@RG\tID:$flowcell_name.$lane\tLB:$library_id\tPL:ILLUMINA\tSM:$sample" \
    ref.fa trimed_R1.fastq.gz trimed_R2.fastq.gz > mapped.sam
```

- Map sequences against a reference genome.
- -a: output all found alignments for single-end or unpaired paired-end reads.
- -M: mark shorter split hits as secondary (for Picard compatibility).
- -t: number of threads.
- -R: complete read group header line.

Time used: 47:35:00
Memory used: 6GB

Convert SAM to BAM (Samtools 1.3)

```
samtools view -h -b -t ref.fa -o mapped.bam mapped.sam
```

- Convert SAM files to BAM files.
- -h: include the header in the output.
- -b: output in BAM format.
- -t: provide additional reference data.

Time used: 02:08:00
Memory used: 5GB

Sort BAM Files (Samtools 1.3)

```
samtools sort -m 10G -o sorted.bam mapped.bam
```

- Sort alignments by leftmost coordinates.
- -m: approximately the maximum required memory per thread.

Time used: 13:50:00
Memory used: 10GB

Mark Duplicates (Picard 1.141)

```
java -jar picard.jar MarkDuplicates \
    INPUT=sorted.bam \
    REMOVE_DUPLICATES=false \
    ASSUME_SORTED=true \
    METRICS_FILE=mkDup_metrics.txt \
    OUTPUT=mkDup.bam
```

- Mark duplicate reads.

Time used: 07:58:00
Memory used: 20GB

Index BAM Files (Samtools 1.3)

```
samtools index mkDup.bam mkDup.bai
```

- Index BAM file for fast random access.

Time used: 01:58:00

Memory used: 5GB

Variant Calling (Bcftools 1.9)

```
bcftools mpileup -q 20 -Q 20 --threads 23 \
  -f ref.fa  -O z mkDup.bam | \
bcftools call --no-version -m -v \
  --threads 23 -O z -o samtools.vcf.gz
```

bcftools mpileup

- Generate VCF or BCF containing genotype likelihoods for one or multiple alignment (BAM or CRAM) files.
- -q 20: minimum mapping quality for an alignment to be used.
- -Q 20: minimum base quality for a base to be considered.
- --threads 23: use multithreading with 23 worker threads.
- -f: the faidx-indexed reference file in the FASTA format.
- -O z: output compressed VCF.

bcftools call

- SNP/indel calling
- --no-version: do not append version and command line information to the output VCF header.
- -m: alternative model for multiallelic and rare-variant calling designed to overcome known limitations in -c calling model.
- -v: output variant sites only.
- --threads 23: use multithreading with 23 worker threads.
- -O z: output compressed VCF.

Time used: 54:24:00
Memory used: 30GB