Angsd analysis of first 20 Teosinte parents (Pal Marchico population)

Simon Renny-Byfield

November 17, 2014

This document is my record of the analysis performed on the first set of data avaialble for the ~70 or so teosinte plants from Pal Marchico, Mexico. The document is modelled on an earlier practice run and on some dummy HapMap2 data. The new data are from the Pal Marchico population and have recently been sequences and UC Berkely. Vince Buffalo has maped and sorted the reads using his paap.py pipeline and the .bam files ready for input into angsd. On the other hand the mapping parameters make the .bam files unsuitable for use with CNVer. This is because the mapping that is currently done has used paired end data and, does not have them interleaved as required. The authors of CNVer strongly recommend using bowtie to map the data, and suggest some parameters to use. This essentailly means that the read will have to be mapped twice, once using BWA-MEM (for GLs, SFS, pi and Tajima's D) and again with `bowtie` for input to `CNVer`.

Estimating the site frequency spectrum

First estimate the site allele frequency likelihood. This requires several things listed below:

1. A file with listed, one per line, all the .bam files you want to analyse. You can grab the files you need by cd'ing to the dir they are in and executing this code on the command line.

```
ls -d $PWD/*.bam > file.list.txt
```

2. Choose which method you want to use with:

```
-doSaf [int 1-4]
```

There are four options listed in detail here, in this case we want to estimate the **inbreeding co-efficient** of the sample, have this ready in a file and use the $-doSaf2$ option (See later for generating an inbreeding coefficient estiamtion).

3. Define your ancestral allele using the flag:

```
-anc <path/to/referencegenome>
```

In my case we do not know the ancestral allele state, which means instead of derived allele SFS we need a minor allele SFS (a folded

SFS). We can still provide an ancestral estimate using the reference genome (B73), but once folding is complete in becomes a minor allele SFS. We need to specify that we want a folded SFS with:

```
-fold 1
```

Or, alternatively we can estimate the ancestral state using *Tripsicum* reads mapped to the ref_v3 genome. We can supply a fasta file with tripsicum alleles placed on the ref_v3 resequence. The original file is stored here:

```
/group/jrigrp3/bottleneckProject/genomes/TRIP.fa
```

But I have a symbolic link in:

```
/home/sbyfield/teosinte_parents/genomes/TRIP.fa
```

4. Define the method for estimating Genotype Likelihoods:

```
-GL [int 1-4]
```

details of the different methods are provided here. This will be important later as we need the GLs to estimate the **inbreeding coefficient**.

5. Define the number of processors to use with:

```
-P [int]
```

6. Define the outfile name using:

```
-out <path/to/outfile>
```

there are several output files and a suffix will be added to each file.

Calculating the GL for each sample

The .bam files are not indexed and so I wrote a quick script to get these indexed:

```
#!/bin/bash -l
#OUTDIR=/home/sbyfield/teosinte_parents/angsd_output
#SBATCH -D /home/sbyfield/teosinte_parents/angsd_output
#SBATCH -o /home/sbyfield/teosinte_parents/logs/out_log-%j.txt
#SBATCH -e /home/sbyfield/teosinte_parents/logs/err_log-%j.txt
#SBATCH --array=1-20
#SBATCH --mem-per-cpu=8000
```

```
##Simon Renny-Byfield, UC Davis, November 17 2014
##Usage: sbatch -p queue <file.sh> <first.list>

echo "Starting Job:"
date

index=0
while read line; do
   file1[index]="$line"
   let "index++"
done < $1

#now index eaxh .bam file

samtools index ${file1[$SLURM_ARRAY_TASK_ID]}

echo "End Job: "
date
```

In order to calculate the inbreeding coefficient we need to know the genotype liklihoods of each sample. The genotype likelihood for each samples are calculated with: "'{options(width = 3)} #!/bin/bash #OUTDIR=/home/sbyfield/teosinte_parents/angsd_output #SBATCH -D /home/sbyfield/teosinte_parents/angsd_output #SBATCH -o /home/sbyfield/teosinte_parents/logs/out_log-%j.txt #SBATCH -e /home/sbyfield/teosinte_parents/logs/err_log-%j.txt
echo "Starting Job:" date
COMMAND="angsd -bam /home/sbyfield/teosinte_parents/file.list.txt -doGlf 3 -GL 1 -out teo_parents20 -doMaf 2 -SNP_pval 1e-6 -doMajorMinor 1 -nThreads 16 -r 10" echo $COMMAND
angsd -bam /home/sbyfield/teosinte_parents/file.list.txt -doGlf 3 -GL 1 -out teo_parents20 -doMaf 2 -SNP_pval 1e-6 -doMajorMinor 1 -nThreads 16 -r 10
echo "Ending Job:" date

```
Note that in this case the $-GL\space1$ parameter means that the GLs are calculated using the SAMtools alg

There is an [examples](https://github.com/fgvieira/ngsF/tree/master/examples) folder in the 'ngsF' github
```

!/bin/bash

OUTDIR=/home/sbyfield/teosinte_parents/angsd_output

SBATCH -D /home/sbyfield/teosinte_parents/angsd_output

SBATCH -o /home/sbyfield/teosinte_parents/logs/out_log-%j.txt

SBATCH -e /home/sbyfield/teosinte_parents/logs/err_log-%j.txt

N_SITES=$((zcat teo_parents20.glf.gz | wc -l-1))
zcat teo_parents20.glf.gz | ../ngsF -n_ind 20 -n_sites $N_SITES
-glf - -min_epsilon 0.001 -out teo_parents20.approx_indF -approx_EM
-seed 12345 -init_values r zcat teo_parents20.glf.gz | ../ngsF -n_ind
20 -n_sites $N_SITES -glf - -min_epsilon 0.001 -out teo_parents20.indF
-init_values teo_parents20.approx_indF.pars
‴