Analysis of Next-Generation Sequencing Data (ANGSD) basic operations

CGRL 10/6/2014
ANGSD Wiki: http://popgen.dk/angsd/index.php/Main_Page#Overview

I. Preparing input

ANGSD can take various types of input files including:
1. BAM files
2. Genotype likelihood files
3. Beagle files

Everything that follows will assume BAM as input.
Terminal commands will be proceeded by a '$' for the remainder of this manual.

A. Index BAM files

$ cd /home/ke/Desktop/CGRL_SNP/pop2_BAM/

This directory contains BAM files for 8 individuals from “population 2”. Now we will index these BAM files using SAMtools faidx:

$ for i in $(seq 1 8); do samtools index ind$i.bam; done

Note that we also need a list of the BAM files. You can see what this file looks like:

$ cat /home/ke/Desktop/CGRL_SNP/pop2_BAM.list

B. Index fasta files

Many of the analyses that we will do require an indexed reference fasta file. In addition, if we want to use the unfolded site frequency spectrum (SFS) for analyses we need to provide an indexed outgroup fasta file to polarize SNPs.

$ cd /home/ke/Desktop/CGRL_SNP/reference/

Create index file for reference:

$ samtools faidx ref.fasta

create index file for outgroup:

$ samtools faidx outgroup.fasta

A .fai (fasta index) file should now exist for the reference and outgroup.

II. Run ANGSD (main)

A. We are going to combine various commands into one ANGSD run:

$ cd ~/Desktop/CGRL_SNP/

$ angsd -bam ./pop2_bam.list -GL 1 -doMaf 2 -doMajorMinor 1 -doGeno 9 -doPost 1 -doSaf 1 -anc ./reference/outgroup.fasta -baq 1 -C 50 -ref ./reference/ref.fasta -rf clean_sites.rf -out ./output/pop2

arguments:

-bam : list of BAM files

-GL : method for calculating genotype likelihoods
1 : SAMtools model with 10 genotype likelihoods per site
2 : GATK model
3 : SOAPsnp model
4 : SYK model

-doMaf : estimate minor allele frequencies
1 : known major and minor allele, based on EM algorithm with genotype likelihoods
2 : known major and unknown minor allele, sums over 3 possible minor alleles weighted by their probabilities, based on EM algorithm with genotype likelihoods
4 : allele frequencies estimated directly from genotype posterior probabilities
8 : allele frequencies based directly on allele counts

-doMajorMinor : method for identifying the major and minor allele
1 : infer major and minor from GL
2 : infer major and minor from allele counts
3 : use a file with specified major and minor allele (requires -sites)
4 : use reference allele as major (requires -ref)
5 : use ancestral allele as major (requires -anc)

-doGeno : call genotypes (can be summed to combine output types)
1 : print major and minor allele
2 : print called genotype as -1, 0, 1, 2
4 : print called genotype as AA, AC, AG, …
8 : print posterior probability for [major, major], [major, minor], [minor, minor]
16 : print posterior probability of called genotype
32 : binary file of posterior probabilities for [major, major], [major, minor], [minor, minor]

9 = 1 + 8 = print major and minor allele AND posterior probability for [major, major], [major, minor], [minor, minor]

-doPost : estimate posterior genotype probability
1 : based on allele frequency as a prior
2 : based on a uniform prior

-doSaf : estimate allele frequency likelihoods for each site
1 : based on individual genotype likelihoods, assumes HWE
2 : based on individual genotype likelihoods, accounts for inbreeding (requires file of inbreeding coefficients)
3 : calculates the posterior probability of the sample allele frequency distribution for each site. Requires a prior SFS distribution (can be obtained with -doSaf 1 followed by -realSFS)
4 : calculate the per-site posterior probabilities of the sample allele frequency distribution based on genotype probabilities

-anc : fasta file of ancestral sequences (can provide outgroup with this option)

-ref : fasta file of reference sequences

-baq : adjust quality scores around indels (as in SAMtools), requires -ref
0 : disabled
1 : enabled

-C : adjust map quality in areas of excessive mismatches (as in SAMtools), requires -ref

-rf : a region file that specifies particular regions for analyses (allows omission of sites or regions that failed quality control filtering)

example contents of a region file (-rf):

$ less clean_sites.rf

Contig41:63-75
Contig240:202
Contig742:112
Contig742:133-143
Contig742:145-162
Contig742:164-347
Contig742:262
Contig1008:194
.
.
.

-out : name and destination of output file

There are also some other useful options that we did not use, notably:

-minMapQ : discard reads with mapping quality below this value

-minQ : discard read bases with quality below this value (default is 13)

-only_proper_pairs : only use reads when the mate can be mapped (default is 1)
0 : disabled
1 : enabled

-uniqueOnly : discard reads that do not map uniquely

-remove_bads : discard 'bad' reads (flag >= 255) (default is 1)
0 : disabled
1: enabled

Also, we are working with the **unfolded site frequency spectrum (SFS)** which has 2*n+1 categories, where n is the diploid sample size. If you aren't able to reliably polarize SNPs, you could use the **folded SFS** which has n+1 categories. You specify that you want to use the folded SFS with the argument **-fold 1** when running ANGSD. This SFS will be based on the major and minor allele instead of the ancestral and derived allele.

B . Let's look at the output

$ cd output

$ gunzip -c pop2.geno.gz | less -S

You should see this:

```
Contig41     73    T    A    1.000000    0.000000    0.000000    ….....
Contig41     74    T    A    1.000000    0.000000    0.000000    ….....
Contig41     75    T    A    1.000000    0.000000    0.000000    ….....
Contig240    202   A    C    1.000000    0.000000    0.000000    ….....
Contig742    112   A    G    0.000000    1.000000    0.000000    ….....
.
.
.
```

file breakdown by column:
1 – chromosome/contig
2 – site
3 – major allele
4 – minor allele
5 – individual 1's probability of [major, major], i.e. TT
6 – individual 1's probability of [major, minor]. i.e. TA
7 – individual 1's probability of [minor, minor]. i.e. AA
For each site, columns 5-7 are repeated for every remaining individual (for individual 2 .. n)

$ gunzip -c pop2.mafs.gz | less -S

You should see this:

```
chromo  position    major  minor  ref   anc   unknownEM     nInd
Contig41     63     C      A      C     C     0.000001      8
Contig41     64     A      C      A     A     0.000001      8
Contig41     65     T      A      T     T     0.000001      8
Contig41     66     G      A      G     G     0.000000      8
```

.
.
.

So for each site, you get the identity of the major and minor allele, as well as the reference base (if -ref is supplied) and ancestral base (if -anc is supplied). The **minor allele frequency** is shown in the "unknownEM" column. The name of this column may differ depending on what -doMaf option you use. Lastly, the number of individuals for which there was data for this site is shown in the "nInd" column.

note: by using the -minMaf option when running ANGSD, you can filter out sites with MAF below a certain frequency.

C. A more direct way to call SNPs and genotypes

$ cd ..

$ angsd -bam pop2_bam.list -GL 1 -doMaf 2 -SNP_pval 1e-4 -doMajorMinor 1 -doGeno 21 -doPost 1 -postCutOff 0.95 -baq 1 -C 50 -ref ./reference/ref.fasta -rf clean_sites.rf -out ./output/pop2_callgeno

- SNP_pval : only print sites with a p-value of being variable (determined from a likelihood ratio test) below the specified value

-doGeno 21 = 1 + 4 + 16 = print major and minor allele AND print the called genotype AND print the posterior probability for the called genotype

D. Let's look at the new output

$ cd output

$ gunzip -c pop2_callgeno.geno.gz | less -S

You should see:

Contig742	112	A	G	AG	1.000000	AA	0.999972	AA	0.999986 ...
Contig742	262	T	A	TT	1.000000	TT	1.000000	TT	0.999984 ...
Contig742	262	T	A	TT	1.000000	TT	1.000000	TT	0.999984 ...
Contig1061	559	C	G	CC	1.000000	CG	0.999999	CC	1.000000 ...
Contig3464	138	C	G	CC	0.999006	CC	0.999876	CC	0.999984 ...
Contig4577	353	C	T	CT	1.000000	CC	0.999969	TT	0.999024 ...
Contig5187	231	A	T	AA	1.000000	AA	1.000000	AA	0.999996 ...

.
.
.

file breakdown by column:
1 – chromosome or contig
2 – site
3 – major allele
4 – minor allele

5 – individual 1's called genotype
6 – individual 1's called genotype posterior probability
For each site, columns 5 and 6 are repeated for every remaining individual (for individuals 2 .. n)

Note that you don't see every site that is in your -rf file. This is because you supplied -SNP_pval, which will cause only sites with a p-value of being variable that is <= to the value provided to -SNP_pval to be printed.

$ gunzip -c pop2_callgeno.mafs.gz | less -S

You should see:

chromo	position	major	minor	ref	unknownEM	pu-EM	nInd
Contig742	112	A	G	A	0.187501	0.000000e+00	8
Contig742	262	T	A	T	0.062509	2.220446e-16	8
Contig742	262	T	A	T	0.062509	2.220446e-16	8
Contig1061	559	C	G	C	0.125000	0.000000e+00	8
Contig3464	138	C	G	C	0.112971	4.323329e-07	8
Contig4577	353	C	T	C	0.499914	0.000000e+00	8

.
.
.

In addition to the information from last time, we also get the probability that each site is variable under the “pu-EM” column, which should be <= -SNP_pval. Note that this time we don't have the identity of the ancestral allele because we did not supply -anc when we ran ANGSD.

III. Estimate the site frequency spectrum (SFS)

A. Calculate the sample SFS

We will use realSFS to obtain a maximum likelihood estimate of the sample SFS from the per-site allele frequency likelihoods.

The arguments to realSFS are:
realSFS [.saf file from -doSaf] [haploid sample size, i.e. the number of chromosomes in your sample]

Note that you can also supply some optional arguments:

-maxIter : maximum number of iterations for EM algorithm

-P : number of threads to use

Also note that it prints to stdout, so we need to redirect the output to where we want it. So let's run realSFS:

$ realSFS pop2.saf 16 > pop2.sfs

B. Let's look at the sample SFS

$ less pop2.sfs

You should see:

-0.361666 -2.740313 -3.200050 -3.471989 -3.750800 -3.749112 -4.503681 -4.096752 -4.209531 -4.795430 -4.845611 -5.138961 -5.360786 -5.349834 -5.048887 -5.775745 -3.308375

There are 2*n+1 values (for the unfolded SFS), which in our case with n = 8 diploid individuals is 17 values. These values are natural log-transformed probabilities of sampling a site in it's respective SFS category:

column 1 := probability of sampling zero derived alleles (natural log scale)
column 2 := probability of sampling 1 derived allele (natural log scale)
column 3 := probability of sampling 2 derived alleles (natural log scale)
column 4 := probability of sampling 3 derived alleles (natural log scale)
.
.
.
column 17 := probability of sampling 16 derived alleles (natural log scale)

C. Plot the SFS

Lets visualize the SFS using R:

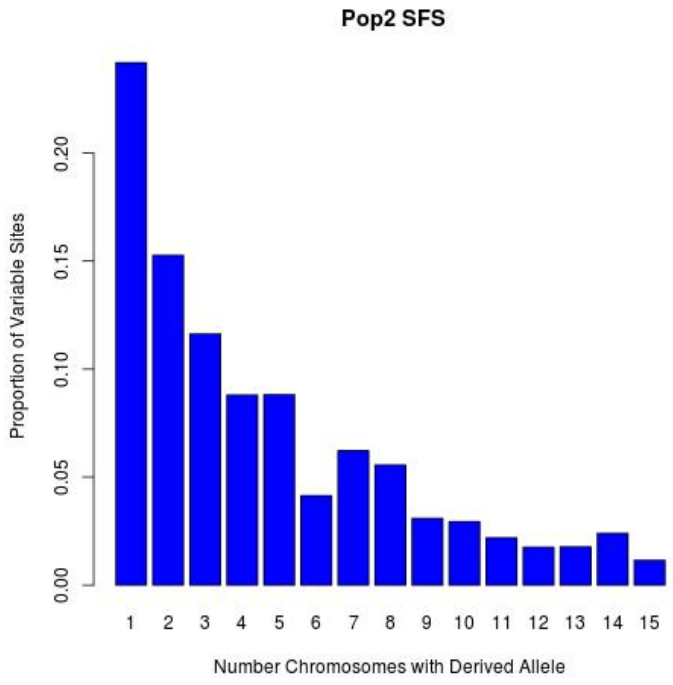
$ R

$ norm <- function(x) x/sum(x) # function to normalize

$ sfs <- exp(scan("pop2.sfs")) # read in the .sfs file and back-transform from natural log scale

$ sfs <- norm(sfs[-c(1,length(sfs))]) # isolate variable categories (1 .. 2n -1) and normalize

$ barplot(sfs, xlab="Number Chromosomes with Derived Allele", names=1:length(sfs), ylab="Proportion of Variable Sites", main=" Pop2 SFS",col='blue') # plot the SFS (for variable categories only)



$ q() # exit R
$ n

D. Calculate allele frequency posterior probability distribution

We will now use the allele frequency probability distribution specified in the .sfs file as a prior on the per-site allele frequency likelihoods specified in the .saf file to calculate the per-site allele frequency posterior probabilities.

$ angsd -bam ../pop2_bam.list -GL 1 -doSaf 1 -pest pop2.sfs -anc ../reference/outgroup.fasta -baq 1 -C 50 -ref ../reference/ref.fasta -rf ../clean_sites.rf -out pop2_post

This will dump a binary file that has the same format as the .saf file (also binary) except that the likelihoods are now posterior probabilities of the allele frequency for each site. Let's see what these binary files actually look like:

$ saf2txt -saf pop2_post.saf -nind 8 -fold 0 > pop2_post_saf.txt

-saf : .saf file from -doSaf

-nind : number of diploid individuals in sample

-fold : whether or not the folded SFS is used
0 : using the unfolded SFS
1 : using the folded SFS

E. Let's look at the allele frequency posterior probabilities

$ less pop2_post_saf.txt
(or to wrap the output: $ less -S pop2_post_saf.txt)

You should see:

```
-0.0108413 -4.5935978 -7.3798281 -10.1186835 -13.0351504 -15.8916940
-19.8127336 -23.0621201 -27.8776818 -43.5111045 -63.5723549
-86.9989691 -111.1813171 -135.9201102 -163.3122749 -192.7560639
-235.4594762
-0.0108413 -4.5935978 -7.3798281 -10.1186835 -13.0351504 -15.8916940
-19.8127336 -23.0621201 -27.8776818 -43.5112024 -63.5901386
-86.7284106 -110.6860971 -135.3640490 -162.9667534 -192.6133117
-234.2392986
-0.0108413 -4.5935978 -7.3798281 -10.1186835 -13.0351504 -15.8916939
-19.8127336 -23.0621200 -27.8776814 -43.4938057 -62.1483036
-85.5117339 -109.6858321 -134.4986008 -162.3007759 -192.0648737
-233.4863031
.
.
.
```

In this file each row (shown here by a different color) corresponds to a different site, while each column corresponds to the derived allele frequency (for our unfolded SFS case). The probabilities are in natural log space. So in this file,

column 1 := probability of sampling zero derived alleles (natural log scale)
column 2 := probability of sampling 1 derived allele (natural log scale)
column 3 := probability of sampling 2 derived alleles (natural log scale)
column 4 := probability of sampling 3 derived alleles (natural log scale)
.
.
.
column 17 := probability of sampling 16 derived alleles (natural log scale)

How do you know the identity of each site in the .saf files?
Look at the .saf.pos.gz file that gets dumped when you run -doSaf

$ gunzip -c pop2.saf.pos.gz | less

You should see:

```
Contig41 63
Contig41 64
Contig41 65
Contig41 66
Contig41 67
Contig41 68
Contig41 69
Contig41 70
```

```
Contig41	71
Contig41	72
Contig41	73
Contig41	74
Contig41	75
Contig240	202
Contig742	112
Contig742	133
.
.
.
```

The chromosome/contig name is listed in column 1 and the site number is listed in column 2. The row order of the .saf.pos.gz file (the order of sites) is the same as in the .saf file, so row X in the .saf file corresponds to row X in the .saf.pos.gz file. Therefore, you can simply match up the .saf.pos.gz and .saf files.

F. Calculate 2D-SFS

Now we will focus on the case in which you have two populations, and you would like to know the *joint* probability of sampling X derived alleles in population 1 AND Y derived alleles in population 2. This joint site frequency spectrum is sometimes referred to as the 2D-SFS, and in the unfolded case is represented by a 2*m+1 X 2*n+1 matrix (where m = number of diploid individuals in population 1 and n = number of diploid individuals in population 2).

To calculate the 2D-SFS use:
realSFS 2dsfs [.saf file for pop1] [.saf file for pop2] [haploid number for pop1 sample] [haploid number for pop2 sample]

(recall that by “haploid number”, I am referring to the number of chromosomes in your sample)

Again you can optionally specify the number of threads that you would like to use with the -P argument.

In order to calculate the 2D-SFS, you must use sites that are common to both the population 1 and population 2 sample. In other words, when generating the .saf files for population 1 and population 2, you should supply the same region file with the -rf argument.

So now let's calculate the 2D-SFS (in this example, the sample sizes, m and n, are both 8 individuals):

```
$ realSFS 2dsfs pop1.saf pop2.saf 16 16 > pop1_pop2_2dsfs.sfs
```

G. Let's look at the 2D-SFS

```
$ less pop1_pop2_2dsfs.sfs
```

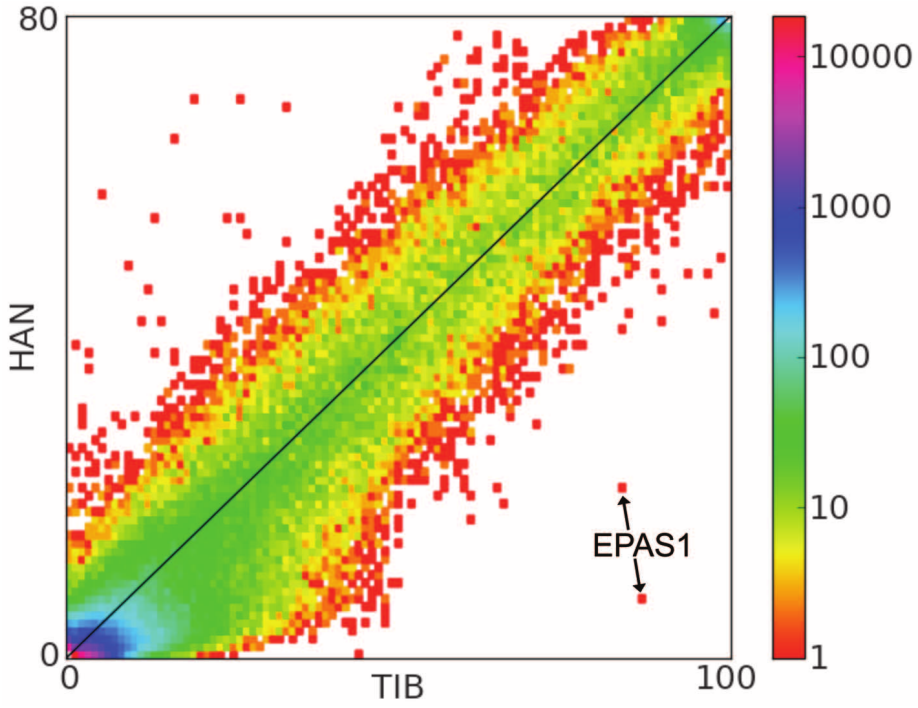
You should see something like this (I am only showing the first 4 rows and 4 columns of the 17 X 17 matrix):

-0.413536 -3.952256 -5.423965 -6.718340 ….
-3.583105 -3.532157 -4.298256 -5.236740 ….
-5.612870 -4.245219 -4.455337 -5.009263 ….
-8.274651 -5.910421 -5.050516 -4.603654 ….
.
.
.
.

Each element in this matrix is the natrual log-transformed probability of sampling X derived alleles from population 1 and Y derived alleles from population 2, where X and Y correspond to the row and column categories, respectively.

This type of analysis is useful for identifying differences in allele frequencies between two populations, and can be visualed using a heatmap:

Sequencing of 50 Human Exomes Reveals Adaptation to High Altitude
Xin Yi *et al.*
Science **329**, 75 (2010);
DOI: 10.1126/science.1190371



IV. Estimating Thetas

A. Estimate Thetas using an Emprical Bayes method

$ angsd -bam ../pop2_bam.list -doThetas 1 -doSaf 1 -pest pop2.sfs -GL 1 -anc ../reference/outgroup.fasta -out pop2

-doThetas : calculates various thetas for each site

This will dump a .thetas.gz file

B. Examine the .thetas.gz file

$ gunzip -c pop2.thetas.gz | less

You should see:

```
#Chromo Pos     Watterson       Pairwise        thetaSingleton  thetaH  thetaL
Contig41        6       -2.945320       -3.071423       -2.703933       -4.033557       -3.440983
Contig41        7       -3.031549       -3.199511       -2.722989       -4.261354       -3.595656
Contig41        8       -3.078859       -3.272205       -2.732913       -4.402018       -3.685379
Contig41        9       -3.278009       -3.574270       -2.789623       -4.964315       -4.045023
Contig41        10      -3.278009       -3.574270       -2.789623       -4.964315       -4.045023
Contig41        11      -3.278019       -3.574291       -2.789622       -4.964378       -4.045052
Contig41        12      -3.278016       -3.574285       -2.789622       -4.964361       -4.045044
Contig41        13      -3.278019       -3.574291       -2.789622       -4.964378       -4.045052
.
.
.
```

.thetas.gz file column breakdown:
1 – chromosome/contig name
2 – site number
3 – Watterson's Theta (natural log scale)
4 – Tajima's Theta or the average number of pairwise differences (natural log scale)
5 – Singleton Theta (natural log scale)
6 – Theta H (natural log scale)
7 – Theta L (natural log scale)

C. Estimate theta for a region

Linear theta estimates, such as Watterson's Theta and Tajima's Theta, can be summed over a region (e.g. a contig) to give a region-wide estimate of theta.

V. Ending Remarks

This manual provides a brief overview of some of the basic functions of ANGSD. There are also methods for performing population genetic analyses implemented in ANGSD. Additionally, the software ngsTools (paper: http://www.ncbi.nlm.nih.gov/pubmed/24458950 download: https://github.com/mfumagalli/ngsTools) is designed to interface with ANGSD for performing additional population genetic analyses.