INTORUCTION TO DILS INPUT

DILS requires a simple input fasta, containing for each individual the sequence of both haplotypes for each locus. Since DILS does not utilize haplotype information, which allele is reported for each SNP on each haplotype does not matter. i.e. the two sequences that are provided for each locus and individual can be pseudo-haplotypes obtained assigning randomly an allele for each variant to each of the two haplotype.

DILS input format is as follows

Header: ">LocusName|pop|individual|allele1" and ">LocusName|pop|individual|allele2"

```
>Locus0001|KAT|KAT-10|allele_1
CTCATGAAAGTCACCATGAGTCTGTTTGAATNCTGTTGACAGCCTGTAGCAGGAGAAGCAGCTG
>Locus0001|KAT|KAT-10|allele_2
CTCATGAAAGTCACCATGAGTCTGTTTGAATCACTGTTGACAGCCTGTAGCAGGAGAAGCAGCTG
>Locus0001|AS|AS_01|allele_1
TGATATTGAAGTGATTATTATTAGTTAGATTACACTCCATTAAAACACGATGTTCTGTATCGCGGACAT
>Locus0001|AS|AS_01|allele_2
TGATATTGAAGTGATTATTATTAGTTAGATTACACTCCATTAAAACACGATGTTCTGTATCGCGGACATACAGCCTTTCCTGTC
```

This is a small example (two individuals, one locus), DILS require hundreds to thousands of loci, each independent (i.e. not in physical linkage with others) and with a length of a few hundreds to a few thousands bp. It is not straightforward to generate such input from a VCF, as we need to have full sequence data, and correctly report data with low quality (e.g. poor mapping, base quality and low read depth) with Ns. So we need to work from BAMS, not VCFs. The following pipeline generates input for DILS starting from:

- Mapped bam files (ideally, these should be realigned around indels)
- An indexed reference genome
- A population map in tab delimited format with three columns, 1) Population name 2) Individual names and 3) path to the corresponding bam file.

The scripts needed are:
1**-Generate_Mappability_Mask.sh (Generate_mapmask.py in same folder)**
2-Generate_loci.sh
3-BAMS2DILS.sh
Check_fasta.sh
You will also need the tools from SNPable
https://lh3lh3.users.sourceforge.net/snpable.shtml), **get them with the following commands (not for user of my lab, they are already on the workstation)**
wget *https://lh3lh3.users.sourceforge.net/download/seqbility-20091110.tar.bz2*
tar -xvjf seqbility-20091110.tar.bz2

STEP 1: GENERATE A SET OF LOCI THAT WILL BE USED TO GENERATE THE PSEUDO HAPLOTYPES FOR EACH INDIVIDUAL.

PART A : we use Heng Li's SNPable approach (see link) to generate a bed file containing the coordinates **of all mappable regions of the genome**. This approach splits the genome into **fragments of a certain length** (which should match the read length of our data, for modern WGS about 150 bp) and **remaps them to the genome, providing the coordinates of regions of the genome where we can reliably map data**. This will effectively remove repetitive regions (including paralogs), and it's important since we cannot do much filtering for that later on.
Use the "**1-GenerateMappabilityMask.sh**" script to do so. The final output is a "**mappable_regions.bed**" file including all regions with high **mappability**. Make sure the "**Generate_mapmask.py**" is in the same folder as the script.

The script can be run as follows
bash 1-Generate_Mappability_Mask.sh -genome genome.fa -dir output_dir -readlen 150 -seqbility /path/to/seqbility

-genome: path to your genome fasta, **needs to be indexed with bwa** (e.g. bwa index genome.fasta)
-dir: output directory (choose something like **mappability**)
-readlen: length of the reads to be mapped. E.g. 36 bp for 2bRAD, 150 bp for WGS with 150bp reads etc… Needs to match your read length.
-seqbility: *the path where the SNPpable tools have been download* *https://lh3lh3.users.sourceforge.net/download/seqbility-20091110.tar.bz2* **. In BRUCE, this is** ***/home/marevo/Softwares/seqbility-20091110***

Once you have a bed file of the list of mappable regions use the script **calculate_mappable_lengths.sh** to check the **average length and total length** of your mappable regions. It's important you choose, in the next step, a locus length that is lower than the average length of your mappable regions.

e.g *bash calculate_mappable_lengths.sh mappable_regions.bed*

This outputs a summary like this:
Regions: 50251
Total length: 550960869
Average length: 10964.2

Part B: We now generate the coordinates for all loci which will be used in DILS. Using the "**2-Generate_loci.sh**" script. The script requires only standard UNIX tools (awk, sort, wc) and can be run like this:
bash 2-Generate_loci.sh <mappable_regions.bed> <window_size> <num_loci>

E.g.: bash 2-Generate_loci.sh mappable_regions.bed 2000 5000
This will generate a maximum of 5000 coordinates of 2kb loci within the mappable space. The script optimizes spacing based on the total mappable space and the number of requested loci, so that they are as evenly spread out as possible.

This script generates a specified number of loci (genomic windows) that are:

- **Evenly spaced across the total mappable genome**
- **Fully contained within high-quality (mappable) regions**
- **Non-overlapping** and **uniform in size**

Input:

- A BED file of **mappable regions**
- Desired **window size** (e.g. 5000 bp)
- Desired **number of loci** (e.g. 2000)

Steps: **1)** Sorts the input BED file by chromosome and position; **2)** Calculates the **total mappable length** (sum of all BED intervals); **3)** Computes an **ideal spacing** between loci: spacing = total_mappable_length / number_of_loci; **4)** Virtually walks along the mappable regions linearly, placing a window every spacing bp **5**) Ensures each window fits entirely inside a mappable region **5)** Stops once the target number of loci is reached.

NOTE: the final number of loci will usually be less than the specified number (e.g. if you select 5000 loci, you will end up with 4500-4900 loci)

Output: a BED file with **up to N windows**. File name includes: Number of loci, Window size (in kb), Distance between windows (in kb).

Example:

Command: bash 2-Generate_loci.sh /mappable_regions_autosomes.bed 2000 5000

Output file: LOCI_N4644_len2kb_mindist110kb_mappable.bed

Here only 4644 Loci were recovered, even if the set number was 5000.

STEP 2: GENERATE THE PSEUDOHAPLOTYPES FOR EACH LOCUS AND EACH INDIVIDUAL.

This script generates a fasta for each locus containing, for each individual, both pseudo-haplotypes. It requires several ***samtools*** and ***bcftools*** installed. It can be run with the following command:

bash 3-BAMS2DILS.sh -genome <genome.fa> -loci <loci.bed> -popmap <pop_map.txt> -out <output_dir> [-minDP 5] [-mapQ 20] [-baseQ 20]"

-genome: in fasta format and **indexed with samtools**.

-loci: in bed format

-popmap: tab delimited file with three colums: population ID, individual name, path to bam file.
e.g

POP1 IND1 /path/to/IND1.bam
POP1 IND2 /path/to/IND2.bam
POP2 IND3 /path/to/IND3.bam
POP3 IND4 /path/to/IND4.bam

It's essential this is in the correct format as it will be used to generate the fasta headers for DILS (e.g >locusX|POP|IND|allele_1)

-out: the output directory for the fasta files. **NOTE: if there is an old analysis with the same output directory, it will append results to existing fasta files. Make sure to delete the folder before re-running with same output directory**

-minDP, -mapQ, -baseQ: minimum read depth, mapping quality and base quality to call a base (otherwise it will **appear as N** in the final fasta output). Defaults are, respectively: **5, 20, 20**. These will depend largely on your sample, but avoid going below the defaults and consider increasing -minDP.

Example for calling the script:

```
bash 3-BAMS2DILS.sh \
-genome /media/A/FLOUNDERS/REF_PFLE/ena_PRJEB60141_sequence.fasta \
-loci Loci_10.bed \
-popmap Pop_map_AD_BLACK.txt \
-out dils_loci \
-minDP 8 \
```

-mapQ 25 \
-baseQ 30

Now what does this script actually do? (read if interested, else skip to STEP 3)

This script extracts pseudo-haplotype sequences for each individual at a set of genomic loci. It prepares input files for **DILS** (a demographic inference tool) by:

- Calling SNPs from BAM files
- Applying quality and depth filters
- Constructing diploid sequences (2 haplotypes per individual)
- Outputting a FASTA file per locus

Loop over loci

For each locus (line) in the BED file:

- Extract chromosome, start, and end
- Define the region (e.g. chr1:1000-3000)
- Create a temporary FASTA reference sequence for this region

Loop over individuals

For each individual in the Pop_map_BAM.txt:

- Retrieve BAM file path, population, and sample name

Process each sample at the current locus

For each sample:

1. **Compute depth per position** using *samtools mpileup*
   → Output: per-base depth file
2. **Call variants** using *bcftools mpileup* + *bcftools call*
   - Filters to biallelic SNPs only
   - Excludes indels and multi-allelic sites

   These steps are essential to ensure perfect alignment is maintained.

3. **Extract genotype and depth per SNP**
   → Output: a .tsv table of position, REF, ALT, GT, DP
4. **Build haplotype sequences**

Using awk:

- Loads the reference sequence
- Modify it based on SNP genotypes:
    - 0/0 → homozygous reference
    - 1/1 → homozygous alternate
    - 0/1 or 1/0 → heterozygous → one allele gets REF, the other ALT, randomly (**they are pseudo-haplotypes, there is no real phasing**)
    - **Low coverage or missing → N** (determined by -minDP, -baseQ, -mapQ values)
- Output both alleles as FASTA entries:
- >Locus0001|POP|SAMPLE|allele_1
- ACTG…
- >Locus0001|POP|SAMPLE|allele_2
- ACTN…

Clean up temporary files

Deletes intermediate files like:

- VCF
- Depth files
- Temporary reference

Final Output

The script cleans up all intermediate files and generates one FASTA file for each locus, each of which contains 2 sequences per individual.

dils_out/Locus0001.fasta

dils_out/Locus0002.fasta

STEP 3: CHECK THE FASTA FILES OUTPUT AND CONCATENATED THEM TO USE FOR DILS

Now that we have generated one fasta file for each locus, we can concatenate them in a single fasta file to use for DILS with a simple command:

*cat dils_out/Locus*fasta > DILS_input.fasta*

However, **before doing that you might want to first check the fasta files are all good**. Each fasta file should have **the same exact length**, as defined when creating the loci, **should have the same number of sequences** (2X number of individuals) and all **individuals should have the same length**. The script *Check_fasta.sh* generates a table containing these information for all loci.

Example:

bash Check_fasta ./dils_loci > Loci_check.tab

cat Loci_check.atb

File	Num_Seqs	Length(s)	Uniform
Locus0001.fasta	4	2000	YES
Locus0002.fasta	4	2000	YES
Locus0003.fasta	4	2000	YES
Locus0004.fasta	4	2000	YES
Locus0005.fasta	4	2000	YES
Locus0006.fasta	4	2000	YES
Locus0007.fasta	4	2000	YES
Locus0008.fasta	4	2000	YES
Locus0009.fasta	4	2000	YES
Locus0010.fasta	4	2000	YES

.....

NumSeqs should be the same for all loci, under length there should be only one number (the length you chose). Remove any locus that does not conform (not uniform).

One you have done this check you know you have fasta files perfectly aligned and in the correct format for DILS. Concatenated them and happy ABC!