Short Introduction

Mappability mask file gives all regions on the chromosome, on which short sequencing reads can be uniquely mapped. Heng Li's software SNPable can be used for this purpose. It first chopped the whole reference genome into K length short reads, which cover the whole genome with K depth, and then mapped these reads back to the reference genome. With the stringency r, regions which are mapped uniquely are kept and otherwise are masked out. I used this mask file in the msmc analysis.

workflow

In my example, I set k as 100 (the same length as my short read data) and r as 0.5. First split the genome, -l 80000000 defines the number of reads of each short read file, which is names as xaa, xab, sac, —-. The number of short read files depends on the length of your chromosome.

```
mkdir chr1
splitfa ../Zea_mays.AGPv3/Zea_mays.AGPv3.22.dna.chromosome.1.fa 100 |split -
l 80000000
mv xa* ./chr1
```

alignment the short reads to the genome with bwa aln, and then merge the sam files into a single one.

```
bwa aln -R 1000000 -O 3 -E 3
../Zea_mays.AGPv3/Zea_mays.AGPv3.22.dna.chromosome.1.fa ./chr1/xaa >
./chr1/xaa.sai
bwa samse ../Zea_mays.AGPv3/Zea_mays.AGPv3.22.dna.chromosome.1.fa
./chr1/xaa.sai ./chr1/xaa > ./chr1/xaa.sam

bwa aln -R 1000000 -O 3 -E 3
../Zea_mays.AGPv3/Zea_mays.AGPv3.22.dna.chromosome.1.fa ./chr1/xab >
./chr1/xab.sai
bwa samse ../Zea_mays.AGPv3/Zea_mays.AGPv3.22.dna.chromosome.1.fa
./chr1/xab.sai ./chr1/xab > ./chr1/xab.sam

bwa aln -R 1000000 -O 3 -E 3
../Zea_mays.AGPv3/Zea_mays.AGPv3.22.dna.chromosome.1.fa ./chr1/xac >
./chr1/xac.sai
bwa samse ../Zea_mays.AGPv3/Zea_mays.AGPv3.22.dna.chromosome.1.fa
./chr1/xac.sai ./chr1/xac > ./chr1/xac.sam
```

```
bwa aln -R 1000000 -O 3 -E 3
../Zea_mays.AGPv3/Zea_mays.AGPv3.22.dna.chromosome.1.fa ./chr1/xad >
./chr1/xad.sai
bwa samse ../Zea_mays.AGPv3/Zea_mays.AGPv3.22.dna.chromosome.1.fa
./chr1/xad.sai ./chr1/xad > ./chr1/xad.sam

bwa aln -R 1000000 -O 3 -E 3
../Zea_mays.AGPv3/Zea_mays.AGPv3.22.dna.chromosome.1.fa ./chr1/xae >
./chr1/xae.sai
bwa samse ../Zea_mays.AGPv3/Zea_mays.AGPv3.22.dna.chromosome.1.fa
./chr1/xae.sai ./chr1/xae > ./chr1/xae.sam

bwa aln -R 1000000 -O 3 -E 3
../Zea_mays.AGPv3/Zea_mays.AGPv3.22.dna.chromosome.1.fa ./chr1/xaf >
./chr1/xaf.sai
bwa samse ../Zea_mays.AGPv3/Zea_mays.AGPv3.22.dna.chromosome.1.fa
./chr1/xaf.sai ./chr1/xaf > ./chr1/xaf.sam

bwa aln -R 1000000 -O 3 -E 3
../Zea_mays.AGPv3/Zea_mays.AGPv3.22.dna.chromosome.1.fa ./chr1/xag >
./chr1/xag.sai
bwa samse ../Zea_mays.AGPv3/Zea_mays.AGPv3.22.dna.chromosome.1.fa
./chr1/xag.sai ./chr1/xag > ./chr1/xag.sam

bwa aln -R 1000000 -O 3 -E 3
../Zea_mays.AGPv3/Zea_mays.AGPv3.22.dna.chromosome.1.fa ./chr1/xah >
./chr1/xah.sai
bwa samse ../Zea_mays.AGPv3/Zea_mays.AGPv3.22.dna.chromosome.1.fa
./chr1/xah.sai ./chr1/xah > ./chr1/xah.sam


java -Xmx240g -Djava.io.tmpdir=${TMPDIR} -jar
/data004/software/GIF/packages/picard-tools/1.106/MergeSamFiles.jar
INPUT=./chr1/xaa.sam INPUT=./chr1/xab.sam INPUT=./chr1/xac.sam
INPUT=./chr1/xad.sam INPUT=./chr1/xae.sam INPUT=./chr1/xaf.sam
INPUT=./chr1/xag.sam INPUT=./chr1/xah.sam OUTPUT=./chr1/chr1.sam
SORT_ORDER=unsorted ASSUME_SORTED=false VALIDATION_STRINGENCY=LENIENT
TMP_DIR=${TMPDIR}
```

Generate the raw mask file and then set the stringency to make the final mask files.

```
perl /data004/software/GIF/packages/SNPable/20141106/gen_raw_mask.pl
./chr1/chr1.sam > ./chr1/chr1_rawMask_100.fa

gen_mask -l 100 -r 0.5 ./chr1/chr1_rawMask_100.fa >
./chr1/chr1_Mask_100_0.5.fa
```

Then I made a file called chr1.txt, which contained two columns (cur and pos), for pos column it indexes from 1 to the last position of the chromosome. Thus the length of chr1.txt is exactly the length of chr1. This file was then fed into the apply_mask_l to generate the chr1_mask.txt, which only keeps the uniquely mapped positions on chr1.

```
apply_mask_l ./chr1/chr1_Mask_100_0.5.fa chr1.txt > chr1_mask.txt
```

Finally, based on chr1_mask.txt, with awk one liner command, I generated the mappability mask file (three columns, chr1, start, end), which defines the SNPable region on the chromosome and the format is accepted by msmc.

```
awk 'NR==1{chr=$1;start=$2;end=$2;next} $2 == end+1 {end=$2;next} {print chr,start,end;start=$2;end=start} END{print chr,start,end}' chr1_mask.txt > chr1_mask_mappability.txt
```