How to simulate Heterozygous dataset

The following commands can be used to generate the two haplotypes of chromosome 1 of individual HG00096. In brief, these commands download the reference genome, extract chromosome 1, and apply the variants of individual HG00096 to it, producing two variants of chromosome 1 from which the reads are simulated.

First, we download the human genome reference, extract chromosome 1 and change the contig name to match the ID contained in the VCF file. The result is file `chr1.fa`.

```
base=ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp
reference=${base}/technical/reference/phase2_reference_assembly_sequence/hs37d5.fa.gz
wget $reference
gunzip $reference
csplit -s -z hs37d5.fa '/>/' '{*}'
cat xx00 | sed 's/>1 dna:chromosome chromosome:GRCh37:1:1:249250621:1/>1/g' > chr1.fa
```

At this point, we download the VCF file with all chromosome 1 variants of all 1000genomes project's individuals, and filter only SNPs and InDels of individual HG00096 using `vcftools`[1]. The result is file `HG00096.vcf.gz`.

```
vcf=ALL.chr1.phase3_shapeit2_mvncall_integrated_v5a.20130502.genotypes.vcf.gz
wget ${base}/release/20130502/${vcf}
zcat $vcf | vcf-subset -c HG00096 -t SNPs,indels | bgzip -c > HG00096.vcf.gz
```

To conclude, using `vcftools` we apply the variants of haplotypes 1 and 2 to `chr1.fa`, obtaining the modified chromosomes `HG00096_haplotype1.fa` and `HG00096_haplotype2.fa`:

```
tabix -p vcf HG00096.vcf.gz
cat chr1.fa | vcf-consensus -H 1 HG00096.vcf.gz > HG00096_haplotype1.fa
cat chr1.fa | vcf-consensus -H 2 HG00096.vcf.gz > HG00096_haplotype2.fa
```

Finally, use `SimSeq`[2] to simulate reads from `HG00096_haplotype1.fa` and `HG00096_haplotype2.fa`, uniformly distributing the coverage among the two chromosome's variants and using the HiSeq error profile[3] publicly available in the SimSeq's repository. In our experiments, we simulated 100-bp synthetic reads with total coverage ranging from 10x to 50x. Finally, we filtered each read set by removing reads containing the symbol N with the tool `fastp`[4]:

```
fastp -V -u 100 -A -n 0 -i input -o output
```

Note that the latter command is required as `ebwt2InDel` can only work on alphabet $\{A, C, G, T\}$.

[1] `http://vcftools.sourceforge.net/`
[2] `https://github.com/jstjohn/SimSeq`
[3] `https://github.com/jstjohn/SimSeq/blob/master/examples/`
[4] `https://github.com/OpenGene/fastp`