University of Zurich[UZH]

URPP Evolution in Action

URPP tutorial

NGS tools

Dr. Heidi E.L. Lischer
University of Zurich
Switzerland

30 October, 2015

Seqtk

Seqtk (https://github.com/lh3/seqtk):

- Tool for processing sequences in FASTA or FASTQ format
- Files can also be compressed by gzip

- **Usage:**

```
seqtk <command> <arguments>
```

- **Getting help:**
  - List of commands:

```
seqtk
```

  - List of arguments of a command:

```
seqtk <command>
```

Seqtk - commands

- Commands:
  - seq
    - Common transformation of FASTA/Q

```
#convert FASTQ to FASTA
seqtk seq -A in.fq.gz > out.fa
```

    - Mask certain bases
    - Quality transformation
    - Reverse complement

```
#reverse complement FASTQ file
seqtk seq -r in.fq > out.fq
```

  - comp get the nucleotide composition of FASTA/Q
  - sample random subsample sequences
  - fqchk fastq QC (base/quality summary)

Seqtk - commands

- subseq — extract subsequences from FASTA/Q

```
#Extract sequences with names in file name.txt
seqtk subseq in.fq.gz name.txt > out.fa
```

- mergepe — interleave two PE FASTA/Q files
- trimfq — trim FASTQ by quality scores / trim fixed length from ends
- hety — regional heterozygosity (FASTA)
- gc — identify high- or low-GC regions (FASTA)
- mutfa — point mutate FASTA at specified positions
- mergefa — merge two FASTA/Q files
- dropse — drop unpaired from interleaved PE FASTA/Q
- rename — rename sequence names
- randbase — choose a random base from hets (FASTA)
- cutN — cut sequence at long N (FASTA)
- listhet — extract the position of each het (FASTA)

VCFtools

VCFtools (https://vcftools.github.io/index.html):

- Program package designed for working with VCF files (Variant Call Format)

- Filter out specific variants
- Compare files
- Summarize variants
- Convert to different file types
- Validate and merge files
- Create intersections and subsets of variants

VCFtools

- Usage:

```
vcftools <arguments>
```

- Getting help:
  - manual: https://vcftools.github.io/man_latest.html

    or

```
man vcftools
```

- Get basic file statistics (e.g. number of variants and individuals)

```
vcftools --vcf input.vcf          #vcf file
vcftools --gzvcf input.vcf.gz     #compressed vcf file
vcftools --bcf input.bcf          #binary vcf file
```

VCFtools

- A lot of filter possibilities
  - Variants (position, SNPs, type, filter flag, …)
  - Alleles (MAF, allele counts, …)
  - Genotypes (depth, HWE, missing, phased, quality, …)
  - Individuals
  - Examples:

```
#filter sites based on location
vcftools --vcf input.vcf --chr 1 --from-bp 500 --to-bp 10000

#remove indels
vcftools --vcf input.vcf --remove-indels

#remove alleles with a minor allele frequency < 0.1
vcftools --vcf input.vcf --maf 0.1

#remove ind2 and ind4
vcftools --vcf input.vcf --remove-indv ind2 --remove-indv ind4
```

VCFtools

- Variants that pass filters
  - Perform analyses (see later)
  - Write to a new VCF files: **--recode**

```
#write new VCF file
vcftools --vcf input.vcf --chr 1 --from-bp 500 --to-bp 10000 \
         --recode
```

→ writes a file to ./out.recode.vcf

```
#write new VCF file
vcftools --vcf input.vcf --chr 1 --from-bp 500 --to-bp 10000 \
         --recode --out input_filtered
```

→ writes a file input_filtered.recode.vcf

  - Write to standard out: **--stdout** or **-c**

```
vcftools --vcf input.vcf --chr 1 --from-bp 500 --to-bp 10000 \
         --recode --stdout | gzip -c > input_filtered.vcf.gz
```

VCFtools

- Convert VCF files:
  - to BCF (binary format of VCF): **--recode-bcf**

```
#writes a compressed bcf file using BGZF
vcftools --vcf input.vcf --recode-bcf --out input_converted
```

→ writes a file input_converted.bcf

  - to PLINK: **--plink**

```
#writes a compressed bcf file using BGZF
vcftools --vcf input.vcf --plink --out input_converted
```

→ writes variants in input_converted.ped and input_converted.map

- Compare VCF files: **--diff-site** or **–diff-indv**

```
vcftools --gzvcf input.vcf.gz --gzdiff input2.vcf.gz \
         --diff-site --out output
Found 343017 sites common to both files.
Found 181815 sites only in main file. ...
```

VCFtools

- Get SNP statistics:
  - allele frequencies over all individuals: **`--freq`**

```
vcftools --vcf input.vcf --freq --out output

output.frq
CHROM POS    N_ALLELES    N_CHR {ALLELE:FREQ}
18    10719 2      120    C:0.991667  G:0.00833333
...
```

  - sequence depth: **`--depth`** (mean depth per individual)
    `--site-depth` (depth per site)

```
vcftools --vcf input.vcf --depth --out output
```

VCFtools

- Estimate population genetic parameters
  - Linkage disequilibrium: **--hap-r2** or **--geno-r2** or **--geno-chisq**

```
#estimate r2 between sites within 50kb of one another
vcftools --vcf input.vcf --geno-r2 --ld-window-bp 50000 \
         --out output
output.geno.ld
CHR    POS1   POS2   N_INDV      R^2
chr1   13554  14594  17     0.259286
chr1   13554  14750  10     0.126984     ...
```

  - Fst: **--weir-fst-pop**

Text file containing list of individuals of a population

```
#estimate per site Fst with two populations
vcftools --vcf input.vcf --weir-fst-pop pop1.txt \
         --weir-fst-pop pop2.txt --out output
output.weir.fst
CHROM POS    WEIR_AND_COCKERHAM_FST
chr1  5145899     0.172223          ...
```

VCFtools

- Nucleotide diversity: **`--site-pi`**
- TajimaD within bin size: **`--TajimaD`** `integer`
- Calculate inbreeding coefficient: **`--het`**
- Hardy-Weinberg Equilibrium test: **`--hardy`**
- Identify long runs of homozygosity: **`--LROH`**
- Estimate relatedness: **`--relatedness`**

→ and many more statistics!

BCFtools

BCFtools (https://samtools.github.io/bcftools/bcftools.html)

- Tools for manipulating VCF and BCF files

- **Usage:**

```
bcftools <command> <arguments>
```

- **Getting help:**
  - manual: https://samtools.github.io/bcftools/bcftools.html
  - List of commands:

```
bcftools
```

  - List of arguments of a command:

```
bcftools <command>
```

BCFtools - commands

- Commands:
  - annotate — edit VCF files, add or remove annotations
  - call — SNP/indel calling
  - concat — concatenate VCF/BCF files from the same set of samples
  - consensus — create consensus sequence by applying VCF variants

```
bcftools consensus -f ref.fa in.vcf.gz -o out.fa
```

  - convert — convert VCF/BCF to other formats and back:

    e.g. VCF, BCF, gVCF, TSV, …
  - filter — filter VCF/BCF files using fixed thresholds
  - gtcheck — check sample concordance, detect sample swaps and contamination
  - index — index VCF/BCF

BCFtools - commands

- isec — intersections, unions and complements of VCF/BCF files

```
#creates intersections and complemnets of two files
bcftools isec A.vcf.gz B.vcf.gz -p out_dir
```

- merge — merge VCF/BCF files from non-overlapping sample sets
- norm — left-align and normalize indels
- plugin — run user-defined plugin (e.g.: count SNPs,…)
- query — transform VCF/BCF into user-defined formats
- reheader — modify VCF/BCF header, change sample names
- roh — identify runs of homo/auto-zygosity
- stats — produce VCF/BCF stats
- view — subset, filter and convert VCF and BCF files
  e.g. filter regions, samples, genotypes, allele frequency…

Acknowledgment

- Sources:
  - https://github.com/lh3/seqtk
  - https://vcftools.github.io/index.html
  - https://samtools.github.io/bcftools/bcftools.html