UNIVERSITÄT BERN

RRHS

RRHS version 1.0.0.1 (March 2014)

A tool for Repeated Random Haplotype Sampling

Author: Heidi Lischer

Computational and Molecular Population Genetics lab (CMPG)
Institute of Ecology and Evolution (IEE)
University of Berne
3012 Bern
Switzerland

Member of the Swiss Institute of Bioinformatics (SIB)

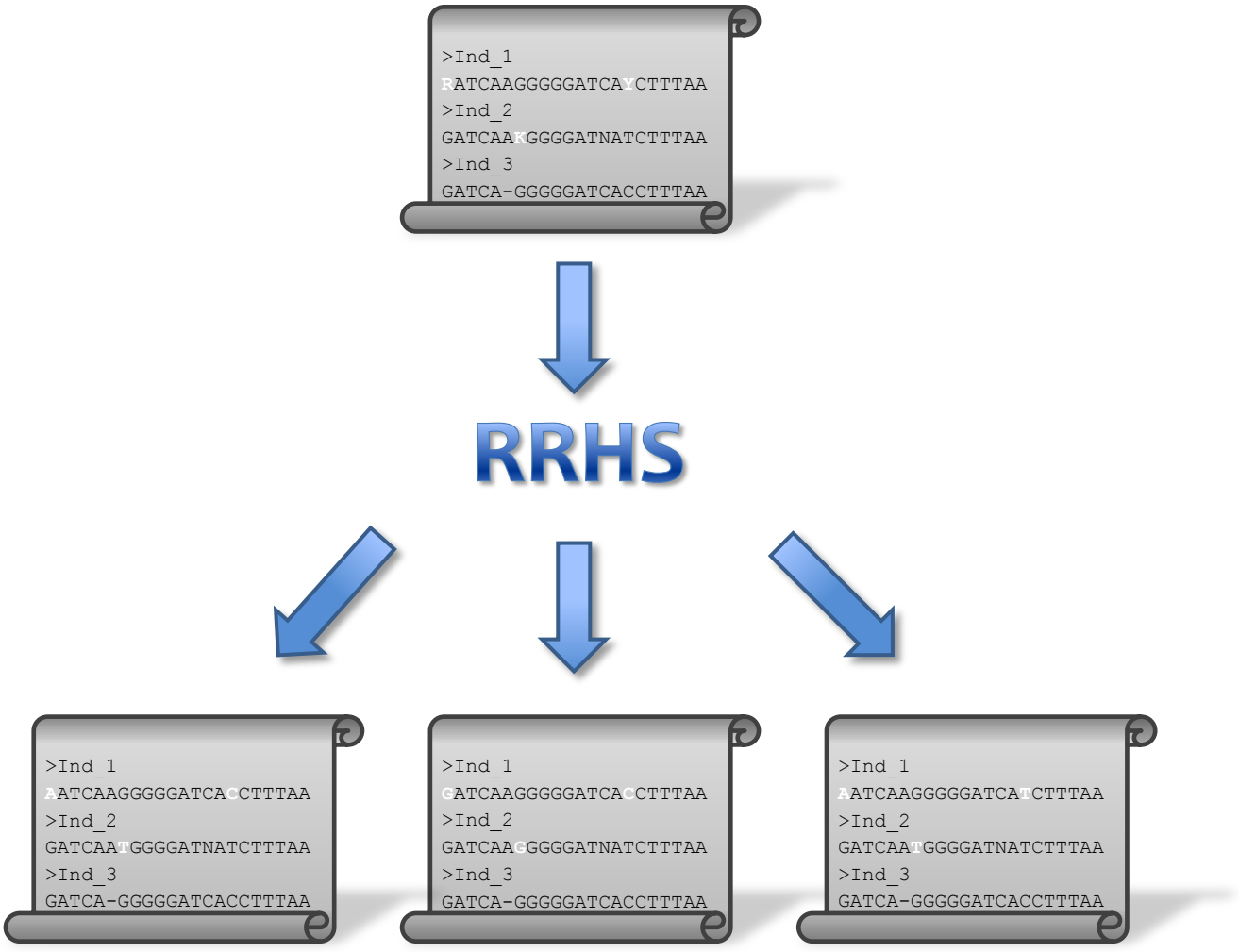
e-mail: heidi.lischer@iee.unibe.ch
Download: http://www.cmpg.iee.unibe.ch/content/softwares__services/computer_programs/rrhs/

Contents

1. Introduction

Phylogenetic reconstruction of the evolutionary history of closely related organisms may be difficult, because they potentially contain a relatively high proportion of heterozygous sites that are usually not handled well by phylogenetic programs. The exclusion of heterozygous sites from evolutionary analysis may cause biased and misleading divergence time estimations in closely related taxa. An approach of repeated random haplotype sampling (RRHS) from sequences with multiple unphased heterozygous sites has been shown to successfully integrate heterozygous information into existing phylogenetic programs (Lischer, et al. in press).

Here we provide the RRHS tool for automatized Repeated Random Haplotype Sampling. The program is able to read in 5 different formats (FASTA, PHYLIP, RAxML, NEXUS and VCF) and outputs random haplotypes between loci and unphased sites in 4 possible formats (FASTA, PHYLIP, RAxML and NEXUS). Heterozygous positions are either provided as ambiguity codes (Y, R, W, S , K or M) in the FASTA, PHYLIP, RAxML or NEXUS format or directly supplied in the VCF format. Some formats do not provide the possibility to specify loci partitions, thus an additional partition file with loci boundaries can be supplied.



2. System requirements

RRHS is written in Java and therefore platform independent, but SUN Java 1.6 RE (or a newer version) has to be installed. Java6 RE can be downloaded under following link:
http://www.oracle.com/technetwork/java/javase/downloads/index.html

3. Installation instructions

1st step:

Install the Java6 RE

- Windows:
  download and install Java6 RE with following link:
  http://www.oracle.com/technetwork/java/javase/downloads/index.html

- Linux:
  - Ubuntu / Debian:
    Execute the following command as root user:
    "apt-get install openjdk-6-jre"
  - Other Linux distributions:
    http://www.oracle.com/technetwork/java/javase/downloads/index.html

- Mac:
  Apple Computer supplies their own version of Java. Use the Software Update feature (available on the Apple menu) to check that you have the most up-to-date version of Java for your Mac. Additionally, make sure that Java version 1.6 is set as first preference version. This can be changed under "Applications - Utilities - Java Preferences.app".
  If you have problems with downloading, installing or using Java on Mac, please contact Apple Computer Technical Support.

2nd step:

Download the RRHS application from
http://www.cmpg.iee.unibe.ch/content/softwares__services/computer_programs/rrhs/ and unzip it on the local drive.

- Execute RRHS (command line): execute the command "`java –jar RRHS.jar`"

4. Execute RRHS

Execute RRHS (command line): execute the command "`java –jar RRHS.jar`" in the console

RRHS can be executed with the following options (the order does not matter):

- **-?** or **–h**:
  To show a help text with the different options

- **-i <file>** (mandatory):
  Specify the path to the input file

- **-iFormat <format>** (mandatory):
  Specify the format of the input file: VCF, FASTA, PHYLIP, RAxML or NEXUS

- **-interleaved**:
  Specify if PHYLIP or RAxML is in interleaved format

- **-o <file>**:
  Specify the path to the output files. If not given the input file path is used.

- **-n <integer>**:
  Specify the number (repetitions) of random haplotypes

- **-part <file>**:
  Specify the path to the partition file with loci boundaries, which may be provided in case of a FASTA, PHYLIP or RAxML input file. The partition file has to be in following format: Each line correspond to one loci (partition) starting with "DNA,", followed by the locus name, equal sign and finally the position of the first and last base. For example:

  ```
  ACRTTAA CGGTATATATCG YGTAAACCTGAAGGTTCTGAAGCT
  ```

  ```
  DNA, locus1 = 1-7
  DNA, locus2 = 8-19
  DNA, locus3 = 20-43
  ```

- **-mQual <integer>**:
  Specify the minimum SNP quality a polymorphic position needs to have to be taken in case of a VCF input file.

- **-fasta**:
  Specify if output format should be FASTA

- **-phylip**:
  Specify if output format should be PHYLIP

- **-RAxML**:
  Specify if output format should be RAxML

- **-nexus**:
  Specify if output format should be nexus. If the input format was already NEXUS, the output file will contain the same blocks as the input file (e.g. MrBayes block). Therefore, the files will contain the same information, except for the random haplotype sequences.

4.1. Examples

- call help:
  `java –jar RRHS.jar -?` or `java –jar RRHS.jar –h`

- read a FASTA file and output 100 PHYLIP files with random haplotypes:
  ```
  java –jar RRHS.jar –i D:\RRHS\example_fasta.fa –iFormat FASTA
  –o D:\RRHS\out.txt –phylip –n 100
  ```

- read a FASTA file with a file giving the loci partitions and output 1000 RAxML files with random haplotypes:
  ```
  java –jar RRHS.jar –i D:\RRHS\example_fasta.fa –iFormat FASTA
  -part D:\RRHS\example_partitions.txt –o D:\RRHS\out –raxml –n
  1000
  ```

- read a interleaved PHYLIP file and output 500 NEXUS files with random haplotypes:
  ```
  java –jar RRHS.jar –i D:\RRHS\example_phylip.txt –iFormat
  PHYLIP –interleaved –nexus –n 500
  ```

- read a VCF file with a minimum SNP quality of 40 and output 100 FASTA files with random haplotypes:
  ```
  java –jar RRHS.jar –i D:\RRHS\example_vcf.vcf –iFormat VCF
  –mQual 40 –o D:\RRHS\VCF\example_out –fasta –n 100
  ```

4.2. Increase Memory

To increase the memory RRHS is allowed to use start the program by executing the command

"`java –Xmx1024m –jar RRHS.jar`" and adapt the `–Xmx` parameter to your needs (`–Xmx1024m` means: maximum memory of 1’024 MB).

5. Reporting bugs and comments

If there are any bugs, send me an e-mail. Please give me a short description of the bug and tell me the input and output file format. If it is possible also attach the input file which caused the problem.

e-mail address: heidi.lischer@iee.unibe.ch

6. How to cite

Lischer HEL, Excoffier E, Heckel G. 2014. Ignoring heterozygous sites biases phylogenomic estimates of divergence times: implications for the evolutionary history of *Microtus* voles. Mol Biol Evol 31 (4): 817-831.

7. File format descriptions

7.1. FASTA

FASTA format is a text based format for representing either nucleic acid sequences or peptide sequences, in which base pairs or amino acids are represented using single-letter codes. Sequence names and comments can also be included before the sequences (Pearson 1990).

7.1.1. Format

- FASTA has no standard file extension. The following extensions are often used: .fa, .mpfa, .fna, .fsa, .fas or .fasta
- The FASTA format begins with a single line description, followed by lines of sequence data. It is recommended that all lines of text be shorter than 80 characters.
- The sequence ends if another line starting with a ”>” appears (this indicates the start of another sequence)

- The header line is arranged as follows:
  - It begins with a ”>”
  - The following word following is the identifier and/or name of the sequence (optional)
  - The rest of the line is the description (optional)
  - There should be no space between the ”>” and the first letter of the identifier
  - The header line may contain more than one header separated by a ^A (Control-A) character
  - Possible sequence identifiers: Many different sequence databases use standardized headers, which helps to automatically extract information from the header:

GenBank	`“gi”|gi-number|”gb”|accession|locus`
EMBL Data Library	`“gi”|gi-number|”emb”|accession|locus`
DDBJ, DNA Database of Japan	`“gi”|gi-number|”dbj”|accession|locus`
General database identifier	`“gnl”|database|identifier`
“simply”	`identifier`

- Sequence representation:
  - The sequences comes after the header line and comments
  - each line of a sequence should have fewer than 80 characters
  - Sequences can contain gaps or alignment characters

- Sequences are expected to be represented in the standard IUPAC nucleic acid codes, with these exceptions: lower-case letters are accepted and are mapped into upper-case, a single hyphen or dash can be used to represent a gap character
- Numerical digits are not allowed but are used in some databases to indicate the position in the sequence

- simple example of sequences from 3 individuals:

```
>Ind_1
C-CTAGGACTA---GATCAAGGGGGATCAYCTTTAAGCCAATATATGCTCTGGTCC
AACTTACGCGCTA
>Ind_2
A-CCAGGACTAGCGGATCAASGGGGATCATCTTTAAGCCAATATATGCTCTGGTCC
AACTTACGCGCTA
>Ind_3
R-CCAGGACTA---GATCAAGGGGGATCACCTTTAAGCCAATATATGCTCTNNNNC
AACTTACGCGCTA
```

7.1.2. Links

Wikipedia: http://en.wikipedia.org/wiki/FASTA_format

NCBI's FASTA format description: http://www.ncbi.nlm.nih.gov/blast/fasta.shtml

7.2. PHYLIP

PHYLIP version 3.69 (September 2009)

PHYLIP, the Phylogeny Inference Package, is a package of programs for inferring phylogenies (evolutionary trees). It can infer phylogenies by parsimony, compatibility, distance matrix methods, and likelihood. It can also compute consensus trees, compute distances between trees, draw trees, resample data sets by bootstrapping or jackknifing, edit trees, and compute distance matrices (Felsenstein 1989, 2004).

7.2.1. Format

For most of the PHYLIP programs, information comes from a series of input files, and ends up in a series of output files.

Nucleotide sequences data:

- The first line contains the number of species and the number of characters. These are in free format, separated by blanks.
- The next lines include information for each species: First, the species name has to be 10 characters long (it can include blanks and punctuation marks), followed by the data for that species (the data have to start at the 11th character of the line!). The name should be on the same line as the first character of the data.
  In the relaxed PHYLIP format (e.g. for RAxML) the species names could be of any length and are separated from the data by a whitespace.
- The conventions for interleaved data are different between the molecular sequence programs and the others. The molecular sequence programs can take the data in “aligned” or “interleaved” format:
  - In the interleaved format DNA sequences can be specified on several lines. It is important that the sequence length in each group is the same for all species. The sequences might look like this:

```
    2   39
Archaeopt CGATGCTTAC CGCCGATGCT
HesperorniCGTTACTCGT TGTCGTTACT

TACCGCCGAT GCTTACCGC
CGTTGTCGTT ACTCGTTGT
```

- In the sequential format the character data can run on a new line at any time. Thus, it is legal to have:

```
    2   39
Archaeopt CGATGCTTAC CGCCGATGCT
TACCGCCGAT GCTTACCGC
Hesperorni
CGTTACTCGT TGTCGTTACTCGTTGTCGTT
ACTCGTTGT
```

- Blanks and digits within sequences are allowed to make them easier to read
- Example:

```
   6   13
Archaeopt CGATGCTTAC CGC
HesperorniCGTTACTCGT TGT
BaluchitheTAATGTTAAT TGT
B. virginiTAATGTTCGT TGT
BrontosaurCAAAACCCAT CAT
B.subtilisGGCAGCCAAT CAC
```

7.2.2. Links and References

Website: http://evolution.genetics.washington.edu/phylip/doc/main.html

(Felsenstein 1989, 2004)

7.3. RAxML

RAxML version 7.2.8

RAxML (Randomized Axelerated Maximum Likelihood) is a program for sequential and parallel Maximum Likelihood based inference of large phylogenetic trees (Stamatakis 2006).

7.3.1. Format

The input format of RAxML corresponds to a relaxed interleaved or sequential PHYLIP format. Relaxed means that sequence names can be of variable length between 1 up to 256 characters.

- Example of relaxed PHYLIP format:

```
   6   13
Archaeopt CGATGCTTAC CGC
Hesperorni CGTTACTCGT TGT
Baluchithea TAATGTTAAT TGT
B.virgini TAATGTTCGT TGT
Brontosaurus CAAAACCCAT CAT
B.subtilis GGCAGCCAAT CAC
```

7.3.2. Links and References

Website: http://www.exelixis-lab.org/

(Stamatakis 2006)

7.4. NEXUS

NEXUS is a file format designed to contain systematic data. The goals of the format are to allow future expansion, to include diverse kinds of information, to be independent of particular computer operating systems, and to be easily processed by a program (Maddison, et al. 1997).

7.4.1. NEXUS format

NEXUS files are free-format, which means that the entire file could conceivably consist of a single, long line of text. It does not matter where the line is broken (as long as you don't split up a keyword or the name of a locus, allele or population), nor does it matter if one space or a dozen spaces are used to separate the individual words (tokens) in the file. Tokens may be casually defined as sequences of characters separated by whitespace (e.g., spaces, carriage returns, line feeds, tabs, etc.)

NEXUS files are for the most part not case-sensitive by default. A big exception is in the matrix command, where (by default) an allele named A is treated as being distinct from a.

The NEXUS files are built as follows:

- Comments can be added by enclosing text within brackets: `[comment]`
- The file has to start with: `#NEXUS`
- The tokens in a NEXUS file are organized into commands, which are in turn organized into blocks.
    - Commands: the first token in the command is the command name, which is followed by a series of tokens and whitespace; the command is terminated by a semicolon: `command-name token token . . . ;`
    - Blocks: series of commands, beginning with a BEGIN command and ending with an END command:

```
BEGIN block-name;
 command-name token . . . ;
 command-name token . . . ;
 ...
END;
```

The most used public blocks are:

(Tokens within [ ] are optional, within { | | } are mutually exclusive and underlined tokens are the default):

- CHARACTERS:
  This block contains the information about discrete and continuous data, including that for morphological structure and molecular sequences. Polymorphism and frequency data can be accommodated. Names can be given to the characters and their states.

```
BEGIN CHARACTERS;
 DIMENSIONS [NEWTAXA NTAX=number-of-taxa] NCHAR=number-of-
characters;
 [FORMAT
  [DATATYPE={STANDARD|DNA|RNA|NUCLEOTIDE|PROTEIN|CONTINUOUS}
  [RESPECTCASE]                                    default: A
and a is the same
  [MISSING=symbol]                                 default: ?
  [GAP=symbol]
  [SYMBOLS="symbol [symbol...]"]
  [EQUATE="symbol=entry [symbol=entry]"]
  [MATCHCHAR=symbol]
  [[No]LABELS]
  [TRANSPOSE]
  [INTERLEAVE]

[ITEMS=([MIN][MAX][MEDIAN][AVERAGE][VARIANCE][STCERROR][SAMPL
ESIZE][STATES])]
  [STATESFORMAT={STATESPRESENT|INDIVIDUALS|COUNT|FREQUENCY}]
  [[No]TOKENS]
 ;]
 [ELIMINATE character-set;]
 [TAXLABELS taxon-name [taxon-name...];]
 [CARSTATELABELS character-number [charact-name] [/state-name
[state-name..]]
  [, character-number [character-name] [/state-name [state-
name...]] ...]
 ;]
 [CHARLABELS character-name [character-name...];]
 [STATELABELS character-number [character-name] [/state-name
[state-name...]]
  [, character-number [character-name] [/state-name [state-
name...]]  ...]
 ;]
 MATRIX data-matrix;
END;
```

- example:

```
BEGIN CHARACTERS;
 DIMENSION NCHAR=20;
MATRIX
  taxon_1 R-CTAGGACTA---GATCAA
  taxon_2 A-CCAGGACTAGCGGATCAA
  taxon_3 AGCCAGGACTA---GTTCAA
END;
```

- DATA:
  DATA is a CHARACTERS block that includes not only the definition of characters but also the definition of taxa.
  - example:

```
BEGIN DATA;
 DIMENSIONS NTAX=5 NCHAR=20;
 FORMAT DATATYPE=DNA GAP=-;
 MATRIX
  taxon-1 A-CTAGGACTA---GATCAA
  taxon-2 A-CCAGGACTAGCGGATCAA
  taxon-3 A-CCAGGACTA---GATCAA
  taxon-4 AGCCAGGACTA---GTTCAA
  taxon-5 ATC-AGGACTA---GATCAA;
END;
```

- SETS:
  This block contains descriptions of collections of objects. These objects include characters, taxa, trees, states, and kinds of changes. In addition, partitions of characters, taxa, and trees can be formed.

```
BEGIN SETS;
 [CHARSET charstet_name [({STANDARD|VECTOR})]=character-set;]
 [STATESET stateset-name [({STANDARD|VECTOR})]=state-set;]
 [CHANGESET changeset-name=state-set<->state-set [state-set<-
>state-set...];]
 [TAXSET taxset-name [({STANDARD|VECTOR})]=taxon-set;]
 [TREESET treeset-name [({STANDARD|VECTOR})]=tree-set;]
 [CHARPARTITION partition-name [([{[NO]TOKENS}]
[{STANDARD|VECTOR}])]
  =subset-name:character-set [, subset-name:character-
set...];]
 [TAXPARTITION partition-name [([{[NO]TOKENS}]
[{STANDARD|VECTOR}])]
  =subset-name:taxon-set [, subset-name:taxon-set...];]
 [TREEPARTITION partition-name [([{[NO]TOKENS}]
[{STANDARD|VECTOR}])]
```

- Names should be unique (no duplicate names), must be single words (no spaces) and cannot consist entirely of digits.

Example:

```
#NEXUS
BEGIN TAXA;
      Dimensions NTax=4;
      TaxLabels fish frog snake mouse;
END;

BEGIN CHARACTERS;
      Dimensions NChar=20;
      Format DataType=DNA;
      Matrix
        fish   ACATA GAGGG TACCT CTAAG
        frog   ACATA GAGGG TACCT CTAAG
        snake  ACATA GAGGG TACCT CTAAG
        mouse  ACATA GAGGG TACCT CTAAG
END;

BEGIN SETS;
     CharSet loci1 = 1-10;
     CharSet loci2 = 11-20;
END;
```

7.4.2. References

Maddison, D. R., D. L. Swofford, et al. (1997). "Nexus: An extensible file format for systematic information." Systematic Biology **46**(4): 590-621.

7.5. VCF

VCF version 4.1 (2. August 2012) without structural variants (only SNP and INDELs)

VCF (Variant Call Format) format stores structural variant data.

7.5.1. VCF format

VCF is a tab-delimited text format with following file extension: *.vcf

The format contains meta-information lines, a header line, and data lines which contain information about a position in the genome.

Meta-information lines

- begins with `##`
- must be key=value pairs
- 'fileformat' (mandatory):
  - VCF format version
  - e.g.: `##fileformat=VCFv4.1`
- 'INFO':
  - `##INFO=<Flag_ID>,<Number_of_Values>,<Value_Type>,<Description>`
  - `<Number_of_Values>`: Integer that describes the number of values that can be included in the INFO field (values varies, unknown or unbounded: -1)
  - `<Value_Types>`: Integer, Float, Character, String and Flag. The 'Flag' type indicates that the INFO field does not contain a Value entry, and hence `<Number_of_Values>` should be 0 in that case.
- 'FILTER':
  - Filters that have been applied to the data
  - `##FILTER=<FILTER_ID>,<Description>`
- 'FORMAT':
  - `##FORMAT=<FORMAT_ID>,<Number_of_Values>,<Value_Type>,<Description>`
  - `<Value_Types>`: Integer, Float, Character, and String.

Header line

- tab delimited
- names the 8 fixed, mandatory columns:
  1. #CHROM
  2. POS
  3. ID

4. REF
5. ALT
6. QUAL
7. FILTER
8. INFO

- If genotype data is present:
    9. FORMAT column header
    10. an arbitrary number of sample ids

Data line

Fixed fields:

- tab-delimited
- missing values: "."
- 8 fixed fields per record:
    1. CHROM chromosome:
        - an identifier from the reference genome.
        - Alphanumeric String, required
    2. POS position:
        - The reference position (1st base having position 1).
        - Positions are sorted numerically, in increasing order, within each reference sequence.
        - Integer, required
    3. ID:
        - A unique identifier. If this is a dbSNP variant: use the rs number.
        - Alphanumeric String, Missing value: ”.”
    4. REF reference base:
        - One of A, C, G, T, N. Bases should be in uppercase.
        - Multiple bases are permitted. The value in the POS field refers to the position of the first base in the String.
        - For InDels, the reference String must include the base before the event (which must be reflected in the POS field).
        - String, required
    5. ALT:
        - Comma separated list of alternate non-reference alleles.
        - Options are A, C, G, T, Dn (for delete n bases starting with the base at POS), I<seq> (where <seq> is a list of ACGT bases to be inserted just after the base at POS).
        - If there are no alternative alleles, then period character should be used.
        - Bases should be in uppercase.
        - Alphanumeric String, Missing value: ”.”
    6. QUAL:
        - Phred-scaled quality scores for the assertion made in ALT.

- If ALT is "." (no variant) then this is -10log_10 p(variant) and if ALT is not "." this is -10log_10 p(no variant).
- High QUAL scores indicate high confidence calls.
- Although traditionally people use integer phred scores, this field is permitted to be a floating point so to enable higher resolution for low confidence calls if desired.
- Numeric, Missing Value: -1

7. FILTER filter:
   - PASS if this position has passed all filters
   - If site not passed all filters, a semicolon-separated list of codes for filters that fail.
   - Alphanumeric String, Missing Value: "."
8. INFO additional information:
   - Alphanumeric String, Missing Value: "."
   - Encoded as a semicolon-separated series of short keys with optional values in the format: <key>=<data>[,data]. The subfields could be e.g.:
     - AA: ancestral allele
     - AC: allele count in genotypes, for each ALT allele, in the same order as listed
     - AF: allele frequency for each ALT allele in the same order as listed: use this when estimated from primary data, not called genotypes
     - AN: total number of alleles in called genotypes
     - BQ: RMS base quality at this position
     - CIGAR: cigar string describing how to align an alternate allele to the reference allele
     - DB: dbSNP membership
     - DP: combined depth across samples, e.g. D=154
     - END: end position of the variant described in this record
     - H2: membership in hapmap2
     - H3: membership in hapmap3
     - MQ: RMS mapping quality, e.g. MQ=52
     - MQ0: Number of MAPQ == 0 reads covering this record
     - NS: Number of samples with data
     - SB: strand bias at this position
     - SOMATIC: indicates that the record is a somatic mutation, for cancer genomics
     - VALIDATED: validated by follow-up experiments
     - 1000G: membership in 1000 Genomes
     - etc. The exact format of each INFO subfield should be specified in the meta-information.
   - It is not necessary to list all the properties that a site does NOT have, by e.g. H2=0.

Genotype fields:

- If genotype information is present, then the same types of data must be present for all samples.
- First a FORMAT field is given specifying the data types and order.

- This is followed by one field per sample, with the colon-separated data in this field corresponding to the types specified in the format.
- The first subfield must always be the genotype (GT)
- There are several common, reserved keywords, which are defined as follows:
    - GT genotype (mandatory):
        - encoded as alleles values separated by ”/” or “|”
        - e.g.: The allele values are 0 for the reference allele (what is in the reference sequence), 1 for the first allele listed in ALT, 2 for the second allele list in ALT and so on. For diploid calls examples could be 0/1 or 1|0 etc.
        - For haploid calls (Y, male X, mitochondrion) only one allele value should be given.
        - missing allele: ”.” (e.g.: ./. for a diploid).
        - The meanings of the separators are:
          “/”': genotype unphased
          “|”': genotype phased.
    - DP:
        - read depth at this position for this sample
        - Integer, Missing value: -1
    - FT:
        - sample genotype filter indicating if this genotype was “called” (similar in concept to the FILTER record for the entire CHROM/POS)
        - PASS: indicate that all filters have been passed
        - a semi-colon separated list of codes for filters that fail
        - ”.”: indicate that filters have not been applied.
        - These values should be described in the meta-information in the same way as FILTERs
        - Alphanumeric String, Missing value: ”.”
    - GL genotype likelihoods:
        - Comma separated log10-scaled likelihoods for all possible genotypes given the set of alleles defined in the REF and ALT fields.
        - If A is the allele in REF and B,C,... are the alleles as ordered in ALT, the ordering of genotypes for the likelihoods is given by: F(j/k) = (k*(k+1)/2)+j
          e.g.: for biallelic sites the ordering is: AA,AB,BB;
          for triallelic sites the ordering is: AA,AB,BB,AC,BC,CC
    - GLE:
        - Genotype likelihoods of heterogenous ploidy
    - PL:
        - Phred-scaled genotype likelihoods rounded to the closest integer
        - Ordering like in GL
    - GP:
        - Phred-scaled genotype posterior probabilities
    - GQ genotype quality:
        - encoded as a phred quality (genotype call is wrong)
        - max quality 99

- Integer, Missing value: -1
- HQ haplotype qualities:
  - two phred qualities comma separated
  - Integer, Missing value: -1 for each quality. e.g. ”-1,-1”
- PS phase set:
  - Non-negative 32-bit integer
- PQ phasing quality:
  - Phred-scaled probability that alleles are ordered incorrectly in a heterozygote
- EC:
  - Comma separated list of expected alternate allele counts for each alternate allele in the same order as listed in the ALT field
- MQ:
  - RMS mapping quality
- Additional Genotype fields can be defined in the meta-information

Example:

```
##fileformat=VCFv4.0
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=1000GenomesPilot-NCBI36
##phasing=partial
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=.,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FILTER=<ID=q10,Description="Quality below 10">
##FILTER=<ID=s50,Description="Less than 50% of samples have data">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##FORMAT=<ID=HQ,Number=2,Type=Integer,Description="Haplotype Quality">
#CHROM POS     ID        REF  ALT     QUAL FILTER INFO
      FORMAT            NA00001          NA00002          NA00003
20    14370   rs6054257 G    A       29   PASS   NS=3;DP=14;AF=0.5;DB;H2
      GT:GQ:DP:HQ       0|0:48:1:51,51         1|0:48:8:51,51  1/1:43:5:.,.
20    17330   .         T    A       3    q10    NS=3;DP=11;AF=0.017
      GT:GQ:DP:HQ       0|0:49:3:58,50   0|1:3:5:65,3       0/0:41:3
20    1110696 rs6040355 A    G,T     67   PASS
      NS=2;DP=10;AF=0.333,0.667;AA=T;DB GT:GQ:DP:HQ          1|2:21:6:23,27
20    1230237 .         T    .       47   PASS   NS=3;DP=13;AA=T
      GT:GQ:DP:HQ       0|0:54:7:56,60   0|0:48:4:51,51     0/0:61:2
20    1234567 microsat1 GTCT G,GTACT 50   PASS   NS=3;DP=9;AA=G
      GT:GQ:DP          0/1:35:4         0/2:17:2           1/1:40:3
```

7.5.2. Links and References

Website: http://www.1000genomes.org/wiki/Analysis/Variant%20Call%20Format/vcf-variant-call-format-version-41

8. References

Felsenstein J. 1989. PHYLIP - Phylogeny inference package (version 3.2). Cladistics 5: 164-166.

Felsenstein J. 2004. PHYLIP (Phylogeny Inference Package) version 3.6. Distributed by the author. Department of Genome Sciences, University of Washington, Seattle.

Lischer HEL, Excoffier E, Heckel G. in press. Ignoring heterozygous sites biases phylogenomic estimates of divergence times: implications for the evolutionary history of *Microtus* voles. Mol Biol Evol.

Maddison DR, Swofford DL, Maddison WP. 1997. Nexus: An extensible file format for systematic information. Syst Biol 46: 590-621.

Pearson WR. 1990. Rapid and sensitive sequence comparison with FASTP and FASTA. Method Enzymol 183: 63-98.

Stamatakis A. 2006. RAxML-VI-HPC: Maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. Bioinformatics 22: 2688-2690.