# MVFtools Documentation

***Release 2017-06-24***

**James B. Pease, Ben K. Rosenzweig**

**Sep 26, 2017**

# CONTENTS:

CHAPTER

ONE

# GETTING STARTED

## 1.1 What is MVFtools?

Multisample Variant Format (MVF), is designed for compact storage and efficient analysis of multi-genome and multi-transcriptome datasets. The programs provided in MVFtools support this format, both with conversion utilities, filtering and transformation programs, and data analysis and visualization modules. MVF format is designed specifically for biological data analysis, since sequence data is encoded based on the information content at a particular aligned sequence site. This contextual encoding allows for rapid computation of phylogenetic and population genetic analyses, and small file sizes that enable data sharing and distribution.

## 1.2 How do I cite this ?

Pease JB and BK Rosenzweig. 2016. "Encoding Data Using Biological Principles: the Multisample Variant Format for Phylogenomics and Population Genomics" *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. In press. http://www.dx.doi.org/10.1109/tcbb.2015.2509997

Please also include the URL <https://www.github.com/jbpease/mvftools> in your methods section where the program is referenced.

## 1.3 Installation

No installation is required, mvftools scripts should work as long as Python3 is installed. The repository can be cloned or downloaded as a .zip file from GitHub.

:: git clone https://www.github.com/jbpease/mvftools

Alternatively, you can download MVftools as a .zip file from the github page.

### 1.3.1 Requirements

- Python 3.x (2.7 should also work, but 3.x recommended) https://www.python.org/downloads/
- RAxML 8.x (7.x should also work, but 8.x recommended) https://sco.h-its.org/exelixis/web/software/raxml/index.html

### 1.3.2 Additional Requirements for Some Modules:

- Scipy: (http://www.scipy.org/)
- Biopython 1.6+: (http://www.biopython.org/),
- Numpy (http://www.numpy.org/),
- RAxML (http://sco.h-its.org/exelixis/web/software/raxml/index.html/)

## 1.4 Preparing your data

### 1.4.1 Sequence Alignment

MVF files can be created from VCF, FASTA, and MAF files using the `vcf2mvf.py`, `fasta2mvf.py`, or `maf2mvf.py` programs respectively. Once converted to MVF format, analyses and manipulations can be carried out using the rest of the tools in MVFtools.

## 1.5 Basic usage examples

**Case #1: Generate phylogenies from 100kb windows using a VCF data**:

```
python3 vcf2mvf.py --vcf DATA.vcf --mvf DATA.mvf
python3 mvf_window_tree.py --mvf DATA.mvf --out WINDOWTREES.txt --windowsize␣
↪100000
```

**Case #2: Convert a large FASTA file, then generate window-based counts for DFOIL/D-statistic introgression testing from the first five samples**:

```
python3 fasta2mvf.py --fasta DATA.fasta --mvf DATA.mvf
python3 mvf_analyze_dna.py PatternCount --mvf DATA.mvf --out PATTERNS.txt --
↪windowsize 100000 --samples 0 1 2 3 4
```

The file is now ready to use as an input file for with dfoil (http://www.github.com/jbpease/dfoil).

CHAPTER

TWO

# MVF FORMAT SPECIFICATION (VERSION 1.2)

## 2.1 Version History

### 2.1.1 v1.1.1

Codons and Proteins accommodated

### 2.1.2 v1.2

Dot masking, multi-line header, adoption of "X" in place of "N" for nucleotides, support for non-reference aligned sequences.

## 2.2 MVF General Notes and Usage

### 2.2.1 General Features

MVF is primarily intended for site-wise analyses in phylogenomics and population genomics. MVF is formatted to contain one aligned site per line, but contains only allelic information, therefore MVF most closely mimics VCF files in formatting, but resembles MAF format in informational content, Additionally, MVF uses special formatting to lower file sizes and speed up filtering and analysis. MVF can readily be adapted from other common sequence formats including VCF, FSATA, and MAF. MVF is also designed to be able to accommodate readily store other information for phylogenomic projects, including tree topologies and sample metadata.

### 2.2.2 Native Gzip read/write

MVF is designed to work natively with GZIP compression and uses a formatting that attempts to strike a balance between fast filtering, easy visual inspection, while using character patterns that create a good Gzip compression ratio. As long as any input or output file path ends with exactly ".gz", all MVF scripts will natively read/write to gzip-compressed files.

### 2.2.3 General Notes on Filtering

MVF was specifically designed as a "vertical" format for rapid filtering of *sites* in large-scale phylogenomic analyses. (rather than being "horizontal" to visually show alignment) Therefore, the following should be noted to take advantage of MVF formatting for rapid filtering (i.e. with grep/zgrep).

- `#` is present iff. the line is in the header
- `@` is present iff. the position is non-reference
- `X` is present in the allele string iff. the positon has ambiguity data
- `#:` can quickly filter by chromosome
- `:#` can quickly filter by coordinate numbers
- Allele strings with one or two characters have full sample coverage (no gaps)
- Allele strings with `@[any]+` have coverage=1, `[not@][any]+` have coverage=2
- One or two-character allele strings, or notation with `[any]+` CANNOT contain homoplasy or synapomorphy (by definition).

## 2.3 Header Specification

All header lines begin with one or more `#` and contain single-space separated fields.

### 2.3.1 MVF declaration line

First header line always starts with `##mvf`, followed by required metadata fields:

- version=1.2
- mvftype=[dna, protein, codon]

and optionally:

- an arbitrary number of metadata fields in key=value format ('mvftype' and 'version' not allowed as key)

### 2.3.2 Sample information

Sample information (columns) header lines are specified by:

- line starts with `#s` ("s" for sample) with no leading spaces
- LABEL (must be unique, no spaces)
- an arbitrary number of metadata fields in key=value format ('label' not allowed as key)

The first entry should be the reference sequence (if aligned to reference) or can be any sequence in the case of non-reference-aligned de novo alignment).

### 2.3.3 Contig information

Contig information header lines are specified by:

- line starts with `#c` ("c" for contig)
- CONTIG_ID (must be unique, alpha-numeric strong recommended, must not contain `*:;,@!+` or spaces)
- `label=[NAME]` (recommended by not required to be unique, no spaces allowed)
- `len=[LENGTH]` (integer > 0, or zero for unknown)
- `ref=[0/1]`, indicates if contig is reference-based (=1) or not (=0)
- an arbitrary number of metadata fields in key=value format ("label", "len", and "ref" not allowed as key)

### 2.3.4 Tree information

Tree information may (optionally) be specified in header lines by:

- line starts with `#t` ("t" for tree/topology)
- ‘‘TREE_ID=[###]‘ (must be unique, alpha-numeric)
- `TOPOLOGY=[tree_String]` in Newick/Phylip/parenthetical format (must end with ';')
- an arbitrary number of metadata fields in key=value format

To take full advantage of MVF tree storage, use the same sample labels as in the `#s` header lines

### 2.3.5 Notes

General project notes may (optionally) be specified in the header lines by:

- line starts with `#n` ("n" for notes)
- Text is unstructured and is not necessarily formatted as metadata

### 2.3.6 Example Header

:: ##mvf version=1.2 mvftype=[MVFTYPE] #s SAMPLE0 meta0=somevalue meta1=0 … #s SAMPLE1 meta0=somethingele meta1=1 … #s SAMPLE2 meta0=somesome meta1=0 … … #c 0 label=CONTIG0 length=100 ref=1 meta0=somevalue … #c 1 label=CONTIG1 length=200 ref=0 meta0=someother … … #t 0 ((SAMPLE0,SAMPLE1),SAMPLE2); model=GTRGAMMA software=RAxML #t 1 ((SAMPLE2,SAMPLE0),SAMPLE1); model=GTRGAMMA software=RAxML partition=chrom1 … #n Notes on this project.

## 2.4 Entry Specification

**Note:** all examples show an MVF entry with REF and four samples

Entries are structured as two space-separated columns:

`ID:POSITION ALLELES [ALLELES ALLELES ...]`

- `ID:POSITION` = chromosomal id matching the first element of a contig in the `#c` header element
- `POSITION` = 1-based position on the contig with matching `CONTIG_ID`
- `ALLELES` = one or more records of alleles at reference-based location specified by `ID:POSITION` and matching the formatting below

### 2.4.1 For mvftype=codon

- Allele columns are `PROTEIN DNA1 DNA2 DNA3` where the three DNA columns represent three codon positions in collated form
- Position is the position of the lowest numbered codon position (regardless of transcript strand) and `DNA1/2/3` codon columns are given in order to match the protein (again regardless of transcript orientation)

### 2.4.2 Allele formatting

**Note:** all examples show an MVF entry with five samples.

For reference-anchored contigs, the first allele is assumed to be the "reference" allele by default. Each entry must either (1) contain the same number of characters as sample labels specified in the header or (2) use one of the special cases in the section below.

`ATCTG` = (REF is 'A' samples 1&3 are 'T', sample 2 is 'C', sample 4 is 'G')

### 2.4.3 Special cases

**Note:** all examples show an MVF entry with five samples

### 2.4.4 Invariant sites

When all alleles are both present (non-gap) and all the same, this is represented by a single base.

```
A = AAAAA
```

### 2.4.5 Monoallelic non-reference samples

When all alleles in the samples (non-REF) are the same but differ from REF, this is represented by two bases.

```
AT = ATTTT Aa = Aaaaa
```

### 2.4.6 Single-variant sites

When only one of the samples varies from the others, this is specified as:

```
[reference_base, majority_base, "+", unique_base, unique_position]
```

This is useful shorthand for both sites with one a single base that differs and samples with only one sample represented. When the site only has coverage via one sample (i.e. all other bases are empty, the '-' is omitted from the second position.

```
AC+T2 = ACTCC  AA+C2 = AACAA  -+A2 = --A--  A+A2 = A-A--  A+a2 =
A-a-- A+C2 = A-C--
```

### 2.4.7 Non-reference aligned sites

Added in MVF v.1.2, this facilitates using MVF for non-reference aligned sequences (e.g. aligned sets of orthologs from de novo assembled transcripts). These non-reference-anchored alignments can comprise the entire MVF file or be included in addition to reference-aligned contigs. Non-reference-contigs in their header entry should include the keyword "nonref" (see Section 1.3). Contigs labels and coordinates are labelled the same as reference-based entries. To denote that the sequence is non-reference and not simply a deletion in the reference, the character "@" should be the first character of the alignment. In the case an entirely non-reference MVF, all contigs can be labelled as "nonref," but one sequence should be chosen as the reference for the purposes of the allele string. When this sequence is not present, `@` is still used.

```
@AATT = -AATT @A+T3 = -A-T- @-+A3 = ---A-
```

## 2.5 Character encoding

### 2.5.1 Nucleotide Notation

- Standard IUPAC nucleotide codes are used: `ACGT`, and `U` for uracil in RNA
- Standard IUPAC bialleic ambiguity codes `KMRSWY` are used also.
- Current MVF formatting does NOT allow triallelic ambiguity codes (`BDHV`), which are converted to ambiguous (`X`) instead.
- Current MVF formatting does NOT recognize rare symbols (`ISOX`, or `Phi`)
- Ambiguous nucleotide is denoted by `X` instead of standard `N`

### 2.5.2 Amino Acid Notation

- Standard IUPAC amino acid codes are used: `ACDEFGHIKLMNPQRSTVWY`
- Standard stop codon symbol `*` is used
- Currently the ambiguous/rare symbols are not recognized (`BZ`)

### 2.5.3 Use of x for ambiguous nucleotides and amino acids

In standard notation, "`N`" is used for an ambiguous nucleotide, which could be any of A/C/G/T. However, in amino acid notation `N` stands for "Asparagine" and is a valid character, while `X` is used for an ambiguous amino acid. MVF v1.2 adopts `X` as unified ambiguity character for both nucleotides and proteins for MVF files for two purposes: 1. To creates a unified ambiguity character for MVF codon files for faster processing 2. To allow fast filtering of ambiguous lines Also note that while 'X' in expanded IUPAC notation refers to 'xanthosine,' MVF currently does not support rare nucleotides. .. note:: In all conversion utilities that export from MVF format to another file format conversion to the standard "N"/"X" for ambiguous nucleotides/amino acids should ALWAYS be implemented.

CHAPTER

THREE

# EXAMPLES OF THE SAME DATA IN MVF FORMAT AND OTHER FORMATS

## 3.1 MVF Format

```
##mvf sourceformat=fasta version=1.2 mvftype=dna ncol=5
#s Hsapiens
#s Ptroglodytes
#s Ppaniscus
#s Ggorilla
#s Mmusculus
#c 1 label=Chromosome1 length=248956422
#n Note: This is an example file showing data formatting
1:100 A
1:101 A
1:102 A
1:103 T
1:104 TT+C4
1:105 GC
1:106 A+A4
1:107 AATTA
1:108 AC+G4
```

## 3.2 FASTA Format

```
>Hsapiens gi:1234 geneid:GeneOfInterest chrom:1 start:100 end:108
AAATTGAAA

>Ptroglodytes geneid:GeneOfInterest
AAATTC-AC

>Ppaniscus geneid:GeneOfInterest
AAATTC-TC

>Ggorilla geneid:GeneOfInterest
AAATTC-TC
```

```
>Mmusculus geneid:GeneOfInterest
AAATCCAAG
```

## 3.3 VCF Format

```
##fileformat=VCFv4.1
##samtoolsVersion=0.1.19-44428cd
##reference=hg19.fa
##contig=<ID=Chromosome1,length=248956422>
##INFO=<ID=DP,Number=1,Type=Integer,Description="Raw read depth">
##INFO=<ID=DP4,Number=4,Type=Integer,Description="# high-quality ref-forward␣
↪bases, ref-reverse, alt-forward and alt-reverse bases">
##INFO=<ID=MQ,Number=1,Type=Integer,Description="Root-mean-square mapping␣
↪quality of covering reads">
##INFO=<ID=FQ,Number=1,Type=Float,Description="Phred probability of all␣
↪samples being the same">
##INFO=<ID=AF1,Number=1,Type=Float,Description="Max-likelihood estimate of␣
↪the first ALT allele frequency (assuming HWE)">
##INFO=<ID=AC1,Number=1,Type=Float,Description="Max-likelihood estimate of␣
↪the first ALT allele count (no HWE assumption)">
##INFO=<ID=AN,Number=1,Type=Integer,Description="Total number of alleles in␣
↪called genotypes">
##INFO=<ID=IS,Number=2,Type=Float,Description="Maximum number of reads␣
↪supporting an indel and fraction of indel reads">
##INFO=<ID=AC,Number=A,Type=Integer,Description="Allele count in genotypes␣
↪for each ALT allele, in the same order as listed">
##INFO=<ID=G3,Number=3,Type=Float,Description="ML estimate of genotype␣
↪frequencies">
##INFO=<ID=HWE,Number=1,Type=Float,Description="Chi^2 based HWE test P-value␣
↪based on G3">
##INFO=<ID=CLR,Number=1,Type=Integer,Description="Log ratio of genotype␣
↪likelihoods with and without the constraint">
##INFO=<ID=UGT,Number=1,Type=String,Description="The most probable␣
↪unconstrained genotype configuration in the trio">
##INFO=<ID=CGT,Number=1,Type=String,Description="The most probable␣
↪constrained genotype configuration in the trio">
##INFO=<ID=PV4,Number=4,Type=Float,Description="P-values for strand bias,␣
↪baseQ bias, mapQ bias and tail distance bias">
##INFO=<ID=INDEL,Number=0,Type=Flag,Description="Indicates that the variant␣
↪is an INDEL.">
##INFO=<ID=PC2,Number=2,Type=Integer,Description="Phred probability of the␣
↪nonRef allele frequency in group1 samples being larger (,smaller) than in␣
↪group2.">
##INFO=<ID=PCHI2,Number=1,Type=Float,Description="Posterior weighted chi^2 P-
↪value for testing the association between group1 and group2 samples.">
##INFO=<ID=QCHI2,Number=1,Type=Integer,Description="Phred scaled PCHI2.">
##INFO=<ID=PR,Number=1,Type=Integer,Description="# permutations yielding a␣
↪smaller PCHI2.">
##INFO=<ID=QBD,Number=1,Type=Float,Description="Quality by Depth: QUAL/#reads
↪">
```

```
##INFO=<ID=RPB,Number=1,Type=Float,Description="Read Position Bias">
##INFO=<ID=MDV,Number=1,Type=Integer,Description="Maximum number of high-
↪quality nonRef reads in samples">
##INFO=<ID=VDB,Number=1,Type=Float,Description="Variant Distance Bias (v2)␣
↪for filtering splice-site artefacts in RNA-seq data. Note: this version may␣
↪be broken.">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=GL,Number=3,Type=Float,Description="Likelihoods for RR,RA,AA␣
↪genotypes (R=ref,A=alt)">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="# high-quality bases">
##FORMAT=<ID=DV,Number=1,Type=Integer,Description="# high-quality non-
↪reference bases">
##FORMAT=<ID=SP,Number=1,Type=Integer,Description="Phred-scaled strand bias P-
↪value">
##FORMAT=<ID=PL,Number=G,Type=Integer,Description="List of Phred-scaled␣
↪genotype likelihoods">
#CHROM        POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT ␣
↪Ptroglodytes Ppaniscus  Ggorilla        Mmusculus
ch01  100     .       A       .       30      .       DP=5;AF1=0;AC1=0;DP4=5,
↪0,0,0;MQ=20;FQ=-23.4     PL:DP   0/0:0,6,40:2:4  0/0:0,6,40:2:4  0/0:0,6,
↪40:2:4  0/0:0,6,40:2:4
ch01  101     .       A       .       30      .       DP=5;AF1=0;AC1=0;DP4=5,
↪0,0,0;MQ=20;FQ=-23.4     PL:DP   0/0:0,6,40:2:4  0/0:0,6,40:2:4  0/0:0,6,
↪40:2:4  0/0:0,6,40:2:4
ch01  102     .       A       .       30      .       DP=5;AF1=0;AC1=0;DP4=5,
↪0,0,0;MQ=20;FQ=-23.4     PL:DP   0/0:0,6,40:2:4  0/0:0,6,40:2:4  0/0:0,6,
↪40:2:4  0/0:0,6,40:2:4
ch01  103     .       T       .       32      .       DP=5;AF1=0;AC1=0;DP4=5,
↪0,0,0;MQ=20;FQ=-23.4     PL:DP   0/0:0,6,40:2:4  0/0:0,6,40:2:4  0/0:0,6,
↪40:2:4  0/0:0,6,40:2:4
ch01  104     .       T       C       7.61    .       DP=2;VDB=6.720000e-02;
↪AF1=1;AC1=58;DP4=0,0,1,1;MQ=20;FQ=-23.8   GT:PL:DP:GQ     0/0:0,6,40:2:4  0/
↪0:0,6,40:2:4  0/0:0,6,40:2:4  1/1:38,6,0:2:4
ch01  105     .       G       C       32.1    .       DP=5;AF1=0;AC1=0;DP4=5,
↪0,0,0;MQ=20;FQ=-23.4     PL:DP   0/0:0,6,40:2:4  0/0:0,6,40:2:4  0/0:0,6,
↪40:2:4  1/1:38,6,0:2:4
ch01  106     .       A       .       30      .       DP=5;AF1=0;AC1=0;DP4=5,
↪0,0,0;MQ=20;FQ=-23.4     PL:DP   0:0     0:0     0:0     0/0:0,6,40:2:4
ch01  107     .       A       T       24.4    .       DP=5;AF1=1;AC1=58;DP4=0,
↪0,1,0;MQ=20;FQ=-23.4    PL:DP   0/0:0,6,40:2:4  1/1:38,6,0:2:4  1/1:38,6,
↪0:2:4  0/0:0,6,40:2:4
ch01  108     .       A       C,G     999     .       DP=52;VDB=6.361343e-02;
↪RPB=-1.264051e+00;AF1=0.9325;AC1=54;DP4=0,2,20,26;MQ=20;FQ=-16.1;PV4=0.5,1,
↪1,1   GT:PL:DP:GQ     1/1:20,3,0,20,3,20:1:11 1/1:36,6,0,36,6,36:2:13 1/
↪1:36,6,0,36,6,36:2:13 1/1:95,95,95,18,18,0:6:8
```

CHAPTER

FOUR

# PROGRAM PARAMETER DESCRIPTIONS

## 4.1 fasta2mvf

### 4.1.1 Description

This program is used to convert a FASTA file into MVF format.

### 4.1.2 Parameters

#### `-h/--help`

**Description:** show this help message and exit

**Type:** boolean flag

#### `-i/--fasta` (required)

**Description:** input FASTA file(s)

**Type:** None; **Default:** None

#### `-o/--out` (required)

**Description:** output MVF file

**Type:** None; **Default:** None

#### `-c/--contig-ids/--contigids`

**Description:** manually specify one or more contig ids as ID:NAME

**Type:** None; **Default:** None

#### --contig-by-file/--contigbyfile

**Description:** Contigs are designated by separate files.

**Type:** boolean flag

#### --contig-field/--contigfield

**Description:** When headers are split by –field-sep, the 0-based index of the contig id.

**Type:** integer; **Default:** None

#### -f/--flavor

**Description:** type of file [dna] or protein

**Type:** None; **Default:** dna

**Choices:** ['dna', 'protein']

#### --manual-coord/--manualcoord

**Description:** manually specify reference coordinates for each file in the format CONTIGID:START..STOP, ...

**Type:** None; **Default:** None

#### --overwrite

**Description:** USE WITH CAUTION: force overwrite of outputs

**Type:** boolean flag

#### -s/--sample-replace/--samplereplace

**Description:** one or more TAG:NEWLABEL or TAG, items, if TAG found in sample label, replace with NEW (or TAG if NEW not specified) NEW and TAG must each be unique

**Type:** None; **Default:** None

#### --sample-field/--samplefield

**Description:** when headers are split by –field-sep, the 0-based index of the sample id

**Type:** integer; **Default:** None

#### -B/--read-buffer/--readbuffer

**Description:** number of lines to hold in READ buffer

**Type:** integer; **Default:** 100000

#### -F/--field-sep/--fieldsep

**Description:** FASTA field separator; assumes '>database accession locus' format

**Type:** None; **Default:** None

**Choices:** ['TAB', 'SPACE', 'DBLSPACE', 'COMMA', 'MIXED', 'PIPE', 'AT', 'UNDER', 'DBLUNDER']

#### -R/--ref-label/--reflabel

**Description:** label for reference sample

**Type:** None; **Default:** REF

#### -W/--write-buffer/--writebuffer

**Description:** number of lines to hold in WRITE buffer

**Type:** integer; **Default:** 100000

## 4.2 maf2mvf

### 4.2.1 Description

This program analyzes a DNA MVF alignment using the modules specified below, use the –morehelp option for additional module information.

### 4.2.2 Parameters

#### -h/--help

**Description:** show this help message and exit

**Type:** boolean flag

#### -i/--maf (required)

**Description:** input MAF file

**Type:** file path; **Default:** None

#### `-o/--out` (required)

**Description:** output MVF file

**Type:** file path; **Default:** None

#### `-s/--sample-tags/--sampletags` (required)

**Description:** one or more TAG:NEWLABEL or TAG, items, if TAG found in sample label, replace with NEW (or TAG if NEW not specified) NEW and TAG must each be unique.

**Type:** None; **Default:** None

#### `--overwrite`

**Description:** None

**Type:** boolean flag

#### `-B/--line-buffer/--linebuffer`

**Description:** number of lines to hold in read/write buffer

**Type:** integer; **Default:** 100000

#### `-M/--mvf-ref-label/--mvfreflabel`

**Description:** new label for reference sample (default='REF')

**Type:** None; **Default:** REF

#### `-R/--ref-tag/--reftag`

**Description:** old reference tag

**Type:** None; **Default:** None

## 4.3 mvf2dump

### 4.3.1 Description

This program exports the entirety of an MVF to FASTA format, with many fewer options than mvf2fasta.py. This is designed to export large MVF files faster, but with less specific formatting and region-finding options.

### 4.3.2 Parameters

#### `-h/--help`

**Description:** show this help message and exit

**Type:** boolean flag

#### `-i/--mvf` (required)

**Description:** Input MVF file.

**Type:** file path; **Default:** None

#### `-o/--outprefix` (required)

**Description:** Target FASTA file

**Type:** file path; **Default:** None

#### `-d/--outdata`

**Description:** output dna, rna or prot data

**Type:** None; **Default:** None

**Choices:** ('dna', 'rna', 'prot')

#### `--quiet`

**Description:** suppress screen output

**Type:** boolean flag

#### `-s/--samples`

**Description:** One or more taxon labels, leave blank for all

**Type:** None; **Default:** None

#### `-t/--tmpdir`

**Description:** directory to write temporary fasta files

**Type:** file path; **Default:** .

#### -B/--buffer

**Description:** size (Mbp) of write buffer for each sample

**Type:** integer; **Default:** 10

## 4.4 mvf2fasta

### 4.4.1 Description

This program takes an MVF file and converts the data to a FASTA file

### 4.4.2 Parameters

#### -h/--help

**Description:** show this help message and exit

**Type:** boolean flag

#### -i/--mvf (required)

**Description:** Input MVF file.

**Type:** file path; **Default:** None

#### -o/--out (required)

**Description:** target FASTA file

**Type:** file path; **Default:** None

#### -r/--regions (required)

**Description:** A file path to a plain-text file withone region per line formatted asformatted as: contigid,start,stop(coordinates are inclusive)

**Type:** None; **Default:** None

#### -d/--outdata

**Description:** Output dna, rna or prot data.

**Type:** None; **Default:** None

**Choices:** ('dna', 'rna', 'prot')

#### -l/--labeltype

**Description:** Long labels with all metadata or short ids

**Type:** None; **Default:** long

**Choices:** ('long', 'short')

#### --quiet

**Description:** suppress screen output

**Type:** boolean flag

#### -s/--samples

**Description:** One or more taxon labels, leave blank for all

**Type:** None; **Default:** None

#### -t/--tmpdir

**Description:** directory to write temporary fasta files

**Type:** None; **Default:** .

#### -B/--buffer

**Description:** size (Mbp) of write buffer for each sample

**Type:** integer; **Default:** 10

## 4.5 mvf2phy

### 4.5.1 Description

This program is used to export MVF data to Phylip format.

### 4.5.2 Parameters

#### -h/--help

**Description:** show this help message and exit

**Type:** boolean flag

#### -i/--mvf (required)

**Description:** Input MVF file.

**Type:** file path; **Default:** None

#### -o/--out (required)

**Description:** Output Phylip file.

**Type:** file path; **Default:** None

#### -d/--outdata

**Description:** Output dna, rna or prot data.

**Type:** None; **Default:** None

**Choices:** ('dna', 'rna', 'prot')

#### -p/--partition

**Description:** Output a CSV partitions file with RAxMLformatting for use in partitioned phylogenetic methods.

**Type:** boolean flag

#### --quiet

**Description:** suppress screen output

**Type:** boolean flag

#### -r/--region

**Description:** Path of a plain text file containing one more lines with entries 'contigid,stop,start' (one per line, inclusive coordinates) all data will be returned if left blank.

**Type:** file path; **Default:** None

#### -s/--samples

**Description:** One or more taxon labels, leave blank for all.

**Type:** None; **Default:** None

#### -t/--tmpdir

**Description:** directory to write temporary fasta files

**Type:** None; **Default:** .

#### -B/--buffer

**Description:** size (bp) of write buffer for each sample

**Type:** integer; **Default:** 100000

#### -L/--labeltype

**Description:** Long labels with all metadata or short ids

**Type:** None; **Default:** short

**Choices:** ('long', 'short')

## 4.6 mvf_analyze_codon

### 4.6.1 Description

This program analyzes a codon MVF using several analysis modules. Run 'python3 mvf_analyze_codon.py –morehelp' for details on module functions.

### 4.6.2 Parameters

#### module

**Description:** None

**Type:** None; **Default:** None

**Choices:** ('Coverage', 'GroupUniqueAlleleWindow', 'PiDiversityWindow', 'PairwiseNS')

#### -h/--help

**Description:** show this help message and exit

**Type:** boolean flag

#### --all-sample-trees/--allsampletrees

**Description:** (GroupUniqueAlleleWindow) Makes trees from all samples instead of only the most complete sequence from each species

**Type:** boolean flag

#### --allele-groups/--allelegroups

**Description: GROUP1:LABEL,LABEL GROUP2:LABEL,LABEL** (GroupUniqueAlleleWindow)

**Type:** None; **Default:** None

#### --branchlrt

**Description:** (GroupUniqueAlleleWindow) Specify the output file for and turn on the RAxML-PAML format LRT test scan for selection on the target branch in addition to the basic patterns scan

**Type:** file path; **Default:** None

#### -c/--contigs

**Description:** List of space-separated contig ids.

**Type:** None; **Default:** None

#### -g/--gff

**Description:** GFF3 file for use in annotation

**Type:** None; **Default:** None

#### -i/--mvf

**Description:** Input MVF file.

**Type:** file path; **Default:** None

#### -m/--mincoverage

**Description:** Minimum number of samples with alleles needed to use site for analysis.

**Type:** integer; **Default:** None

#### --morehelp

**Description:** Get additional information on modules.

**Type:** boolean flag

#### --num-target-species/--targetspec

**Description:** (GroupUniqueAlleleWindow) Specify the minimum number of taxa in the target set that are required to conduct analysis

**Type:** integer; **Default:** 1

#### -o/--out

**Description:** output file

**Type:** file path; **Default:** None

#### --output-align/--outputalign

**Description:** (GroupUniqueAlleleWindow) Output alignment to this file path in phylip format.

**Type:** None; **Default:** None

#### --pamltmp

**Description:** path for temporary folder for PAML output files

**Type:** file path; **Default:** pamltmp

#### -s/--samples

**Description:** List of space-separated sample names.

**Type:** None; **Default:** None

#### --species-groups/--speciesgroups

**Description:** None

**Type:** None; **Default:** None

#### --target

**Description:** (GroupUniqueAlleleWindow) Specify the taxa labels that define the target lineage-specific branch to be tested.

**Type:** None; **Default:** None

#### -w/--windowsize

**Description:** Window size in bp, use -1 for whole contig.

**Type:** integer; **Default:** -1

#### -x/--chi-test/--chitest

**Description:** (GroupUniqueAlleleWindow,PairwiseDNDS)Input two number values for expected Nonsynonymous and Synonymous expected values.

**Type:** None; **Default:** None

#### -E/--endcontig

**Description:** Numerical id for the ending contig.

**Type:** integer; **Default:** 100000000

#### -L/--uselabels

**Description:** Use contig labels instead of IDs in output.

**Type:** boolean flag

#### -O/--outgroup

**Description:** (GroupUniqueAlleleWindow) Specify sample name with which to root trees.

**Type:** None; **Default:** None

#### -P/--codemlpath

**Description:** Full path for PAML codeml executable.

**Type:** file path; **Default:** codeml

#### -S/--startcontig

**Description:** Numerical ID for the starting contig.

**Type:** integer; **Default:** 0

#### -X/--raxmlpath

**Description:** Full path to RAxML program executable.

**Type:** file path; **Default:** raxml

## 4.7 mvf_analyze_dna

### 4.7.1 Description

This program analyzes a DNA MVF alignment using the modules specified below, use the –morehelp option for additional module information.

### 4.7.2 Parameters

#### module

**Description:** analysis module to run

**Type:** None; **Default:** None

**Choices:** ('BaseCountWindow', 'Coverage', 'DstatComb', 'PairwiseDistance', 'PairwiseDistanceWindow', 'PatternCount', 'PatternList')

#### -h/--help

**Description:** show this help message and exit

**Type:** boolean flag

#### -i/--mvf (required)

**Description:** Input MVF file.

**Type:** file path; **Default:** None

#### -o/--out (required)

**Description:** output file

**Type:** file path; **Default:** None

**--base-match**

**Description:** [BaseCountWindow] string of bases to match (i.e. numerator).

**Type:** None; **Default:** None

**--base-total**

**Description:** [BaseCountWindow] string of bases for total (i.e. denominator).

**Type:** None; **Default:** None

**-c/--contigs**

**Description:** limit analyses to these contigs

**Type:** None; **Default:** None

**-m/--mincoverage**

**Description:** mininum sample coverage for site

**Type:** integer; **Default:** None

**--morehelp**

**Description:** get additional information on modules

**Type:** boolean flag

**-s/--samples**

**Description:** limit analyses to these samples

**Type:** None; **Default:** None

**-w/--windowsize**

**Description:** window size, use -1 to use whole contigs

**Type:** integer; **Default:** 100000

## 4.8 mvf_annotate

### 4.8.1 Description

This program takes a DNA MVF alignment and annotates the output into gene boudaries.

### 4.8.2 Parameters

#### `-h/--help`

**Description:** show this help message and exit

**Type:** boolean flag

#### `-g/--gff`

**Description:** Input gff annotation file.

**Type:** file path; **Default:** None

#### `-i/--mvf`

**Description:** Input MVF file.

**Type:** file path; **Default:** None

#### `-o/--out`

**Description:** Output annotated MVF file

**Type:** file path; **Default:** None

#### `--overwrite`

**Description:** USE WITH CAUTION: force overwrite of outputs

**Type:** boolean flag

#### `--quiet`

**Description:** suppress progress meter

**Type:** boolean flag

#### -B/--linebuffer

**Description:** Number of entries to store in memory at a time.

**Type:** integer; **Default:** 100000

#### -F/--filter_annotation

**Description:** Skip entries in the GFF file that contain this string in their 'Notes'

**Type:** None; **Default:** None

#### -M/--nongenic-margin

**Description:** for –unnanotated-mode, only retain positions that are this number of bp away from an annotated region boundary

**Type:** integer; **Default:** 0

#### -N/--nongenic-mode

**Description:** Instead of returning annotated genes, return the non-genic regions without without changing contigs or coordinates

**Type:** boolean flag

## 4.9 mvf_check

### 4.9.1 Description

This program checks an MVF file for inconsistencies or errors

### 4.9.2 Parameters

#### mvf

**Description:** Input MVF file.

**Type:** file path; **Default:** None

#### -h/--help

**Description:** show this help message and exit

**Type:** boolean flag

## 4.10 mvf_chromoplot

### 4.10.1 Description

This program creates a chromoplot from an MVF alignment. A chromoplot shows a genome-wide diagram of different evolutionary histories for a given quartet of taxa.

### 4.10.2 Parameters

#### `-h/--help`

**Description:** show this help message and exit

**Type:** boolean flag

#### `-i/--mvf` (required)

**Description:** Input MVF file.

**Type:** file path; **Default:** None

#### `-s/--samples` (required)

**Description:** 3 or more taxa to use for quartets

**Type:** None; **Default:** None

#### `-G/--outgroup` (required)

**Description:** 1 or more outgroups to use for quartets

**Type:** None; **Default:** None

#### `-c/--contigs`

**Description:** Enter the ids of one or more contigs in the order they will appear in the chromoplot. (defaults to all ids in order present in MVF)

**Type:** None; **Default:** None

#### `-o/--outprefix`

**Description:** Output prefix (not required).

**Type:** None; **Default:** None

#### -q/--quiet

**Description:** suppress all output messages

**Type:** boolean flag

#### -w/--windowsize

**Description:** None

**Type:** integer; **Default:** 100000

#### -x/--xscale

**Description:** Width (in number of pixels) for each window

**Type:** integer; **Default:** 1

#### -y/--yscale

**Description:** Height (in number of pixels) for each track

**Type:** integer; **Default:** 20

#### -C/--colors

**Description:** three colors to use for chromoplot

**Type:** None; **Default:** None

**Choices:** {'lgrey': (250, 250, 250), 'dgrey': (192, 192, 192), 'black': (0, 0, 0), 'white': (255, 255, 255), 'red': (192, 0, 0), 'orange': (217, 95, 2), 'yellow': (192, 192, 0), 'green': (0, 192, 0), 'blue': (0, 0, 192), 'teal': (27, 158, 119), 'puce': (117, 112, 179), 'purple': (192, 0, 192), 'none': ()}

#### -E/--emptymask

**Description:** Mask empty regions with this color.

**Type:** None; **Default:** none

**Choices:** {'lgrey': (250, 250, 250), 'dgrey': (192, 192, 192), 'black': (0, 0, 0), 'white': (255, 255, 255), 'red': (192, 0, 0), 'orange': (217, 95, 2), 'yellow': (192, 192, 0), 'green': (0, 192, 0), 'blue': (0, 0, 192), 'teal': (27, 158, 119), 'puce': (117, 112, 179), 'purple': (192, 0, 192), 'none': ()}

#### -I/--infotrack

**Description:** Include an additional coverage information track that will show empty, uninformative, and informative loci. (Useful for ranscriptomes/RAD or other reduced sampling.

**Type:** boolean flag

#### -M/--majority

**Description:** Plot only 100% shading in the majority track rather than shaded proportions in all tracks.

**Type:** boolean flag

#### -P/--plottype

**Description:** PNG image (default) or graph via matplotlib (experimental)

**Type:** None; **Default:** image

**Choices:** ['graph', 'image']

## 4.11 mvf_filter

### 4.11.1 Description

This program filters an MVF alignment using the modules specified below, use the –morehelp option for additional module information.

### 4.11.2 Parameters

#### -h/--help

**Description:** show this help message and exit

**Type:** boolean flag

#### -a/--actions

**Description:** set of actions:args to perform, note these are done in order as listed

**Type:** None; **Default:** None

#### -i/--mvf

**Description:** Input MVF file.

**Type:** file path; **Default:** None

#### -l/--labels

**Description:** use sample labels instead of indices

**Type:** boolean flag

#### --morehelp

**Description:** prints full module list and descriptions

**Type:** boolean flag

#### -o/--out

**Description:** Output MVF file

**Type:** file path; **Default:** None

#### --overwrite

**Description:** USE WITH CAUTION: force overwrite of outputs

**Type:** boolean flag

#### -q/--quiet

**Description:** suppress progress meter

**Type:** boolean flag

#### --test

**Description:** manually input a line for testing

**Type:** None; **Default:** None

#### --test-nchar

**Description:** total number of samples for test string

**Type:** integer; **Default:** None

#### -B/--linebuffer

**Description:** number of lines to write at once to MVF

**Type:** integer; **Default:** 100000

#### `-V/--verbose`

**Description:** report every line (for debugging)

**Type:** boolean flag

## 4.12 mvf_join

### 4.12.1 Description

This program checks an MVF file for inconsistencies or errors

### 4.12.2 Parameters

#### `-h/--help`

**Description:** show this help message and exit

**Type:** boolean flag

#### `-i/--mvf` (required)

**Description:** One or more mvf files.

**Type:** file path; **Default:** None

#### `-o--out` (required)

**Description:** Output mvf file.

**Type:** file path; **Default:** None

#### `-c/--newcontigs`

**Description:** By default, contigs are matched between files using their text labels in the header. Use this option to turn matching off and treat each file's contigs as distinct.

**Type:** boolean flag

#### `--overwrite`

**Description:** USE WITH CAUTION: force overwrite of outputs

**Type:** boolean flag

#### `--quiet`

**Description:** suppress progress meter

**Type:** boolean flag

#### `-s/--newsamples`

**Description:** By default, samples are matched between files using their text labels in the header. Use this option to turn matching off and treat each file's sample columns as distinct.

**Type:** boolean flag

#### `-B/--linebuffer`

**Description:** number of entries to write in a block

**Type:** integer; **Default:** 100000

#### `-M/--main_header_file`

**Description:** Output file will use same headers as this input file (default=first in list).

**Type:** None; **Default:** None

## 4.13 mvf_translate

### 4.13.1 Description

This program translates a DNA MVF file into a codon or protein MVF file using a GFF3 annotation file.

### 4.13.2 Parameters

#### `-h/--help`

**Description:** show this help message and exit

**Type:** boolean flag

#### `-i/--mvf` (required)

**Description:** Input MAF file

**Type:** file path; **Default:** None

#### -o/--out (required)

**Description:** Output MVF file

**Type:** None; **Default:** None

#### -g/--gff

**Description:** Input GFF3 file. If GFF3 not provided, alignments are assumed to be in-frame coding sequences.

**Type:** file path; **Default:** None

#### --overwrite

**Description:** USE WITH CAUTION: force overwrite of outputs

**Type:** boolean flag

#### --quiet

**Description:** suppress progress meter

**Type:** boolean flag

#### -t--outtype

**Description:** protein=single data column of protein alleles; codon=four columns with: protein frame1 frame2 frame3

**Type:** None; **Default:** codon

**Choices:** ['protein', 'codon']

#### -B/--line-buffer/--linebuffer

**Description:** number of entries to write in a block

**Type:** integer; **Default:** 100000

#### -F/--filter-annotation

**Description:** skip GFF entries with text matching this in their 'Notes' field

**Type:** None; **Default:** None

## 4.14 mvf_window_tree

### 4.14.1 Description

This program makes phylogenies from individual genomic windows of a DNA MVF alignment (Requires: BioPython).

### 4.14.2 Parameters

#### `-h/--help`

**Description:** show this help message and exit

**Type:** boolean flag

#### `-i/--mvf` (required)

**Description:** Input MVF file.

**Type:** file path; **Default:** None

#### `-o/--out` (required)

**Description:** Tree list output text file.

**Type:** file path; **Default:** None

#### `-b/--bootstrap`

**Description:** turn on rapid bootstrapping for RAxML and perform specified number of replicates

**Type:** integer; **Default:** None

#### `-c/--contigs`

**Description:** Contig ids to use in analysis (default=all)

**Type:** None; **Default:** None

#### `-d/--duplicateseq`

**Description:** dontuse=remove duplicate sequences prior to RAxML tree inference, then add them to the tree manually as zero-branch-length sister taxa; keep=keep in for RAxML tree inference (may cause errors for RAxML); remove=remove entirely from alignment

**Type:** None; **Default:** dontuse

**Choices:** ['dontuse', 'keep', 'remove']

#### -e/--outputempty

**Description:** Include entries of windows with no data in output.

**Type:** boolean flag

#### -g/--raxml-outgroups/--raxml_outgroups

**Description:** Outgroups taxon labels to use in RAxML.

**Type:** None; **Default:** None

#### -m/--raxml_model

**Description:** choose RAxML model

**Type:** None; **Default:** GTRGAMMA

#### --outputcontiglabels

**Description:** Output will use contig labels instead of id numbers.

**Type:** boolean flag

#### --quiet

**Description:** suppress screen output

**Type:** boolean flag

#### -r/--rootwith

**Description:** Root output trees with these taxa after RAxML.

**Type:** None; **Default:** None

#### -s/--samples

**Description:** One or more taxon labels (default=all)

**Type:** None; **Default:** None

#### --tempdir

**Description:** Temporary directory path

**Type:** file path; **Default:** ./raxmltemp

#### --tempprefix

**Description:** Temporary file prefix

**Type:** None; **Default:** mvftree

#### -w/--windowsize

**Description:** specify genomic region size, or use -1 for whole contig

**Type:** integer; **Default:** 10000

#### -A/--choose_allele/--hapmode

**Description:** Chooses how heterozygous alleles are handled. (none=no splitting (default); randomone=pick one allele randomly (recommended); randomboth=pick two alleles randomly, but keep both; major=pick the more common allele; minor=pick the less common allele; majorminor= pick the major in 'a' and minor in 'b'

**Type:** None; **Default:** none

**Choices:** ['none', 'randomone', 'randomboth', 'major', 'minor', 'majorminor']

#### -C/--minseqcoverage

**Description:** **proportion of total alignment a sequence** must cover to be retianed [0.1]

**Type:** float; **Default:** 0.1

#### -D/--mindepth

**Description:** minimum number of alleles per site

**Type:** integer; **Default:** 4

#### -M/--minsites

**Description:** minimum number of sites

**Type:** integer; **Default:** 100

#### -R/--raxmlopts

**Description:** specify additional RAxML arguments as a double-quotes encased string

**Type:** None; **Default:**

#### -X/--raxmlpath

**Description:** RAxML path for manual specification.

**Type:** None; **Default:** raxml

## 4.15 vcf2mvf

### 4.15.1 Description

MVFtools: Multisample Variant Format Toolkit James B. Pease and Ben K. Rosenzweig http://www.github.org/jbpease/mvftools

### 4.15.2 Parameters

#### -h/--help

**Description:** show this help message and exit

**Type:** boolean flag

#### --out (required)

**Description:** output MVF file

**Type:** None; **Default:** None

#### --vcf (required)

**Description:** input VCF file

**Type:** file path; **Default:** None

#### --allelesfrom

**Description: get additional alignment columns** from INFO fields (:-separated)

**Type:** None; **Default:** None

**--contigids**

**Description: manually specify one or more contig ids** as ID;VCFLABE;MVFLABEL, note that VCFLABEL must match EXACTLY the contig string labels in the VCF file

**Type:** None; **Default:** None

**--fieldsep**

**Description:** VCF field separator (default='TAB')

**Type:** None; **Default:** TAB

**Choices:** ['TAB', 'SPACE', 'DBLSPACE', 'COMMA', 'MIXED']

**--linebuffer**

**Description:** number of lines to hold in read/write buffer

**Type:** integer; **Default:** 100000

**--lowdepth**

**Description:** below this read depth coverage, convert to lower case set to 0 to disable

**Type:** integer; **Default:** 3

**--lowqual**

**Description: below this quality convert to lower case** set to 0 to disable

**Type:** integer; **Default:** 20

**--maskdepth**

**Description:** below this read depth mask with N/n

**Type:** integer; **Default:** 1

**--maskqual**

**Description: low quality cutoff, bases replaced by N/-** set to 0 to disable

**Type:** integer; **Default:** 3

#### --no_autoindex

**Description:** do not automatically index contigs from the VCF

**Type:** boolean flag

#### --outflavor

**Description:** choose output MVF flavor to include quality scores and/or indels

**Type:** None; **Default:** dna

**Choices:** ['dna', 'dnaqual', 'dnaqual-indel', 'dna-indel']

#### --overwrite

**Description:** USE WITH CAUTION: force overwrite of outputs

**Type:** boolean flag

#### --qual

**Description:** Include Phred genotype quality (GQ) scores

**Type:** boolean flag

#### --reflabel

**Description:** label for reference sample (default='REF')

**Type:** None; **Default:** REF

#### --samplereplace

**Description: one or more TAG:NEWLABEL or TAG, items,** if TAG found in sample label, replace with NEW (or TAG if NEW not specified) NEW and TAG must each be unique

**Type:** None; **Default:** None

CHAPTER

FIVE

# FREQUENTLY ASKED QUESTIONS

See also our forum at: https://groups.google.com/forum/#!forum/mvftools

Coming soon.

CHAPTER

SIX

# VERSION HISTORY

**2017-06-25**

*Major Upgrade*: Full manual documentation added, standardization and cleanup of paramaters and upgrades and bugfixes throughout.

**2017-05-18**

Fixes to VCF conversion for compatibility

**2017-04-10**

Added MVF-to-Phylip output conversion `mvf2phy`

**2017-03-25**

Multiple bug fixes, merged and removed the development instance

**2016-02-15**

Fix to vcf2mvf for VCF with truncated entries

**2016-10-25**

Efficiency upgrades for mvfbase entry iteration.

**2016-09-10**

Minor fixes to gz reading and MVF chromoplot shading

**2016-08-02**

Python3 conversion, integrate analysis_base

**2016-01-11**

fix for dna ambiguity characters

**2016-01-01**

Python3 compatiblity fix

**2015-12-31**

Header changesand cleanup

**2015-12-15**

Python3 compatibilty fix

**2015-09-04**

Small style fixes

**2015-06-09**

MVF1.2.1 upgrade

**2015-02-26**

Efficiency upgrades for iterators

**2015-02-01**

First Public Release

CHAPTER

SEVEN

# LICENSE

MVFtools is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. MVFtools is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with MVFtools. If not, see <http://www.gnu.org/licenses/>.

CHAPTER

EIGHT

# INDICES AND TABLES

- genindex
- modindex
- search