# Package ‘VLF’

April 10, 2025

**Type** Package

**Title** Frequency Matrix Approach for Assessing Very Low Frequency Variants in Sequence Records

**Version** 1.1-3

**Date** 2025-04-010

**Maintainer** Taryn B. T. Athey <taryn.athey@gmail.com>

**Description** Using frequency matrices, very low frequency variants (VLFs) are assessed for amino acid and nucleotide sequences. The VLFs are then compared to see if they occur in only one member of a species, singleton VLFs, or if they occur in multiple members of a species, shared VLFs. The amino acid and nucleotide VLFs are then compared to see if they are concordant with one another. Amino acid VLFs are also assessed to determine if they lead to a change in amino acid residue type, and potential changes to protein structures. Based on Stoeckle and Kerr (2012) <doi:10.1371/journal.pone.0043992> and Phillips et al. (2023) <doi:10.3897/BDJ.

**License** GPL (>= 3)

**Depends** R (>= 2.10)

**NeedsCompilation** no

**Repository** CRAN

**Author** Taryn B. T. Athey [cre],
Paul D. McNicholas [aut],
Jarrett D. Phillips [ctb]

**Date/Publication** 2025-04-10 19:20:36 UTC

## Contents

VLF-package . . . . . 3
aa.compare . . . . . 3
aa.conservation_first . . . . . 4
aa.conservation_two . . . . . 5
aa.count.function . . . . . 6
aa.find.singles . . . . . 7
aa.frequency.matrix.function . . . . . 8
aa.MODE . . . . . 9

aa.MODE.freq . . . 10
aa.MODE.second.freq . . . 10
aa.specimen.frequencies . . . 11
aa.VLF.convert.matrix . . . 12
aa.VLF.count.pos . . . 13
aa.VLF.count.spec . . . 14
aa.VLF.reduced . . . 15
aaVLFs.to.modalchanges . . . 16
aminoAcid.matching.positions . . . 17
aminoAcidFun . . . 18
birds . . . 20
birds_aminoAcids . . . 20
compare . . . 21
concordanceFun . . . 22
concordant.to.modalchanges . . . 23
conservation_first . . . 25
conservation_two . . . 26
count.function . . . 27
Decile.Plot . . . 28
Error.Rate . . . 29
fasta.read . . . 30
ffrequency.matrix.function . . . 30
find.matching . . . 31
find.singles . . . 33
matched.codon.position . . . 34
MODE . . . 35
MODE.freq . . . 36
MODE.second.freq . . . 37
nucleotide.matching.positions . . . 38
overall.matched . . . 39
separate . . . 41
Sliding.Window . . . 42
specimen.frequencies . . . 43
VLF.aminoAcids . . . 44
VLF.convert.matrix . . . 45
VLF.count.pos . . . 46
VLF.count.spec . . . 47
VLF.nucleotides . . . 48
VLF.reduced . . . 49
vlfFun . . . 50

**Index** **52**

## VLF-package — *Frequency Matrix Approach for Assessing Very Low Frequency Variants in Sequence Records*

### Description

Using frequency matrices, very low frequency variants (VLFs) are assessed for amino acid and nucleotide sequences. The VLFs are then compared to see if they occur in only one member of a species, singleton VLFs, or if they occur in multiple members of a species, shared VLFs. The amino acid and nucleotide VLFs are then compared to see if they are concordant with one another. Amino acid VLFs are also assessed to determine if they lead to a change in amino acid residue type, and potential changes to protein structures.

### Details

vlfFun() aminoAcidFun() concordanceFun()

### Author(s)

Taryn B. T. Athey and Paul D. McNicholas

Maintainer: Taryn B. T. Athey <tathey@uoguelph.ca>

### Examples

```
## Not run: #VLF analysis
data(birds)
bird_vlfAnalysis <- vlfFun(birds)

#Amino Acid analysis
data(birds_aminoAcids)
bird_aaAnalysis <- aminoAcidFun(birds_aminoAcids)

#Concordance analysis
nuc_matrix <- bird_vlfAnalysis$VLFmatrix
aa_matrix <- bird_aaAnalysis$VLFmatrix
aa_modal <- bird_aaAnalysis$modal
bird_Concordance <- concordanceFun(nuc_matrix, aa_matrix, 648, 216, aa_modal)
## End(Not run)
```

## aa.compare — *Amino Acid Comparison*

### Description

Compares amino acid very low frequency variants between specimen of the same species

### Usage

```
aa.compare(x, seqlength)
```

### Arguments

| | |
|---|---|
| x | A list of amino acid sequences separated by species name. |
| seqlength | The length of the amino acid sequences within the list. |

### Details

The argument x can be calculated using the separate function.

### Value

A matrix containing two vectors, one with singleton VLF counts for each position of the sequence, and one with shared VLF counts for each position of the sequence.

### Author(s)

Taryn B.T. Athey and Paul D. McNicholas

### Examples

```
## Not run: data(birds_aminoAcids)
birds_aminoAcid_speciesNames <- birds_aminoAcids[,2]
aminoAcids_specimenNumber <- nrow(birds_aminoAcids)
birds_aminoAcid_count <- aa.count.function(birds_aminoAcids, 216)
aminoAcid_frequency.Matrix <- aa.frequency.matrix.function(birds_aminoAcid_count, 216)
bird_aminoAcid_frequencies <- aa.specimen.frequencies(aminoAcid_frequency.Matrix, birds_aminoAcids,
    birds_aminoAcid_speciesNames, 216)
birds_aminoAcid_specimenVLFcount <- aa.VLF.count.spec(bird_aminoAcid_frequencies, 0.001, 216)
birds_aaVLFconvert <- aa.VLF.convert.matrix(birds_aminoAcids, bird_aminoAcid_frequencies, 0.001,
    216)
birds_aminoAcidVLFs <- VLF.aminoAcids(birds_aaVLFconvert, birds_aminoAcids, 216)
birds_aaVLFreduced <- aa.VLF.reduced(birds_aminoAcidVLFs, birds_aminoAcid_specimenVLFcount, 216)
birds_aaSpecies <- separate(birds_aaVLFreduced)
birds_aminoAcid_singleAndShared <- aa.find.singles(birds_aaSpecies, 216)
## End(Not run)
```

---

## aa.conservation_first *First Modal Amino Acid Conservation*

---

### Description

Calculates the conservation of the first amino acids that occur most often in a matrix of amino acid sequences.

### Usage

```
aa.conservation_first(modal, p, seqlength)
```

### Arguments

| | |
|---|---|
| modal | A vector of the frequencies for the amino acids in the first modal sequence. |
| p | A conservation value for the amino acid frequencies to be compared to |
| seqlength | Length of the amino acid sequence |

### Details

The item modal can be calculated using the aa.MODE.freq function.

### Value

A vector that contains how many amino acids from the first modal sequence are conserved at the specified conservation level.

### Author(s)

Taryn B. T. Athey and Paul D. McNicholas

### Examples

```
## Not run: data(birds_aminoAcids)
birds_aminoAcid_count <- aa.count.function(birds_aminoAcids, 216)
aminoAcid_frequency.Matrix <- aa.frequency.matrix.function(birds_aminoAcid_count, 216)
aminoAcid_Modal <- aa.MODE(aminoAcid_frequency.Matrix, 216)
aminoAcid_firstModalFreq <- aa.MODE.freq(aminoAcid_frequency.Matrix, 216)
aminoAcid_firstConservation_100 <- aa.conservation_first(aminoAcid_firstModalFreq, 1, 216)
## End(Not run)
```

---

## aa.conservation_two *First and Second Modal Amino Acid Conservation*

---

### Description

Calculates the conservation of the amino acids that occur first and second most often in a matrix of sequences

### Usage

```
aa.conservation_two(modal1, modal2, p, seqlength)
```

### Arguments

| | |
|---|---|
| `modal1` | A vector of the frequencies for the amino acids in the first modal sequence. |
| `modal2` | A vector of the frequencies for the amino acids in the second modal sequence |
| `p` | A conservation value for the amino acid frequencies to be compared to. |
| `seqlength` | The length of the amino acid sequence. |

### Details

The argument modal1 can be calculated using the aa.MODE.freq function, and the argument modal2 can be calculated using he aa.MODE.second.freq function.

### Value

A vector that contains how many amino acids from the first and second modal sequences are conserved at the specified conservation level.

### Author(s)

Taryn B. T. Athey and Paul D. McNicholas

### Examples

```
## Not run: data(birds_aminoAcids)
birds_aminoAcid_count <- aa.count.function(birds_aminoAcids, 216)
aminoAcid_frequency.Matrix <- aa.frequency.matrix.function(birds_aminoAcid_count, 216)
aminoAcid_Modal <- aa.MODE(aminoAcid_frequency.Matrix, 216)
aminoAcid_firstModalFreq <- aa.MODE.freq(aminoAcid_frequency.Matrix, 216)
aminoAcid_secondModalFreq <- aa.MODE.second.freq(aminoAcid_frequency.Matrix, 216)
aminoAcid_secondConservation_99.9 <- aa.conservation_two(aminoAcid_firstModalFreq,
    aminoAcid_secondModalFreq, 0.999, 216)
## End(Not run)
```

---

## `aa.count.function` *Amino Acid Count*

---

### Description

Counts the number of each amino acid in each positino of the barcode.

### Usage

```
aa.count.function(aminoAcids, seqlength)
```

### Arguments

| | |
|---|---|
| `aminoAcids` | A matrix of barcode amino acid sequences. |
| `seqlength` | Length of the amino acid sequences. |

### Details

The first and second column of the aminoAcid argument must contain the unique specimen identifier and the species name, respectively, followed by the amino acid sequence.

### Value

A matrix containing the number of each amino acid in each position of the sequence. Each row is a different amino acid count, while the columns represent the sequence position.

### Author(s)

Taryn B. T. Athey and Paul D. McNicholas

### Examples

```
## Not run: data(birds_aminoAcids)
birds_aminoAcid_count <- aa.count.function(birds_aminoAcids, 216)
## End(Not run)
```

---

## aa.find.singles *Find amino acid singles*

---

### Description

Determines the number of shared and singleton amino acid VLFs.

### Usage

```
aa.find.singles(aaSpecies, seqlength)
```

### Arguments

| | |
|---|---|
| aaSpecies | List of amino acid sequences separated by species name. |
| seqlength | Length of amino acid sequences. |

### Details

The argument aaSpecies contains only amino acid VLFs, and NAs in any other position in the sequence. The list can be created using the separate function.

### Value

A matrix containing the number of singleton and shared aaVLFs in each position of the barcode.

### Author(s)

Taryn B. T. Athey and Paul D. McNicholas

## Examples

```
## Not run: data(birds_aminoAcids)
birds_aminoAcid_speciesNames <- birds_aminoAcids[,2]
aminoAcids_specimenNumber <- nrow(birds_aminoAcids)
birds_aminoAcid_count <- aa.count.function(birds_aminoAcids, 216)
aminoAcid_frequency.Matrix <- aa.frequency.matrix.function(birds_aminoAcid_count, 216)
bird_aminoAcid_frequencies <- aa.specimen.frequencies(aminoAcid_frequency.Matrix, birds_aminoAcids,
    birds_aminoAcid_speciesNames, 216)
birds_aminoAcid_specimenVLFcount <- aa.VLF.count.spec(bird_aminoAcid_frequencies, 0.001, 216)
birds_aaVLFconvert <- aa.VLF.convert.matrix(birds_aminoAcids, bird_aminoAcid_frequencies, 0.001,
    216)
birds_aminoAcidVLFs <- VLF.aminoAcids(birds_aaVLFconvert, birds_aminoAcids, 216)
birds_aaVLFreduced <- aa.VLF.reduced(birds_aminoAcidVLFs, birds_aminoAcid_specimenVLFcount, 216)
birds_aaSpecies <- separate(birds_aaVLFreduced)
birds_aminoAcid_singleAndShared <- aa.find.singles(birds_aaSpecies, 216)
## End(Not run)
```

---

`aa.frequency.matrix.function`

*Amino Acid Frequency Matrix*

---

## Description

Calculates the frequency of each amino acid.

## Usage

```
aa.frequency.matrix.function(aa.count, seqlength)
```

## Arguments

| | |
|---|---|
| `aa.count` | A matrix containing the number of each amino acid in each position. |
| `seqlength` | The length of the amino acid sequence |

## Details

The aa.count argument can be calculated using the function aa.count.function

## Value

A matrix of the frequencies for each amino acid in each position of the barcode sequence.

## Author(s)

Taryn B. T. Athey and Paul D. McNicholas

## Examples

```
## Not run: data(birds_aminoAcids)
birds_aminoAcid_count <- aa.count.function(birds_aminoAcids, 216)
aminoAcid_frequency.Matrix <- aa.frequency.matrix.function(birds_aminoAcid_count, 216)
## End(Not run)
```

---

# aa.MODE *Amino Acid Modal Sequence*

---

## Description

Calculates the amino acid sequence that occurs most often in a matrix of amino acid sequences

## Usage

```
aa.MODE(freq.matrix, seqlength)
```

## Arguments

| | |
|---|---|
| freq.matrix | Freuqncy matrix for amino acids. |
| seqlength | Length of amino acid sequences. |

## Details

The argument freq.matrix can be calculated using the function aa.frequency.matrix.function

## Value

A vector containing the amino acid sequence that occurs most often.

## Author(s)

Taryn B. T. Athey and Paul D. McNicholas

## Examples

```
## Not run: data(birds_aminoAcids)
birds_aminoAcid_count <- aa.count.function(birds_aminoAcids, 216)
aminoAcid_frequency.Matrix <- aa.frequency.matrix.function(birds_aminoAcid_count, 216)
aminoAcid_Modal <- aa.MODE(aminoAcid_frequency.Matrix, 216)
## End(Not run)
```

## aa.MODE.freq *Amino Acid Modal Frequencies*

### Description

Returns the frequencies of the amino acids that occur most often in each position of the sequence.

### Usage

```
aa.MODE.freq(freq.matrix, seqlength)
```

### Arguments

| | |
|---|---|
| freq.matrix | Frequency matrix for amino acids. |
| seqlength | Length of the amino acid sequences |

### Details

The argument freq.matrix can be calculated using the function aa.frequency.matrix.function

### Value

A vector of frequencies for the first modal sequence.

### Author(s)

Taryn B. T. Athey and Paul D. McNicholas

### Examples

```
## Not run: data(birds_aminoAcids)
birds_aminoAcid_count <- aa.count.function(birds_aminoAcids, 216)
aminoAcid_frequency.Matrix <- aa.frequency.matrix.function(birds_aminoAcid_count, 216)
aminoAcid_firstModalFreq <- aa.MODE.freq(aminoAcid_frequency.Matrix, 216)
## End(Not run)
```

## aa.MODE.second.freq *Amino Acid Second Modal Frequency*

### Description

Returns the frequencies of the amino acids that occur second most often in each position of a matrix of amino acid sequences.

### Usage

```
aa.MODE.second.freq(freq.matrix, seqlength)
```

### Arguments

| | |
|---|---|
| `freq.matrix` | Frequency matrix for amino acids. |
| `seqlength` | Length of amino acid sequences. |

### Details

The argument freq.matrix can be calculated using the function aa.frequency.matrix.function

### Value

A vector containing the frequencies of the second modal amino acid sequence.

### Author(s)

Taryn B. T. Athey and Paul D. McNicholas

### Examples

```
## Not run: data(birds_aminoAcids)
birds_aminoAcid_count <- aa.count.function(birds_aminoAcids, 216)
aminoAcid_frequency.Matrix <- aa.frequency.matrix.function(birds_aminoAcid_count, 216)
aminoAcid_secondModalFreq <- aa.MODE.second.freq(aminoAcid_frequency.Matrix, 216)
## End(Not run)
```

---

## `aa.specimen.frequencies` *Specimen Nucleotide Frequencies*

---

### Description

Converts a matrix of amino acid sequences into a matrix of amino acid frequencies.

### Usage

```
aa.specimen.frequencies(freq, seq.matrix, spec.names, seqlength)
```

### Arguments

| | |
|---|---|
| `freq` | Frequency matrix for amino acids. |
| `seq.matrix` | Matrix of specimen amino acid sequences. |
| `spec.names` | A vector of the species names for each specimen in aminoAcids in the ordfer they appear in the matrix. |
| `seqlength` | Length of amino acid sequences. |

### Details

The argument freq can be calculated using the function aa.frequency.matrix.function.

### Value

A matrix containing the frequencies of each amino acid in the sequence.

### Author(s)

Taryn B. T. Athey and Paul D. McNicholas

### Examples

```
## Not run: data(birds_aminoAcids)
birds_aminoAcid_speciesNames <- birds_aminoAcids[,2]
aminoAcids_specimenNumber <- nrow(birds_aminoAcids)
birds_aminoAcid_count <- aa.count.function(birds_aminoAcids,216)
aminoAcid_frequency.Matrix <- aa.frequency.matrix.function(birds_aminoAcid_count,216)
bird_aminoAcid_frequencies <- aa.specimen.frequencies(aminoAcid_frequency.Matrix, birds_aminoAcids,
    birds_aminoAcid_speciesNames, 216)
## End(Not run)
```

---

## aa.VLF.convert.matrix *Convert Amino Acid Matrix*

---

### Description

Converts a matrix of amino acid frequencies for each specimen into a matrix consisting of only VLF values and NAs in every non-VLF position.

### Usage

```
aa.VLF.convert.matrix(seq.matrix, freq, p, seqlength)
```

### Arguments

| | |
|---|---|
| seq.matrix | A matrix of aligned DNA barcode amino acid sequences. |
| freq | A matrix of amino acid frequencies for each specimen. |
| p | A very low frequency variant cut-off frequency. Any frequency in the freq matrix below this value is considered to be a very low frequency variant. |
| seqlength | The length of the amino acid sequences. |

### Value

A matrix of VLF amino acid frequencies, containing only those nucleotide frequencies that occur less than the designation p value, and NAs in each other position of the matrix.

### Author(s)

Taryn B. T. Athey and Paul D. McNicholas

## Examples

```
## Not run: data(birds_aminoAcids)
birds_aminoAcid_speciesNames <- birds_aminoAcids[,2]
aminoAcids_specimenNumber <- nrow(birds_aminoAcids)
birds_aminoAcid_count <- aa.count.function(birds_aminoAcids, 216)
aminoAcid_frequency.Matrix <- aa.frequency.matrix.function(birds_aminoAcid_count, 216)
bird_aminoAcid_frequencies <- aa.specimen.frequencies(aminoAcid_frequency.Matrix, birds_aminoAcids,
    birds_aminoAcid_speciesNames, 216)
birds_aaVLFconvert <- aa.VLF.convert.matrix(birds_aminoAcids, bird_aminoAcid_frequencies, 0.001,
    216)
## End(Not run)
```

---

# aa.VLF.count.pos *VLF position count*

---

## Description

Calculates the number of very low frequency variants in each position in a matrix of sequences

## Usage

```
aa.VLF.count.pos(freq, p, seqlength)
```

## Arguments

| | |
|---|---|
| freq | A matrix of frequencies for each speicmen. |
| p | A very low frequency variant cut off frequency. Any frequency in the freq matrix below this value is considered to be a very low frequency variant. |
| seqlength | The length of the amino acid sequences. |

## Value

A vector containing the amino acid VLF count for each position of the sequence.

## Author(s)

Taryn B. T. Athey and Paul D. McNicholas

## Examples

```
## Not run: data(birds_aminoAcids)
birds_aminoAcid_speciesNames <- birds_aminoAcids[,2]
aminoAcids_specimenNumber <- nrow(birds_aminoAcids)
birds_aminoAcid_count <- aa.count.function(birds_aminoAcids, 216)
aminoAcid_frequency.Matrix <- aa.frequency.matrix.function(birds_aminoAcid_count, 216)
bird_aminoAcid_frequencies <- aa.specimen.frequencies(aminoAcid_frequency.Matrix, birds_aminoAcids,
    birds_aminoAcid_speciesNames, 216)
birds_aminoAcid_positionVLFcount <- aa.VLF.count.pos(bird_aminoAcid_frequencies, 0.001, 216)
## End(Not run)
```

## aa.VLF.count.spec *VLF Specimen Count*

### Description

Calculates the number of very low frequency variants for each specimen in a matrix of sequences.

### Usage

```
aa.VLF.count.spec(freq, p, seqlength)
```

### Arguments

| | |
|---|---|
| freq | A matrix of amino acid frequencies for each specimen. |
| p | A very low frequency variant cut-off frequency. Any frequency in the freq matrix below this value is considered to be a very low frequency variant. |
| seqlength | The length of the amino acid sequences. |

### Value

A vector containing the aaVLF count for every specimen.

### Author(s)

Taryn B. T. Athey and Paul D. McNicholas

### Examples

```
## Not run: data(birds_aminoAcids)
birds_aminoAcid_speciesNames <- birds_aminoAcids[,2]
aminoAcids_specimenNumber <- nrow(birds_aminoAcids)
birds_aminoAcid_count <- aa.count.function(birds_aminoAcids, 216)
aminoAcid_frequency.Matrix <- aa.frequency.matrix.function(birds_aminoAcid_count, 216)
bird_aminoAcid_frequencies <- aa.specimen.frequencies(aminoAcid_frequency.Matrix, birds_aminoAcids,
    birds_aminoAcid_speciesNames, 216)
birds_aminoAcid_specimenVLFcount <- aa.VLF.count.spec(bird_aminoAcid_frequencies, 0.001, 216)
## End(Not run)
```

## aa.VLF.reduced *Amino Acid Reduced*

### Description

Reduces a matrix of amino acid very low frequency variants (aaVLFs) so that only those specimen that contain aaVLFs remain

### Usage

```
aa.VLF.reduced(NA.matrix, sCount, seqlength)
```

### Arguments

| | |
|---|---|
| NA.matrix | A matrix with values for amino acid very low frequency variants for each specimen and NAs in all other positions. |
| sCount | A vector for amino acid very low frequency variant (VLF) counts for each specimen in the NA.matrix. |
| seqlength | Length of the amino acid sequences. |

### Details

The argument NA.matrix can be calculated using aa.VLF.convert.matrix and VLF.aminoAcids, the sCount argument can be calculaed using the aa.VLF.count.spec function.

### Value

A matrix containing only specimen with aaVLFs, and only the aaVLF values in the sequences. All other positions of the sequence contain NAs.

### Author(s)

Taryn B. T. Athey and Paul D. McNicholas

### Examples

```
## Not run: data(birds_aminoAcids)
birds_aminoAcid_speciesNames <- birds_aminoAcids[,2]
aminoAcids_specimenNumber <- nrow(birds_aminoAcids)
birds_aminoAcid_count <- aa.count.function(birds_aminoAcids, 216)
aminoAcid_frequency.Matrix <- aa.frequency.matrix.function(birds_aminoAcid_count, 216)
bird_aminoAcid_frequencies <- aa.specimen.frequencies(aminoAcid_frequency.Matrix, birds_aminoAcids,
    birds_aminoAcid_speciesNames, 216)
birds_aminoAcid_specimenVLFcount <- aa.VLF.count.spec(bird_aminoAcid_frequencies, 0.001, 216)
birds_aaVLFconvert <- aa.VLF.convert.matrix(birds_aminoAcids, bird_aminoAcid_frequencies, 0.001,
    216)
birds_aminoAcidVLFs <- VLF.aminoAcids(birds_aaVLFconvert, birds_aminoAcids, 216)
birds_aaVLFreduced <- aa.VLF.reduced(birds_aminoAcidVLFs, birds_aminoAcid_specimenVLFcount, 216)
## End(Not run)
```

## aaVLFs.to.modalchanges

*Amino Acid Changes*

### Description

Determines how many aaVLFs have changed "type" of amino acid from the modal amino acid sequence. Amino acid types are polar charged, polar uncharged, non-polar, and those with a unique side group.

### Usage

```
aaVLFs.to.modalchanges(modal, AminoAcidList, aalength)
```

### Arguments

| | |
|---|---|
| `modal` | The modal amino acid sequence (i.e., the amino acid sequence that occurs most often based on the amino acid frequency matrix) |
| `AminoAcidList` | Matrix of VLF amino acid sequences containing only aaVLFs and NAs anywhere else |
| `aalength` | Amino Acid sequence length. |

### Details

The argument modal can be created using the MODE function. The argument AminoAcidList can be created using the aa.VLF.convert.matrix, VLF.aminoAcids, and aa.VLF.reduced functions.

### Value

A sameAll value representative of the number of amino acids that were the same type as the modal, a changedAll value representative of the number of amino acids that changed amino acid type from the modal.

### Author(s)

Taryn B. T. Athey and Paul D. McNicholas

### Examples

```
## Not run: data(birds_aminoAcids)
birds_aminoAcid_speciesNames <- birds_aminoAcids[,2]
aminoAcids_specimenNumber <- nrow(birds_aminoAcids)
birds_aminoAcid_count <- aa.count.function(birds_aminoAcids, 216)
aminoAcid_frequency.Matrix <- aa.frequency.matrix.function(birds_aminoAcid_count, 216)
bird_aminoAcid_frequencies <- aa.specimen.frequencies(aminoAcid_frequency.Matrix, birds_aminoAcids,
    birds_aminoAcid_speciesNames, 216)
aminoAcid_Modal <- aa.MODE(aminoAcid_frequency.Matrix, 216)
birds_aminoAcid_specimenVLFcount <- aa.VLF.count.spec(bird_aminoAcid_frequencies, 0.001, 216)
```

```
birds_aaVLFconvert <- aa.VLF.convert.matrix(birds_aminoAcids, bird_aminoAcid_frequencies, 0.001,
    216)
birds_aminoAcidVLFs <- VLF.aminoAcids(birds_aaVLFconvert, birds_aminoAcids, 216)
birds_aaVLFreduced <- aa.VLF.reduced(birds_aminoAcidVLFs, birds_aminoAcid_specimenVLFcount, 216)
All_aaType_change <- aaVLFs.to.modalchanges(aminoAcid_Modal, birds_aaVLFreduced, 216)
## End(Not run)
```

---

`aminoAcid.matching.positions`

*Matching Amino Acid Positions*

---

## Description

Gives the position and residue of the amino acid VLFs in a matrix containing amino acid VLFs and NAs.

## Usage

```
aminoAcid.matching.positions(matchAA, aalength)
```

## Arguments

| | |
|---|---|
| `matchAA` | A matrix containing aaVLFs and NAs in all other positions of the sequences. |
| `aalength` | Amino acid sequence length. |

## Details

The argument matchAA can be calculated usingthe find.matching function and taking the first argument from the returned value.

## Value

A list for each aaVLF with a matching specimen identifier to a ntVLF. The first position in each entry of the list contains the specimen identifier, the second position contains the species name, the third position contains the sequence position of the aaVLF, and the fourth position contains the aaVLF.

## Author(s)

Taryn B. T. Athey and Paul D. McNicholas

## Examples

```
## Not run: #Nucleotide VLF analysis
data(birds)
species.names <- birds[,2]
specimen.Number <- nrow(birds)
rownames(birds) <- species.names
Nuc.count <- count.function(birds, specimen.Number, 648)
frequency.matrix <- ffrequency.matrix.function(Nuc.count, 648)
birdSpec.freq <- specimen.frequencies(frequency.matrix, birds, specimen.Number, species.names, 648)
Bird_specimen_VLFcount <- VLF.count.spec(birdSpec.freq, 0.001, 648)
bird_VLFconvert <- VLF.convert.matrix(birds, birdSpec.freq, 0.001, 648)
bird_VLFnuc <- VLF.nucleotides(bird_VLFconvert, birds, 648)
bird_VLFreduced <- VLF.reduced(bird_VLFnuc, Bird_specimen_VLFcount, 648)
bird_species <- separate(bird_VLFreduced)
birds_singleAndShared <- find.singles(bird_species, 648)

#Amino Acid VLF Analysis
data(birds_aminoAcids)
birds_aminoAcid_speciesNames <- birds_aminoAcids[,2]
aminoAcids_specimenNumber <- nrow(birds_aminoAcids)
birds_aminoAcid_count <- aa.count.function(birds_aminoAcids, 216)
aminoAcid_frequency.Matrix <- aa.frequency.matrix.function(birds_aminoAcid_count, 216)
bird_aminoAcid_frequencies <- aa.specimen.frequencies(aminoAcid_frequency.Matrix, birds_aminoAcids,
    birds_aminoAcid_speciesNames, 216)
aminoAcid_Modal <- aa.MODE(aminoAcid_frequency.Matrix, 216)
birds_aminoAcid_specimenVLFcount <- aa.VLF.count.spec(bird_aminoAcid_frequencies, 0.001, 216)
birds_aaVLFconvert <- aa.VLF.convert.matrix(birds_aminoAcids, bird_aminoAcid_frequencies, 0.001,
    216)
birds_aminoAcidVLFs <- VLF.aminoAcids(birds_aaVLFconvert, birds_aminoAcids, 216)
birds_aaVLFreduced <- aa.VLF.reduced(birds_aminoAcidVLFs, birds_aminoAcid_specimenVLFcount, 216)
birds_aaSpecies <- separate(birds_aaVLFreduced)
birds_aminoAcid_singleAndShared <- aa.find.singles(birds_aaSpecies, 216)

#Concordance Analysis
VLF_match <- find.matching(bird_VLFreduced, birds_aaVLFreduced, 648, 216)
position_matchingNuc <- nucleotide.matching.positions(VLF_match[[2]], 648)
position_matchingAA <- aminoAcid.matching.positions(VLF_match[[1]], 216)
## End(Not run)
```

---

`aminoAcidFun` *Amino Acid VLF Analysis Function*

---

## Description

Runs the full amino acid VLF analysis for the user and outputs results.

## Usage

```
aminoAcidFun(x, p = 0.001, seqlength = 216, own = NULL)
```

### Arguments

| | |
|---|---|
| `x` | A matrix of amino acid sequences with the first column containing the unique specimen identifier, the second column containing the species name and the remaining columns containing the amino acid sequence. |
| `p` | A VLF designation frequency cut-off to be used within the analysis. By default p = 0.001. |
| `seqlength` | The length of the amino acid sequence. By default seqlength = 216. |
| `own` | If the user wants to compare their own sequences separate from reference sequences, then argument can be used. Similar to x, this argument is a matrix of amino acid sequences with the first column containing the unique specimen identifier, the second column contains the species name and the remaining columns containing the nucleotide sequence. By default own = NULL.) |

### Value

| | |
|---|---|
| `modal` | A vector containing the amino acid sequence that occurs most often in the dataset. |
| `con100` | The number of amino acid positions that are 100% conserved in the sequence |
| `conp` | The number of amino acid positions that are (1-p)% conserved in the sequence |
| `combine` | The number of amino acid positions that are (1-p)% conserved when combining the first and second modal sequences. |
| `specimen` | A vector containing the number of VLFs for each specimen in the dataset. |
| `position` | A vector containing the number of VLFs for each position in the sequences. |
| `sas` | A matrix containing vectors of single and shared amino acid VLF counts for each position of the sequence. |
| `VLFmatrix` | A matrix containing only those specimen that have VLFs as well as the amino acid at the positions that contain VLFs and NAs in all other positions. |
| `ownSpecCount` | A vector containing the number of VLFs for each specimen in the users own specified dataset. Only appears if own is not NULL. |
| `ownPosCount` | A vector containing the number of VLFs for each position in the sequences of the users own specified dataset. Only appears if own is not NULL. |
| `ownVLFMatrix` | A matrix containing only those amino acids at the positions that contain VLFs and NAs in all other positions of the sequence. Only appears if own is not NULL. |
| `ownVLFreduced` | A matrix containing only those specimen that have VLFs as well as the amino acids at the positions that contain VLFs and NAs in all other positions of the sequence. Only appears if own is not NULL. |

### Author(s)

Taryn B. T. Athey and Paul D. McNicholas

### Examples

```
## Not run: data(birds_aminoAcids)
bird_aaAnalysis <- aminoAcidFun(birds_aminoAcids)
## End(Not run)
```

---

## birds *Bird Nucleotide Sequences*

---

### Description

Data set containing nucleotide sequences for 11,333 bird barcodes.

### Usage

```
data(birds)
```

### Format

The format is: chr [1:11333, 1:650] "gi|359280039|gb|JQ173884.1|" ... - attr(*, "dimnames")=List of 2 ..$ : NULL ..$ : chr [1:650] "V1" "V2" "V3" "V4" ...

### Source

Stoeckle, M. Y. and Kerr, K. C. R. (2012) Frequency Matrix Approach Demonstrates High Sequence Quality in Avian BARCODEs and Highlights Cryptic Pseudogenes. PLoS ONE. 7, e43992.

### Examples

```
## Not run: data(birds)
```

---

## birds_aminoAcids *Bird Amino Acid Sequences*

---

### Description

Data set containing amino acid sequences for 11,333 bird barcodes.

### Usage

```
data(birds_aminoAcids)
```

### Format

The format is: chr [1:11333, 1:218] "gi|359280039|gb|JQ173884.1|" ... - attr(*, "dimnames")=List of 2 ..$ : NULL ..$ : chr [1:218] "V1" "V2" "V3" "V4" ...

### Source

Stoeckle, M. Y. and Kerr, K. C. R. (2012) Frequency Matrix Approach Demonstrates High Sequence Quality in Avian BARCODEs and Highlights Cryptic Pseudogenes. PLoS ONE. 7, e43992.

### Examples

```
## Not run: data(birds_aminoAcids)
```

---

## compare — *Compare VLFs within Species*

---

### Description

Compares VLFs between specimen of the same species.

### Usage

```
compare(x, seqlength)
```

### Arguments

| | |
|---|---|
| x | A list of sequences separated by species name. Each entry in the list contains a matrix of sequences from the same species. |
| seqlength | Length of the sequences. |

### Details

List of sequences by species names, x, can be created using the separate function

### Value

A matrix containing two vectors, one with singleton VLF counts for each position of the sequence, and one with shared VLF counts for each position of the sequence.

### Author(s)

Taryn B. T. Athey and Paul D. McNicholas

### Examples

```
## Not run: data(birds)
species.names <- birds[,2]
specimen.Number <- nrow(birds)
rownames(birds) <- species.names
Nuc.count <- count.function(birds, specimen.Number, 648)
frequency.matrix <- ffrequency.matrix.function(Nuc.count, 648)
birdSpec.freq <- specimen.frequencies(frequency.matrix, birds, specimen.Number, species.names, 648)
Bird_specimen_VLFcount <- VLF.count.spec(birdSpec.freq, 0.001, 648)
```

```
bird_VLFconvert <- VLF.convert.matrix(birds, birdSpec.freq, 0.001, 648)
bird_VLFnuc <- VLF.nucleotides(bird_VLFconvert, birds, 648)
bird_VLFreduced <- VLF.reduced(bird_VLFnuc, Bird_specimen_VLFcount, 648)
bird_species <- separate(bird_VLFreduced)

#The compare function is called on from within the find.singles function
birds_singleAndShared <- find.singles(bird_species, 648)
## End(Not run)
```

---

`concordanceFun` *VLF Concordance Check Function*

---

## Description

Compares ntVLFs to aaVLFs to see if they are concordant (i.e., if the ntVLF causes the aaVLF).

## Usage

```
concordanceFun(nuc, aa, nuclength = 648, aalength = 216, aminoAcid_Modal)
```

## Arguments

| | |
|---|---|
| `nuc` | A matrix of ntVLFs, that contains only those specimen with VLFs, and a sequence with only VLF nucleotides and NAs in all other positions of the nucleotide sequences. |
| `aa` | A matrix of aaVLFs, that contains only those specimen with VLFs, and a sequence with only VLF amino acids and NAs in all other positions of the amino acid sequence. |
| `nuclength` | The length of the nucleotide sequence. By default is 648. |
| `aalength` | The length of the amino acid sequence. By default is 216. |
| `aminoAcid_Modal` | The modal amino acid sequence (i.e., the amino acid sequence that occurs most often in the given sequences) |

## Details

The argument nuc can be taken from the VLFmatrix output from the vlfFun function. The argument aa can be taken from the VLFmatrix output from the aminoAcidFun function. The argument aminoAcid_Modal can be taken from the modal output from the aminoAcidFun function.

## Value

| | |
|---|---|
| `matched` | A list of the concordant ntVLFs and aaVLFs. Contains the specimen identifier, the species name, the concordant amino acid, the amino acid position, and the concordant amino acid position. There may be mulitple entries for the same aaVLF if that VLF is concordant to more than one ntVLF. |

| | |
|---|---|
| codons | A vector containing calculations for how many of the concordant amino acids were caused by changes in each of the nucleotide codon positions. |
| concordantType | Contains information on how many of the concordant aaVLFs had a change in amino acid residue type and how many remained in the same amino acid residue category. |
| aminoAcidType | Contains information on how many of the aaVLFs had a change in amino acid residue type and how many remained in the same amino acid residue category. |
| concordNuc | Gives the number of ntVLFs that showed concordance to aaVLFs. |
| concordAA | Gives the number of aaVLFs that showed concordance to ntVLFs. |
| sequences | Gives the number of sequences that had both ntVLFs and aaVLFs. |

## Author(s)

Taryn B. T. Athey and Paul D. McNicholas

## Examples

```
## Not run: #VLF analysis
data(birds)
bird_vlfAnalysis <- vlfFun(birds)

#Amino Acid analysis
data(birds_aminoAcids)
bird_aaAnalysis <- aminoAcidFun(birds_aminoAcids)

#Concordance analysis
bird_Concordance <- concordanceFun(bird_vlfAnalysis$VLFmatrix, bird_aaAnalysis$VLFmatrix, 648, 216,
    bird_aaAnalysis$modal)
## End(Not run)
```

---

# concordant.to.modalchanges

*Concordant Residue Changes*

---

## Description

Deternubes how many concordant aaVLFs have changed type of amino acid from the modal amino acid sequence. Amino acid residue types are polar charged, polar uncharged, non-polar, and amino acids with a unique side group.

## Usage

```
concordant.to.modalchanges(matched, modal)
```

### Arguments

| | |
|---|---|
| matched | A list containing the concordant aaVLFs and their properties (e.g., sequence position). |
| modal | A vector containing the modal amino acid sequence. |

### Details

The matched argument can be calculated using the overall.matched function. The modal argument can be calculated using the aa.MODE function.

### Value

A vector containing the number of concordant aaVLFs that changed amino acid residue type, and the number that contained the same residue type.

### Author(s)

Taryn B. T. Athey and Paul D. McNicholas

### Examples

```
## Not run: #Nucleotide VLF analysis
data(birds)
species.names <- birds[,2]
specimen.Number <- nrow(birds)
rownames(birds) <- species.names
Nuc.count <- count.function(birds, specimen.Number, 648)
frequency.matrix <- ffrequency.matrix.function(Nuc.count, 648)
birdSpec.freq <- specimen.frequencies(frequency.matrix, birds, specimen.Number, species.names, 648)
Bird_specimen_VLFcount <- VLF.count.spec(birdSpec.freq, 0.001, 648)
bird_VLFconvert <- VLF.convert.matrix(birds, birdSpec.freq, 0.001, 648)
bird_VLFnuc <- VLF.nucleotides(bird_VLFconvert, birds, 648)
bird_VLFreduced <- VLF.reduced(bird_VLFnuc, Bird_specimen_VLFcount, 648)
bird_species <- separate(bird_VLFreduced)
birds_singleAndShared <- find.singles(bird_species, 648)

#Amino Acid VLF Analysis
data(birds_aminoAcids)
birds_aminoAcid_speciesNames <- birds_aminoAcids[,2]
aminoAcids_specimenNumber <- nrow(birds_aminoAcids)
birds_aminoAcid_count <- aa.count.function(birds_aminoAcids, 216)
aminoAcid_frequency.Matrix <- aa.frequency.matrix.function(birds_aminoAcid_count, 216)
bird_aminoAcid_frequencies <- aa.specimen.frequencies(aminoAcid_frequency.Matrix, birds_aminoAcids,
    birds_aminoAcid_speciesNames, 216)
aminoAcid_Modal <- aa.MODE(aminoAcid_frequency.Matrix, 216)
birds_aminoAcid_specimenVLFcount <- aa.VLF.count.spec(bird_aminoAcid_frequencies, 0.001, 216)
birds_aaVLFconvert <- aa.VLF.convert.matrix(birds_aminoAcids, bird_aminoAcid_frequencies, 0.001,
    216)
birds_aminoAcidVLFs <- VLF.aminoAcids(birds_aaVLFconvert, birds_aminoAcids, 216)
birds_aaVLFreduced <- aa.VLF.reduced(birds_aminoAcidVLFs, birds_aminoAcid_specimenVLFcount, 216)
birds_aaSpecies <- separate(birds_aaVLFreduced)
```

```
birds_aminoAcid_singleAndShared <- aa.find.singles(birds_aaSpecies, 216)

#Concordance Analysis
VLF_match <- find.matching(bird_VLFreduced, birds_aaVLFreduced, 648, 216)
position_matchingNuc <- nucleotide.matching.positions(VLF_match[[2]], 648)
position_matchingAA <- aminoAcid.matching.positions(VLF_match[[1]], 216)
matching_comparison <- overall.matched(position_matchingNuc, position_matchingAA, 648, 216)
concordant_aaType_change <- concordant.to.modalchanges(matching_comparison, aminoAcid_Modal)
## End(Not run)
```

---

`conservation_first` *First Modal Conserved*

---

## Description

Calculates the conservation of the nucleotides that occur most often in a matrix of sequences/

## Usage

```
conservation_first(modal, p, seqlength)
```

## Arguments

| | |
|---|---|
| `modal` | A vector of the frequencies of the nucleotides in the first modal sequences. |
| `p` | A conservation value for the nucleotide frequencies to be compared to. |
| `seqlength` | The length of the nucleotide sequence. |

## Details

The argument modal can be calculated using the MODE.freq function.

## Value

A vector that contains how many nucleotides from the first modal sequence are conserved at the specified conservation level for each codon position.

## Author(s)

Taryn B. T. Athey and Paul D. McNicholas

## Examples

```
## Not run: data(birds)
species.names <- birds[,2]
specimen.Number <- nrow(birds)
rownames(birds) <- species.names
Nuc.count <- count.function(birds, specimen.Number, 648)
frequency.matrix <- ffrequency.matrix.function(Nuc.count, 648)
birdSpec.freq <- specimen.frequencies(frequency.matrix, birds, specimen.Number, species.names, 648)
```

```
nucleotide.modalSequence <- MODE(frequency.matrix, 648)
Bird_first.modal.frequencies <- MODE.freq(frequency.matrix, 648)
First_conserved_100 <- conservation_first(Bird_first.modal.frequencies, 1, 648)
First_conserved_99.9 <- conservation_first(Bird_first.modal.frequencies, 0.999, 648)
## End(Not run)
```

---

`conservation_two` *First and Second Modal Conserved*

---

## Description

Calculates the conservation of the nucleotides that occur first and second most often in a matrix of sequences.

## Usage

```
conservation_two(modal1, modal2, p, seqlength)
```

## Arguments

| | |
|---|---|
| `modal1` | A vector of the frequencies for the nucleotides in the first modal sequence. |
| `modal2` | A vector of the frequencies for the nucleotides that occur second most often. |
| `p` | A conservation value for the nucleotide frequencies to be compared to. |
| `seqlength` | The nucleotide sequence length. |

## Details

The argument modal1 can be calculated using the function MODE.freq. The argument modal2 can be calculated using the function MODE.second.freq.

## Value

A vector that contains how many nucleotides from the first and second modal sequences are conserved at the specified conservation level for each codon position.

## Author(s)

Taryn B. T. Athey and Paul D. McNicholas

## Examples

```
## Not run: data(birds)
species.names <- birds[,2]
specimen.Number <- nrow(birds)
rownames(birds) <- species.names
Nuc.count <- count.function(birds, specimen.Number, 648)
frequency.matrix <- ffrequency.matrix.function(Nuc.count, 648)
birdSpec.freq <- specimen.frequencies(frequency.matrix, birds, specimen.Number, species.names, 648)
```

```
nucleotide.modalSequence <- MODE(frequency.matrix, 648)
Bird_first.modal.frequencies <- MODE.freq(frequency.matrix, 648)
Bird_second.modal.frequencies <- MODE.second.freq(frequency.matrix, 648)
FirstAndSecond_conserved_99.9 <- conservation_two(Bird_first.modal.frequencies,
    Bird_second.modal.frequencies, 0.999, 648)
## End(Not run)
```

---

`count.function` *Nucleotide Count*

---

## Description

Counts the number of each dNTP in each position of an aligned barcode matrix.

## Usage

```
count.function(nucleotides, spec.no, seqlength)
```

## Arguments

| | |
|---|---|
| `nucleotides` | A matrix of aligned DNA barcode sequences. DNA sequences should start at the third column of the matrix, while the first column contains a unique specimen identifier and the second column contains the species name. |
| `spec.no` | The number of specimen/sequences in the nucldeotide matrix. |
| `seqlength` | The length of the nucleotide sequences. |

## Value

A matrix containing the number of each nucleotide in each position of the sequence. Each row is a different dNTP count, while the columns represent the sequence position.

## Author(s)

Taryn B. T. Athey and Paul D. McNicholas

## Examples

```
## Not run: data(birds)
Nuc.count <- count.function(birds, specimen.Number, 648)
## End(Not run)
```

`Decile.Plot` *VLF Decile Plot*

## Description

Creates a plot of VLF distributions summed for each decile segment.

## Usage

```
Decile.Plot(VLF, seqlength)
```

## Arguments

| | |
|---|---|
| `VLF` | A list of VLFs in each barcode position. May be a matrix containing vectors of singleton and shared VLFs, or can be a single vector of total VLFs. |
| `seqlength` | The length of the sequence. Usually 648 for nucleotide sequences and 216 for amino acid sequences. |

## Value

A barplot containing the sum of the VLFs for each decile barcode segment.

## Author(s)

Taryn B. T. Athey and Paul D. McNicholas

## Examples

```
## Not run: data(birds)
species.names <- birds[,2]
specimen.Number <- nrow(birds)
rownames(birds) <- species.names
Nuc.count <- count.function(birds, specimen.Number, 648)
frequency.matrix <- ffrequency.matrix.function(Nuc.count, 648)
birdSpec.freq <- specimen.frequencies(frequency.matrix, birds, specimen.Number, species.names, 648)
Bird_position_VLFcount <- VLF.count.pos(birdSpec.freq, 0.001, 648)
Decile.Plot(Bird_position_VLFcount, 648)
## End(Not run)
```

`Error.Rate` *Error Rate*

## Description

Calculates shared, singleton, and total second codon position error rate in a matrix of sequences.

## Usage

```
Error.Rate(single, shared, spec, seqlength)
```

## Arguments

| | |
|---|---|
| `single` | A vector of singleton very low frequency variant (VLF) counts for each position in the sequence. |
| `shared` | A vector of shared very low frequenct variant (VLF) counts for each position in the sequence. |
| `spec` | The number of specimen being considered in the dataset. |
| `seqlength` | The length of the barcode sequence. |

## Details

The arguments single and shared can be calculated simultaneously using the find.singles function. The spec argument can be calculated by using the nrow() function on the sequence matrix.

## Value

A vector containing the single, shared, and total error rate based on the number of second position VLFs.

## Author(s)

Taryn B. T. Athey and Paul D. McNicholas

## Examples

```
## Not run: data(birds)
species.names <- birds[,2]
specimen.Number <- nrow(birds)
rownames(birds) <- species.names
Nuc.count <- count.function(birds, specimen.Number, 648)
frequency.matrix <- ffrequency.matrix.function(Nuc.count, 648)
birdSpec.freq <- specimen.frequencies(frequency.matrix, birds, specimen.Number, species.names, 648)
Bird_specimen_VLFcount <- VLF.count.spec(birdSpec.freq, 0.001, 648)
Bird_position_VLFcount <- VLF.count.pos(birdSpec.freq, 0.001, 648)
bird_VLFconvert <- VLF.convert.matrix(birds, birdSpec.freq, 0.001, 648)
bird_VLFnuc <- VLF.nucleotides(bird_VLFconvert, birds, 648)
bird_VLFreduced <- VLF.reduced(bird_VLFnuc, Bird_specimen_VLFcount, 648)
```

```
bird_species <- separate(bird_VLFreduced)
birds_singleAndShared <- find.singles(bird_species, 648)
Bird_error <- Error.Rate(birds_singleAndShared[1,], birds_singleAndShared[2,], specimen.Number, 648)
## End(Not run)
```

---

`fasta.read` *Read Fasta Files*

---

## Description

Reads in fasta files and converts into sequence matrix

## Usage

```
fasta.read(file, seqlength = 648, pos1 = 1, pos2 = 3)
```

## Arguments

| | |
|---|---|
| `file` | A fasta file to be read in. |
| `seqlength` | Length of sequence. |
| `pos1` | The position within the fasta title of the unique specimen identifier. By default pos1 = 1. |
| `pos2` | The position within the fasta title of the species name. By default pos2 = 3. |

## Value

A matrix of sequences, with the unique specimen identifers in the first column, the species names in the second column, and the sequence starting in the third column.

## Author(s)

Taryn B. T. Athey and Paul D. McNicholas

---

`ffrequency.matrix.function` *Frequency Matrix*

---

## Description

Calculates the frequency of each dNTP in each position of a nucleotide count matrix.

## Usage

```
ffrequency.matrix.function(count.matrix, seqlength)
```

### Arguments

| | |
|---|---|
| `count.matrix` | A matrix of the counts for each dNTP from a matrix of aligned sequences. |
| `seqlength` | Length of sequences. |

### Details

The argument count.matrix can be calculated using the function count.function.

### Value

A matrix of the frequencies for each dNTP in each position of the barcode sequence.

### Author(s)

Taryn B. T. Athey and Paul D. McNicholas

### Examples

```
## Not run: #Nucleotide VLF analysis
data(birds)
Nuc.count <- count.function(birds, specimen.Number, 648)
frequency.matrix <- ffrequency.matrix.function(Nuc.count, 648)
## End(Not run)
```

---

## `find.matching` *Find Matching ntVLF and aaVLF Specimen.*

---

### Description

Compares a list of aaVLF and ntVLF matrices for common specimen identifiers.

### Usage

```
find.matching(NucleotideList, AminoAcidList, nuclength, aalength)
```

### Arguments

| | |
|---|---|
| `NucleotideList` | Matrix of VLF nucleotide sequences containing the only the nucleotidies that are VLFs and NAs in the other positions of the sequences. |
| `AminoAcidList` | Matrix of VLF amino acid sequences containing only the aaVLFs and NAs in the other positions of the sequences. |
| `nuclength` | Length of the nucleotide sequence (should be 3X the length of the amino acid sequence). |
| `aalength` | Length of the amino acid sequence (should be 1/3 the length of the nucleotide sequence). |

## Details

The argument NucleotideList can be calculated using the VLF.convert.matrix, VLF.nucleotides, and VLF.reduced functions. The argument AminoAcidList can be calculated using the aa.VLF.convert.matrix, VLF.aminoAcids, and aa.VLF.reduced functions.

## Value

A list containing matrices of aaVLFs in the first position and ntVLFs in the second position who have matching specimen identifiers.

## Author(s)

Taryn B. T. Athey and Paul D. McNicholas

## Examples

```
## Not run: #Nucleotide VLF analysis
data(birds)
species.names <- birds[,2]
specimen.Number <- nrow(birds)
rownames(birds) <- species.names
Nuc.count <- count.function(birds, specimen.Number, 648)
frequency.matrix <- ffrequency.matrix.function(Nuc.count, 648)
birdSpec.freq <- specimen.frequencies(frequency.matrix, birds, specimen.Number, species.names, 648)
Bird_specimen_VLFcount <- VLF.count.spec(birdSpec.freq, 0.001, 648)
bird_VLFconvert <- VLF.convert.matrix(birds, birdSpec.freq, 0.001, 648)
bird_VLFnuc <- VLF.nucleotides(bird_VLFconvert, birds, 648)
bird_VLFreduced <- VLF.reduced(bird_VLFnuc, Bird_specimen_VLFcount, 648)
bird_species <- separate(bird_VLFreduced)
birds_singleAndShared <- find.singles(bird_species, 648)

#Amino Acid VLF Analysis
data(birds_aminoAcids)
birds_aminoAcid_speciesNames <- birds_aminoAcids[,2]
aminoAcids_specimenNumber <- nrow(birds_aminoAcids)
birds_aminoAcid_count <- aa.count.function(birds_aminoAcids, 216)
aminoAcid_frequency.Matrix <- aa.frequency.matrix.function(birds_aminoAcid_count, 216)
bird_aminoAcid_frequencies <- aa.specimen.frequencies(aminoAcid_frequency.Matrix, birds_aminoAcids,
    birds_aminoAcid_speciesNames, 216)
aminoAcid_Modal <- aa.MODE(aminoAcid_frequency.Matrix, 216)
birds_aminoAcid_specimenVLFcount <- aa.VLF.count.spec(bird_aminoAcid_frequencies, 0.001, 216)
birds_aaVLFconvert <- aa.VLF.convert.matrix(birds_aminoAcids, bird_aminoAcid_frequencies, 0.001,
    216)
birds_aminoAcidVLFs <- VLF.aminoAcids(birds_aaVLFconvert, birds_aminoAcids, 216)
birds_aaVLFreduced <- aa.VLF.reduced(birds_aminoAcidVLFs, birds_aminoAcid_specimenVLFcount, 216)
birds_aaSpecies <- separate(birds_aaVLFreduced)
birds_aminoAcid_singleAndShared <- aa.find.singles(birds_aaSpecies, 216)

#Concordance Analysis
VLF_match <- find.matching(bird_VLFreduced, birds_aaVLFreduced, 648, 216)
## End(Not run)
```

## find.singles *Single and Shared VLF Find*

### Description

Calculates the number of singleton and shared VLFs for each position of the nucleotide, by first seeing if there is only one specimen for a species, and then calling on the compare() function to calculate the number of singleton and shared VLFs for those species with multiple specimen.

### Usage

```
find.singles(species, seqlength)
```

### Arguments

| | |
|---|---|
| species | A list of sequences separated byh species name. Each entry in the list contains a matrix of sequences from the same species. |
| seqlength | Length of the nucleotide sequence. |

### Details

The argument species can be calculated using the separate function.

### Value

A matrix containing the number of singleton and shared ntVLFs in each position of the barcode.

### Author(s)

Taryn B. T. Athey and Paul D. McNicholas

### Examples

```
## Not run: #Nucleotide VLF analysis
data(birds)
species.names <- birds[,2]
specimen.Number <- nrow(birds)
rownames(birds) <- species.names
Nuc.count <- count.function(birds, specimen.Number, 648)
frequency.matrix <- ffrequency.matrix.function(Nuc.count, 648)
birdSpec.freq <- specimen.frequencies(frequency.matrix, birds, specimen.Number, species.names, 648)
Bird_specimen_VLFcount <- VLF.count.spec(birdSpec.freq, 0.001, 648)
bird_VLFconvert <- VLF.convert.matrix(birds, birdSpec.freq, 0.001, 648)
bird_VLFnuc <- VLF.nucleotides(bird_VLFconvert, birds, 648)
bird_VLFreduced <- VLF.reduced(bird_VLFnuc, Bird_specimen_VLFcount, 648)
bird_species <- separate(bird_VLFreduced)
birds_singleAndShared <- find.singles(bird_species, 648)
## End(Not run)
```

matched.codon.position

*Codon Position of Matching aa and ntVLFs*

## Description

Counts which codon positions of ntVLFs lead to the concordant aaVLF.

## Usage

```
matched.codon.position(matched)
```

## Arguments

| | |
|---|---|
| matched | A list of the nucleotide position of concordant ntVLFs and their associated aaVLFs. |

## Details

The argument matched can be calculated using the function overall.matched.

## Value

A vector containing the number of concordant VLFs caused by ntVLFs in each codon position.

## Author(s)

Taryn B. T. Athey and Paul D. McNicholas

## Examples

```
## Not run: #Nucleotide VLF analysis
data(birds)
species.names <- birds[,2]
specimen.Number <- nrow(birds)
rownames(birds) <- species.names
Nuc.count <- count.function(birds, specimen.Number, 648)
frequency.matrix <- ffrequency.matrix.function(Nuc.count, 648)
birdSpec.freq <- specimen.frequencies(frequency.matrix, birds, specimen.Number, species.names, 648)
Bird_specimen_VLFcount <- VLF.count.spec(birdSpec.freq, 0.001, 648)
bird_VLFconvert <- VLF.convert.matrix(birds, birdSpec.freq, 0.001, 648)
bird_VLFnuc <- VLF.nucleotides(bird_VLFconvert, birds, 648)
bird_VLFreduced <- VLF.reduced(bird_VLFnuc, Bird_specimen_VLFcount, 648)
bird_species <- separate(bird_VLFreduced)
birds_singleAndShared <- find.singles(bird_species, 648)

#Amino Acid VLF Analysis
data(birds_aminoAcids)
birds_aminoAcid_speciesNames <- birds_aminoAcids[,2]
```

```
aminoAcids_specimenNumber <- nrow(birds_aminoAcids)
birds_aminoAcid_count <- aa.count.function(birds_aminoAcids, 216)
aminoAcid_frequency.Matrix <- aa.frequency.matrix.function(birds_aminoAcid_count, 216)
bird_aminoAcid_frequencies <- aa.specimen.frequencies(aminoAcid_frequency.Matrix, birds_aminoAcids,
    birds_aminoAcid_speciesNames, 216)
aminoAcid_Modal <- aa.MODE(aminoAcid_frequency.Matrix, 216)
birds_aminoAcid_specimenVLFcount <- aa.VLF.count.spec(bird_aminoAcid_frequencies, 0.001, 216)
birds_aaVLFconvert <- aa.VLF.convert.matrix(birds_aminoAcids, bird_aminoAcid_frequencies, 0.001,
    216)
birds_aminoAcidVLFs <- VLF.aminoAcids(birds_aaVLFconvert, birds_aminoAcids, 216)
birds_aaVLFreduced <- aa.VLF.reduced(birds_aminoAcidVLFs, birds_aminoAcid_specimenVLFcount, 216)
birds_aaSpecies <- separate(birds_aaVLFreduced)
birds_aminoAcid_singleAndShared <- aa.find.singles(birds_aaSpecies, 216)

#Concordance Analysis
VLF_match <- find.matching(bird_VLFreduced, birds_aaVLFreduced, 648, 216)
position_matchingNuc <- nucleotide.matching.positions(VLF_match[[2]], 648)
position_matchingAA <- aminoAcid.matching.positions(VLF_match[[1]], 216)
matching_comparison <- overall.matched(position_matchingNuc, position_matchingAA, 648, 216)
matching_codons <- matched.codon.position(matching_comparison)
## End(Not run)
```

---

MODE *Modal Sequence*

---

## Description

Calculates the nucleotide sequence that occurs most often in a matrix of sequences.

## Usage

```
MODE(freq, seqlength)
```

## Arguments

| | |
|---|---|
| freq | Frequenct matrix for nucleotides. |
| seqlength | Length of nucleotide sequence. |

## Details

The argument freq can be calculated using the function ffrequency.matrix.function.

## Value

A vector containing the first modal sequence.

## Author(s)

Taryn B. T. Athey and Paul D. McNicholas

## Examples

```
## Not run: data(birds)
species.names <- birds[,2]
specimen.Number <- nrow(birds)
rownames(birds) <- species.names
Nuc.count <- count.function(birds, specimen.Number, 648)
frequency.matrix <- ffrequency.matrix.function(Nuc.count, 648)
birdSpec.freq <- specimen.frequencies(frequency.matrix, birds, specimen.Number, species.names, 648)
nucleotide.modalSequence <- MODE(frequency.matrix, 648)
## End(Not run)
```

---

# MODE.freq *Modal Frequencies*

---

## Description

Returns the frequencies of the nucleotides in each position of nucleotide sequence that occurs most often.

## Usage

```
MODE.freq(freq, seqlength)
```

## Arguments

| | |
|---|---|
| freq | Frequency matrix for nucleotides. |
| seqlength | Length of the nucleotide sequence. |

## Details

The argument freq can be calculated using the function ffrequency.matrix.function.

## Value

A vector of frequencies for the first modal sequence.

## Author(s)

Taryn B. T. Athey and Paul D. McNicholas

## Examples

```
## Not run: data(birds)
species.names <- birds[,2]
specimen.Number <- nrow(birds)
rownames(birds) <- species.names
Nuc.count <- count.function(birds, specimen.Number, 648)
frequency.matrix <- ffrequency.matrix.function(Nuc.count, 648)
```

```
birdSpec.freq <- specimen.frequencies(frequency.matrix, birds, specimen.Number, species.names, 648)
nucleotide.modalSequence <- MODE(frequency.matrix, 648)
Bird_first.modal.frequencies <- MODE.freq(frequency.matrix, 648)
## End(Not run)
```

---

`MODE.second.freq` *Second Modal Frequency*

---

## Description

Calculates the frequencies of the nucleotides that occur second most often in a matrix of sequences.

## Usage

```
MODE.second.freq(freq, seqlength)
```

## Arguments

| | |
|---|---|
| `freq` | Frequency matrix for nucleotides. |
| `seqlength` | Length of nucleotide sequences. |

## Details

The argument freq can be calculated using the function ffrequency.matrix.function.

## Value

A vector containing the frequencies of the nucleotide sequence that occurs second most often.

## Author(s)

Taryn B. T. Athey and Paul D. McNicholas

## Examples

```
## Not run: data(birds)
species.names <- birds[,2]
specimen.Number <- nrow(birds)
rownames(birds) <- species.names
Nuc.count <- count.function(birds, specimen.Number, 648)
frequency.matrix <- ffrequency.matrix.function(Nuc.count, 648)
birdSpec.freq <- specimen.frequencies(frequency.matrix, birds, specimen.Number, species.names, 648)
nucleotide.modalSequence <- MODE(frequency.matrix, 648)
Bird_first.modal.frequencies <- MODE.freq(frequency.matrix, 648)
Bird_second.modal.frequencies <- MODE.second.freq(frequency.matrix, 648)
## End(Not run)
```

## nucleotide.matching.positions

*Matching Nucleotide Positions*

### Description

Calculates the position of the VLFs in a matrix contain ntVLFs whose specimen identifiers match identifiers of a matrix containing aaVLFs.

### Usage

```
nucleotide.matching.positions(matchNuc, nuclength)
```

### Arguments

| | |
|---|---|
| matchNuc | A matrix containing only of the nucleotides that are VLFs and NAs in all other positions of the sequences. |
| nuclength | The length of the nucleotide sequence. |

### Details

The argument matchNuc can be calculated using the function find.matching.

### Value

A list for each ntVLF containing the specimen identifier in the first position of each list entry, the species name in the second position of each list entry, and the position of the ntVLF in the third position of each list entry.

### Author(s)

Taryn B. T. Athey and Paul D. McNicholas

### Examples

```
## Not run: #Nucleotide VLF analysis
data(birds)
species.names <- birds[,2]
specimen.Number <- nrow(birds)
rownames(birds) <- species.names
Nuc.count <- count.function(birds, specimen.Number, 648)
frequency.matrix <- ffrequency.matrix.function(Nuc.count, 648)
birdSpec.freq <- specimen.frequencies(frequency.matrix, birds, specimen.Number, species.names, 648)
Bird_specimen_VLFcount <- VLF.count.spec(birdSpec.freq, 0.001, 648)
bird_VLFconvert <- VLF.convert.matrix(birds, birdSpec.freq, 0.001, 648)
bird_VLFnuc <- VLF.nucleotides(bird_VLFconvert, birds, 648)
bird_VLFreduced <- VLF.reduced(bird_VLFnuc, Bird_specimen_VLFcount, 648)
bird_species <- separate(bird_VLFreduced)
birds_singleAndShared <- find.singles(bird_species, 648)
```

```
#Amino Acid VLF Analysis
data(birds_aminoAcids)
birds_aminoAcid_speciesNames <- birds_aminoAcids[,2]
aminoAcids_specimenNumber <- nrow(birds_aminoAcids)
birds_aminoAcid_count <- aa.count.function(birds_aminoAcids, 216)
aminoAcid_frequency.Matrix <- aa.frequency.matrix.function(birds_aminoAcid_count, 216)
bird_aminoAcid_frequencies <- aa.specimen.frequencies(aminoAcid_frequency.Matrix, birds_aminoAcids,
    birds_aminoAcid_speciesNames, 216)
aminoAcid_Modal <- aa.MODE(aminoAcid_frequency.Matrix, 216)
birds_aminoAcid_specimenVLFcount <- aa.VLF.count.spec(bird_aminoAcid_frequencies, 0.001, 216)
birds_aaVLFconvert <- aa.VLF.convert.matrix(birds_aminoAcids, bird_aminoAcid_frequencies, 0.001,
    216)
birds_aminoAcidVLFs <- VLF.aminoAcids(birds_aaVLFconvert, birds_aminoAcids, 216)
birds_aaVLFreduced <- aa.VLF.reduced(birds_aminoAcidVLFs, birds_aminoAcid_specimenVLFcount, 216)
birds_aaSpecies <- separate(birds_aaVLFreduced)
birds_aminoAcid_singleAndShared <- aa.find.singles(birds_aaSpecies, 216)

#Concordance Analysis
VLF_match <- find.matching(bird_VLFreduced, birds_aaVLFreduced, 648, 216)
position_matchingNuc <- nucleotide.matching.positions(VLF_match[[2]], 648)
## End(Not run)
```

---

`overall.matched` *Final Matching*

---

## Description

Compares the ntVLFs and aaVLFs with the same specimen identifier, and determines which ntVLFs are concordant with aaVLFs.

## Usage

```
overall.matched(positionNuc, positionAA, nuclength, aalength)
```

## Arguments

| | |
|---|---|
| `positionNuc` | A list containing the names of the specimen and the ntVLF positions for specimens that have both aaVLFs and ntVLFs. |
| `positionAA` | A list containing the names of the specimen, the aaVLF, and the position of the aaVLF for specimens that have both aaVLFs and ntVLFs. |
| `nuclength` | The length of the nucleotide sequence (should by 3X the length of the amino acid sequence) |
| `aalength` | The length of the amino acid sequence (should be 1/3 the length of the nucleotide sequence) |

## Details

The argument positionNuc can be calculated using the function nucleotide.matching.positions. The argument positionAA can be calculated using the function aminoAcid.matching.positions.

## Value

A list of each ntVLF containing the specimen identifier in the first position of each list entry, the species name in the second position of each list entry, the aaVLF in the third position of each entry, the amino acid position of the aaVLF in the fourth entry, and the codon position of the concordant ntVLF in each position of the entry. If multiple ntVLFs have concordance with one aaVLF, then that aaVLF may contain multiple entries in the list, one for each ntVLF.

## Author(s)

Taryn B. T. Athey and Paul D. McNicholas

## Examples

```
## Not run: #Nucleotide VLF analysis
data(birds)
species.names <- birds[,2]
specimen.Number <- nrow(birds)
rownames(birds) <- species.names
Nuc.count <- count.function(birds, specimen.Number, 648)
frequency.matrix <- ffrequency.matrix.function(Nuc.count, 648)
birdSpec.freq <- specimen.frequencies(frequency.matrix, birds, specimen.Number, species.names, 648)
Bird_specimen_VLFcount <- VLF.count.spec(birdSpec.freq, 0.001, 648)
bird_VLFconvert <- VLF.convert.matrix(birds, birdSpec.freq, 0.001, 648)
bird_VLFnuc <- VLF.nucleotides(bird_VLFconvert, birds, 648)
bird_VLFreduced <- VLF.reduced(bird_VLFnuc, Bird_specimen_VLFcount, 648)
bird_species <- separate(bird_VLFreduced)
birds_singleAndShared <- find.singles(bird_species, 648)

#Amino Acid VLF Analysis
data(birds_aminoAcids)
birds_aa_speciesNames <- birds_aminoAcids[,2]
aminoAcids_specimenNumber <- nrow(birds_aminoAcids)
birds_aminoAcid_count <- aa.count.function(birds_aminoAcids, 216)
aa_freq.Mat <- aa.frequency.matrix.function(birds_aminoAcid_count, 216)
bird_aa_freq <- aa.specimen.frequencies(aa_freq.Mat, birds_aminoAcids, birds_aa_speciesNames, 216)
aminoAcid_Modal <- aa.MODE(aa_freq.Mat, 216)
birds_aminoAcid_specimenVLFcount <- aa.VLF.count.spec(bird_aa_freq, 0.001, 216)
birds_aaVLFconvert <- aa.VLF.convert.matrix(birds_aminoAcids, bird_aa_freq, 0.001, 216)
birds_aminoAcidVLFs <- VLF.aminoAcids(birds_aaVLFconvert, birds_aminoAcids, 216)
birds_aaVLFreduced <- aa.VLF.reduced(birds_aminoAcidVLFs, birds_aminoAcid_specimenVLFcount, 216)
birds_aaSpecies <- separate(birds_aaVLFreduced)
birds_aminoAcid_singleAndShared <- aa.find.singles(birds_aaSpecies, 216)

#Concordance Analysis
VLF_match <- find.matching(bird_VLFreduced, birds_aaVLFreduced, 648, 216)
position_matchingNuc <- nucleotide.matching.positions(VLF_match[[2]], 648)
```

```
position_matchingAA <- aminoAcid.matching.positions(VLF_match[[1]], 216)
matching_comparison <- overall.matched(position_matchingNuc, position_matchingAA, 648, 216)
## End(Not run)
```

---

separate *Separate Specimen by Species Names*

---

## Description

Separates specimen into lists by species names.

## Usage

```
separate(x)
```

## Arguments

| | |
|---|---|
| x | A matrix of sequences, usually reduced sequences containing only VLFs, where the second position of the matrix contains the species name for the specimen. |

## Details

If the argument x needs to be a reduced matrix, it can be calculated using the function VLF.reduced.

## Value

A list containing a matrix of sequences for each species.

## Author(s)

Taryn B. T. Athey and Paul D. McNicholas

## Examples

```
## Not run: data(birds)
species.names <- birds[,2]
specimen.Number <- nrow(birds)
rownames(birds) <- species.names
Nuc.count <- count.function(birds, specimen.Number, 648)
frequency.matrix <- ffrequency.matrix.function(Nuc.count, 648)
birdSpec.freq <- specimen.frequencies(frequency.matrix, birds, specimen.Number, species.names, 648)
Bird_specimen_VLFcount <- VLF.count.spec(birdSpec.freq, 0.001, 648)
Bird_position_VLFcount <- VLF.count.pos(birdSpec.freq, 0.001, 648)
bird_VLFconvert <- VLF.convert.matrix(birds, birdSpec.freq, 0.001, 648)
bird_VLFnuc <- VLF.nucleotides(bird_VLFconvert, birds, 648)
bird_VLFreduced <- VLF.reduced(bird_VLFnuc, Bird_specimen_VLFcount, 648)
bird_species <- separate(bird_VLFreduced)
## End(Not run)
```

`Sliding.Window` *Sliding Window*

## Description

Creates a sliding window analysis plot for the VLFs in a matrix of sequences.

## Usage

```
Sliding.Window(VLF, seqlength, n = 30)
```

## Arguments

| | |
|---|---|
| `VLF` | A vector of VLFs per position across the barcocde. Can be a single vector of all VLFs per positions, or can be a matrix containing singleton and shared VLFs. |
| `seqlength` | Length of the barcode sequence. |
| `n` | The number of positions to average the window across (n = 30 by default). |

## Details

The argument VLF can be calculated using the function VLF.count.pos for all VLFs, or find.singles for singleton and shared VLFs.

## Value

A sliding window plot for the VLFs in each position of the barcode averaged over n.

## Author(s)

Taryn B. T. Athey and Paul D. McNicholas

## Examples

```
## Not run: data(birds)
species.names <- birds[,2]
specimen.Number <- nrow(birds)
rownames(birds) <- species.names
Nuc.count <- count.function(birds, specimen.Number, 648)
frequency.matrix <- ffrequency.matrix.function(Nuc.count, 648)
birdSpec.freq <- specimen.frequencies(frequency.matrix, birds, specimen.Number, species.names, 648)
Bird_position_VLFcount <- VLF.count.pos(birdSpec.freq, 0.001, 648)
Sliding.Window(Bird_position_VLFcount, 648)
## End(Not run)
```

## specimen.frequencies *Specimen Nucleotide Frequencies*

### Description

Converts a matrix of sequences into a matrix of nucleotide frequencies.

### Usage

```
specimen.frequencies(freq, seq.matrix, no.spec, spec.names, seqlength)
```

### Arguments

| | |
|---|---|
| freq | Frequency matrix for nucleotides. |
| seq.matrix | Matrix of specimen sequences, where the sequence starts in the third position of the matrix and the first and second position contain the unique specimen identifier and the species name, respectively. |
| no.spec | The number of specimen in seq.matrix. |
| spec.names | A vector containing the names of the specimen in the seq.matrix, in the order they appear in the matrix. |
| seqlength | The length of the nucleotide sequence. |

### Details

The argument freq can be calculated using the function ffrequency.matrix.function. The number of specimen can be calculated by using the nrow() function on seq.matrix.

### Value

A matrix containing the unique specimen identifer in the first position, the species name in the second position, and the frequencies for each nucleotide in the sequences starting at the third position.

### Author(s)

Taryn B. T. Athey and Paul D. McNicholas

### Examples

```
## Not run: data(birds)
species.names <- birds[,2]
specimen.Number <- nrow(birds)
rownames(birds) <- species.names
Nuc.count <- count.function(birds, specimen.Number, 648)
frequency.matrix <- ffrequency.matrix.function(Nuc.count, 648)
birdSpec.freq <- specimen.frequencies(frequency.matrix, birds, specimen.Number, species.names, 648)
## End(Not run)
```

## VLF.aminoAcids *Amino Acid VLFs*

### Description

Converts a matrix of amino acid frequencies into a matrix of amino acids.

### Usage

```
VLF.aminoAcids(convert.matrix, seq.matrix, seqlength)
```

### Arguments

| | |
|---|---|
| convert.matrix | A matrix consisting of only aaVLF frequencies for each specimen, and NAs in every other position of the sequence. |
| seq.matrix | A matrix of amino acid sequences. |
| seqlength | The length of the amino acid sequence. |

### Details

The argument convert.matrix can be calculated using the function aa.VLF.convert.matrix

### Value

A matrix containing only aaVLFs and NAs in every other position of the sequence.

### Author(s)

Taryn B. T. Athey and Paul D. McNicholas

### Examples

```
## Not run: data(birds_aminoAcids)
birds_aminoAcid_speciesNames <- birds_aminoAcids[,2]
aminoAcids_specimenNumber <- nrow(birds_aminoAcids)
birds_aminoAcid_count <- aa.count.function(birds_aminoAcids, 216)
aminoAcid_frequency.Matrix <- aa.frequency.matrix.function(birds_aminoAcid_count, 216)
bird_aminoAcid_frequencies <- aa.specimen.frequencies(aminoAcid_frequency.Matrix, birds_aminoAcids,
birds_aminoAcid_speciesNames, 216)
aminoAcid_Modal <- aa.MODE(aminoAcid_frequency.Matrix, 216)
birds_aminoAcid_specimenVLFcount <- aa.VLF.count.spec(bird_aminoAcid_frequencies, 0.001, 216)
birds_aaVLFconvert <- aa.VLF.convert.matrix(birds_aminoAcids, bird_aminoAcid_frequencies, 0.001,
    216)
birds_aminoAcidVLFs <- VLF.aminoAcids(birds_aaVLFconvert, birds_aminoAcids, 216)
## End(Not run)
```

## VLF.convert.matrix *VLF Matrix Convert*

### Description

Converts a matrix of nucleotide frequencies for each specimen into a matrix consisting entirely of very low frequency variant (VLF) frequencies and NAs in each other position.

### Usage

```
VLF.convert.matrix(seq.matrix, freq, p, seqlength)
```

### Arguments

| | |
|---|---|
| seq.matrix | A matrix of aligned DNA barcode sequences. |
| freq | A matrix of nucleotide frequencies for each specimen. |
| p | A very low frequency variant designation cut off frequency. Any frequency in the freq matrix below this value is considered to be a very low frequency variant. |
| seqlength | Length of nucleotide sequence. |

### Details

The argument freq can be calculated using the function specimen.frequencies.

### Value

A matrix of VLF nucleotide frequencies, containing only those nucleotide frequencies that occur less than the designation p value, and NAs in each other position of the matrix.

### Author(s)

Taryn B. T. Athey and Paul D. McNicholas

### Examples

```
## Not run: data(birds_aminoAcids)
birds_aminoAcid_speciesNames <- birds_aminoAcids[,2]
aminoAcids_specimenNumber <- nrow(birds_aminoAcids)
birds_aminoAcid_count <- aa.count.function(birds_aminoAcids, 216)
aminoAcid_frequency.Matrix <- aa.frequency.matrix.function(birds_aminoAcid_count, 216)
bird_aminoAcid_frequencies <- aa.specimen.frequencies(aminoAcid_frequency.Matrix, birds_aminoAcids,
    birds_aminoAcid_speciesNames, 216)
aminoAcid_Modal <- aa.MODE(aminoAcid_frequency.Matrix, 216)
birds_aminoAcid_specimenVLFcount <- aa.VLF.count.spec(bird_aminoAcid_frequencies, 0.001, 216)
birds_aaVLFconvert <- aa.VLF.convert.matrix(birds_aminoAcids, bird_aminoAcid_frequencies,0.001,216)
## End(Not run)
```

`VLF.count.pos` *VLF Count for Sequence Positions*

## Description

Calculates the number of very low frequency variants (VLFs) in each position in a matrix of sequences.

## Usage

```
VLF.count.pos(freq, p, seqlength)
```

## Arguments

| | |
|---|---|
| `freq` | A matrix of frequencies for each specimen. |
| `p` | A very low frequency variant designation cut off frequency. Any frequency in the freq matrix below this value is considered to be a very low frequency variant. |
| `seqlength` | The length of the sequences. |

## Details

The argument freq can be calculated using the specimen.frequencies function.

## Value

A vector containing the number of VLFs for each position in the sequence.

## Author(s)

Taryn B. T. Athey and Paul D. McNicholas

## Examples

```
## Not run: data(birds)
species.names <- birds[,2]
specimen.Number <- nrow(birds)
rownames(birds) <- species.names
Nuc.count <- count.function(birds, specimen.Number, 648)
frequency.matrix <- ffrequency.matrix.function(Nuc.count, 648)
birdSpec.freq <- specimen.frequencies(frequency.matrix, birds, specimen.Number, species.names, 648)
Bird_specimen_VLFcount <- VLF.count.spec(birdSpec.freq, 0.001, 648)
Bird_position_VLFcount <- VLF.count.pos(birdSpec.freq, 0.001, 648)
## End(Not run)
```

`VLF.count.spec` *VLF Count for Specimens*

## Description

Calculates the number of very low frequency variants (VLFs) for each specimen in a matrix of sequence nucleotide frequencies.

## Usage

```
VLF.count.spec(freq, p, seqlength)
```

## Arguments

| | |
|---|---|
| `freq` | A matrix of nucleotide frequencies for each specimen. |
| `p` | A very low frequency variant designation vut off frequency. Any frequency in the freq matrix below this value is considered to be a very low frequency variant. |
| `seqlength` | The length of the sequences. |

## Details

The argument freq can be calculated using the function specimen.frequencies.

## Value

A vector containing the number of VLFs for each specimen in the matrix.

## Author(s)

Taryn B. T. Athey and Paul D. McNicholas

## Examples

```
## Not run: data(birds)
species.names <- birds[,2]
specimen.Number <- nrow(birds)
rownames(birds) <- species.names
Nuc.count <- count.function(birds, specimen.Number, 648)
frequency.matrix <- ffrequency.matrix.function(Nuc.count, 648)
birdSpec.freq <- specimen.frequencies(frequency.matrix, birds, specimen.Number, species.names, 648)
Bird_specimen_VLFcount <- VLF.count.spec(birdSpec.freq, 0.001, 648)
## End(Not run)
```

## VLF.nucleotides *Nucleotide VLF Convert*

### Description

Converts a matrix of nucleotide frequencies for each specimen into a matrix of nucleotides for each specimen.

### Usage

```
VLF.nucleotides(convert.matrix, seq.matrix, seqlength)
```

### Arguments

| | |
|---|---|
| convert.matrix | A matrix consisting of only very low frequency cariant frequencies for each specimen, and NAs in all other positions of the sequence. |
| seq.matrix | A matrix of DNA sequences. |
| seqlength | The length of the sequences. |

### Details

The argument convert.matrix can be calculated using the function VLF.convert.matrix.

### Value

A matrix containing only ntVLFs in each position of the sequences, and NAs in all other positions.

### Examples

```
## Not run: data(birds)
species.names <- birds[,2]
specimen.Number <- nrow(birds)
rownames(birds) <- species.names
Nuc.count <- count.function(birds, specimen.Number, 648)
frequency.matrix <- ffrequency.matrix.function(Nuc.count, 648)
birdSpec.freq <- specimen.frequencies(frequency.matrix, birds, specimen.Number, species.names, 648)
Bird_specimen_VLFcount <- VLF.count.spec(birdSpec.freq, 0.001, 648)
bird_VLFconvert <- VLF.convert.matrix(birds, birdSpec.freq, 0.001, 648)
bird_VLFnuc <- VLF.nucleotides(bird_VLFconvert, birds, 648)
## End(Not run)
```

VLF.reduced *Reduced VLF Matrix*

## Description

Reduces a matrix of very low frequency variants (VLFs) so that only those specimen that contain VLFs remain in the matrix.

## Usage

```
VLF.reduced(NA.matrix, sCount, seqlength)
```

## Arguments

| | |
|---|---|
| NA.matrix | A matrix with values for very low frequency variants (VLFs) for each specimen and NAs in all other positions of the sequence. |
| sCount | A vector of the very low frequency variant (VLF) counts for each specimen in the NA.matrix. |
| seqlength | The length of the sequences. |

## Details

The argument NA.matrix can be calculated using the function VLF.convert.matrix and VLF.nucleotides. The argument sCount can be calculated using the function VLF.count.spec.

## Value

A matrix containing only those specimen that have VLFs, and only VLFs in their positions in the sequence, all other positions contain NAs.

## Author(s)

Taryn B. T. Athey and Paul D. McNicholas

## Examples

```
## Not run: data(birds)
species.names <- birds[,2]
specimen.Number <- nrow(birds)
rownames(birds) <- species.names
Nuc.count <- count.function(birds, specimen.Number, 648)
frequency.matrix <- ffrequency.matrix.function(Nuc.count, 648)
birdSpec.freq <- specimen.frequencies(frequency.matrix, birds, specimen.Number, species.names, 648)
Bird_specimen_VLFcount <- VLF.count.spec(birdSpec.freq, 0.001, 648)
bird_VLFconvert <- VLF.convert.matrix(birds, birdSpec.freq, 0.001, 648)
bird_VLFnuc <- VLF.nucleotides(bird_VLFconvert, birds, 648)
bird_VLFreduced <- VLF.reduced(bird_VLFnuc, Bird_specimen_VLFcount, 648)
## End(Not run)
```

vlfFun *Nucleotide VLF Assessment Function*

## Description

Runs the full nucleotide VLF analysis for the user and outputs the results.

## Usage

```
vlfFun(x, p = 0.001, seqlength = 648, own = NULL)
```

## Arguments

| | |
|---|---|
| x | A matrix of nucleotide sequences with the first column containing the unique specimen identifier, the second column containing the species name and the remaining columns containing the nucleotide sequence. |
| p | A VLF designation frequency cut-off to be used within the analysis. By default p = 0.001. |
| seqlength | The length of the nucleotide sequence. By default seqlength = 648. |
| own | If the user wants to compare their own sequences separate from reference sequences, then argument can be used. Similar to x, this argument is a matrix of nucleotide sequences with the first column containing the unique specimen identifier, the second column contains the species name and the remaining columns containing the nucleotide sequence. By default own = NULL.) |

## Value

| | |
|---|---|
| modal | A vector containing the nucleotide sequence that occurs most often in the dataset. |
| con100 | The number of nucleotide positions that are 100% conserved in the sequence, separated by codon position. |
| conp | The number of nucleotide positions that are (1-p)% conserved in the sequence, separated by codon position. |
| combine | The number of nucleotide positions that are (1-p)% conserved when combining the first and second modal sequences. |
| specimen | A vector containing the number of VLFs for each specimen in the dataset. |
| position | A vector containing the number of VLFs for each position in the sequences. |
| sas | A matrix containing vectors of single and shared ntVLF counts for each position in the sequences. |
| VLFmatrix | A matrix containing only those specimen that have VLFs as well as the nucleotides at the positions that contain VLFs and NAs in all other positions of the sequence. |
| ownSpecCount | A vector containing the number of VLFs for each specimen in the users own specified dataset. Only appears if own is not NULL. |

`ownPosCount` A vector containing the number of VLFs for each position in the sequences of the users own specified dataset. Only appears if own is not NULL.

`ownVLFMatrix` A matrix containing only those nucleotides at the positions that contain VLFs and NAs in all other positions of the sequence. Only appears if own is not NULL.

`ownVLFreduced` A matrix containing only those specimen that have VLFs as well as the nucleotides at the positions that contain VLFs and NAs in all other positions of the sequence. Only appears if own is not NULL.

## Author(s)

Taryn B. T. Athey and Paul D. McNicholas

## Examples

```
## Not run: data(birds)
bird_vlfAnalysis <- vlfFun(birds)
## End(Not run)
```

# Index


∗ **VLFs**
VLF-package, 3
∗ **datasets**
birds, 20
birds_aminoAcids, 20

aa.compare, 3
aa.conservation_first, 4
aa.conservation_two, 5
aa.count.function, 6
aa.find.singles, 7
aa.frequency.matrix.function, 8
aa.MODE, 9
aa.MODE.freq, 10
aa.MODE.second.freq, 10
aa.specimen.frequencies, 11
aa.VLF.convert.matrix, 12
aa.VLF.count.pos, 13
aa.VLF.count.spec, 14
aa.VLF.reduced, 15
aaVLFs.to.modalchanges, 16
aminoAcid.matching.positions, 17
aminoAcidFun, 18

birds, 20
birds_aminoAcids, 20

compare, 21
concordanceFun, 22
concordant.to.modalchanges, 23
conservation_first, 25
conservation_two, 26
count.function, 27

Decile.Plot, 28

Error.Rate, 29

fasta.read, 30
ffrequency.matrix.function, 30
find.matching, 31
find.singles, 33

matched.codon.position, 34
MODE, 35
MODE.freq, 36
MODE.second.freq, 37

nucleotide.matching.positions, 38

overall.matched, 39

separate, 41
Sliding.Window, 42
specimen.frequencies, 43

VLF *(VLF-package)*, 3
VLF-package, 3
VLF.aminoAcids, 44
VLF.convert.matrix, 45
VLF.count.pos, 46
VLF.count.spec, 47
VLF.nucleotides, 48
VLF.reduced, 49
vlfFun, 50