Package ‘VDAP’

Type Package
Title Peptide Array Analysis Tools
Version 2.0.0
Date 2016-04-26
Author Cody Moore
Maintainer Cody Moore <Jumper9400@gmail.com>
Description Analyze Peptide Array Data and characterize peptide sequence space. Allows for high level visualization of global signal. Quality control based on replicate correlation and/or relative Kd, calculation of peptide Length/Charge/Kd parameters, Hits selection based on RFU Signal, and amino acid composition/basic motif recognition with RFU signal weighting. Basic signal trends can be used to generate peptides that follow the observed compositional trends.
License GPL-2
Imports stringr, drc, ggplot2, reshape2
LazyData TRUE
NeedsCompilation no
Repository CRAN
Date/Publication 2016-05-22 22:54:19

R topics documented:

aaDist .......................................................... 2
aaStruct .......................................................... 3
Attrib ............................................................ 3
Dups ............................................................... 4
genPep ........................................................... 5
hitSel ............................................................. 6
KdA ............................................................... 7
lcScan ............................................................ 8
QCKd .............................................................. 9
QCon ............................................................. 11
aadist
Position Independent Amino Acid Distributions

Description
Generates Position Independent Amino Acid Distributions within VDAP data sets

Usage
aadist(x, plotName = NULL, linker = TRUE)

Arguments
- x: An R object, usually a data.frame generally created by the function fLoad()
- plotName: A plot title may be entered here surrounded by "quotations" or a class(character) object
- linker: Logical determining if a 3 residue linker "GSG" is present or not. If linker = TRUE, the "GSG" linker portion of each peptide will be excluded from distribution calculations. Default is FALSE.

Details
Uses both stringr and ggplot2 for peptide calculations and plotting

Value
aadist will return a data.frame that contains a table with the amino acid distribution over the entire array object. A ggplot2 object will also be displayed with the same information as the histogram.

Author(s)
Cody Moore

Examples
protEx <- data.frame(Peptides = c("PWGPWARVGSG","GYNRVQGSG","PWGPWARVGSG","GYNRVQGSG","GSG"))

## Plot example with GSG linker ##

aaDistEx <- aadist(protEx,"aadistEx Plot",linker = TRUE)
Positional Amino Acid Composition Calculations

**Description**

Calculates the probability of each amino acid residue at each position within a peptide. A sub-function of `vMotif` and `vComp`.

**Usage**

```r
aaStruct(x, y, sigWeight = TRUE)
```

**Arguments**

- `x` A `data.frame`, containing the peptides to be calculated
- `y` Object containing the signal set of interest for the defined peptides in argument `x`
- `sigWeight` Logical which determines if signal is incorporated into weight calculations

**Details**

A sub-function of `vMotif` and `vComp`

**Author(s)**

Cody Moore

**See Also**

`vComp`, `vMotif`

---

**Calculate Peptide Length and Charge Attributes**

**Description**

Calculates the length and charge of peptides in the first column of a given dataset. A sub-function of `vFormat`

**Usage**

```r
Attrib(x)
```

**Arguments**

- `x` An R object, generally a `data.frame`, containing peptides in the first column
**Value**

Returns a data.frame of 3 columns, starting with Peptide, the peptide's length, followed by charge.

**Note**

Uses the R Package: `stringr` created by Hadley Wickham

**Author(s)**

Cody Moore

**Examples**

```r
protEx <- data.frame(Peptides = c("PWRGPWARVGSG","GYNRVQGSG","PNGYRSGVGSG"),
                      C_6uM = c(65011.48,47462.24,24778),
                      C_3uM = c(62637.81,31899.85,21313.67),
                      C_1.5uM = c(57893.22,25911.35,10397.99))

attribEx <- Attrib(protEx)
```

---

**Dups**

*Average duplicated peptides from a VDAP dataset*

**Description**

Looks for duplicate peptides in the first column of the dataset, averages the signal of duplicates and replaces them with a single row. A subfunction of vFormat

**Usage**

`Dups(x)`

**Arguments**

- `x` An R object, generally a data.frame with peptides in column 1, followed by signal values at various concentrations.

**Value**

Returns a data.frame without duplicated peptides. Duplicate entries display the mean of the signal at each concentration

**Note**

Duplicated peptide entries will generally be at the top of the dataset
**Description**

Generates the specified number of peptides whose positional composition is determined by a weighted matrix given by the VDAP functions vcomp or vmotif.

**Usage**

```
genPep(Struct, draw)
```

**Arguments**

- **Struct**: The output positional weight matrix from the VDAP functions vcomp or vmotif
- **draw**: An integer value, the number of peptides to be generated

**Details**

The final composition of residues at each position should reflect the relative weight present in the argument Struct, as the relative weights at each position are used to weight the sampling of amino acids at each position.

**Value**

A `data.frame` containing the number of peptides given by the argument draw in a single column.

**Note**

The weighted values are squared before being used to weight random residue draws at each position. This is done in order to further penalize peptides that appear less frequently than the global distribution (Have weights < 1), and enrich peptides that appear more often than the global distribution (Have weights > 1).
hitSel

Signal Based Hits Selection for VDAP

Description

Filters the dataset based upon signal from the specified columns. Can be normalized to the average signal of any given peptide at the given concentration. Works for multiple RFU signal inputs or a single Kd input.

Usage

hitSel(File, AvgSet, CutOff, Kd = FALSE)

Arguments

File
An R object, usually a data.frame generally created by the function FLoad()

AvgSet
An integer sequence, defines the columns that contain the concentration data to be used for hits selection. A given peptide will have to qualify as a hit at all given concentration columns to be considered a true peptide hit. Ex: Hits based upon 3 concentrations in columns 5 through 8 = 5:8. If Kd = TRUE, then a single column with the calculated Kd values (generally column 4 created by vFormat) should be entered.

CutOff
A character string that defines the peptide to to normalize to. Hits must be 5 times higher in signal than the given peptide to be returned as hits. Normally "GSG". If Kd = TRUE, hits will be defined as peptides that have a calculated Kd less than one half of the CutOff peptide.

Examples

protEx.Motif <- data.frame(Peptides = c("PWRGPKARVGSG","GYNRVGQGSG","PNGYRSVKGSG","GSG"), Length = c(12,10,12,3),Charge = c(2,1,2,0),Kd = c(0.2572361,2.8239730,3.3911868,281.3058), C_6uM = c(65011.48,47462.24,24778,2613.03), C_6uM2 = c(62637.81,20723.85,21313.67,2300.216))

## Output weighted matrix generated by vMotif ##

vMotif.lcEx <- vMotif.lc(protEx.Motif,protEx.Motif, 12,2,5,Kd = FALSE)

## Generation of 10 peptides based on vMotif matrix weights##

genPepEx <- genPep(vMotif.lcEx,10)
Kd

Toggle that determines if hits will be selected by RFU signal or Kd values. If Kd = TRUE, hits will be defined as peptides that have a calculated Kd less than one half of the Cutoff peptide.

Value

A data.frame will be returned only with the peptides that are hits in the given context. (Hits must have Avg signal 5 times greater than the average signal of the peptide specified in the argument Cutoff. Or one fifth (0.2) the Cutoff Kd value if Kd = TRUE)

Author(s)

Cody Moore

Examples

protEx.hitSel <- data.frame(Peptides = c("PWRRPGVARVSG", "GYNRVRGQGSG", "PNGYRSGVKGSG", "GSG"),
    Kd = c(0.2572361, 2.8239730, 3.3911865, 281.3858),
    C_GuM = c(65011.48, 47462.24, 24778.2613, 0.03),
    C_3uM = c(62637.81, 31899.85, 21313.67, 1161.216),
    C_1.5uM = c(97893.22, 25911.35, 10397.99, 630.4025))

## Hits selection by RFU signal ##

hitSelRFU <- hitSel(protEx.hitSel, 3:5, "GSG", Kd = FALSE)

## Hits selection by calculated Kd ##

hitSelKd <- hitSel(protEx.hitSel, 2, "GSG", Kd = TRUE)

---

KdA Peptide Dissociation Rate Constant (Kd) Calculations

Description

Calculates the Kd of each peptide using a non-linear single site specific binding model. A sub-function of vFormat

Usage

KdA(x, y, z)

Arguments

x An R object, generally a data.frame, containing peptides in the first column

y The concentrations of each column used for Kd calculations, separated by commas. The order must match the relative position of the columns.

z The columns used for Kd calculations, expressed as a sequence. Ex: Columns 2 through 4 = 2:4
**lcScan**

**Signal or Kd Distributions separated by Length/Charge attributes**

**Description**

Calculates the mean with standard error, and population peptides at each length/charge combination within a VDAP dataset. If the argument Glob = TRUE, average signals will be compared against a global set of peptides and p-values will be calculated for hypothesis testing. lcScan will also return a plot for visualization of signal, population, and hypothesis testing.

**Usage**

```
lcScan(File, Glob = NULL, Conc = 5, Kd = FALSE)
```

**Arguments**

- **File**
  
  An R object, usually a data.frame generally created by the function FLoad()

- **Glob**
  
  A second data.frame with the global set of peptides. If the original File argument contains peptides hits, Glob should contain the dataset before hits were filtered out.

- **Conc**
  
  The column containing the concentration or Kd data to be analyzed, an integer. Default is column 5 which is generally the highest concentration average according to the default formatting function vFormat

  Ex: Column 1 = 1

- **Kd**
  
  Toggle to calculate by a defined signal column or by calculated Kd values, affects final plot behavior and labels. If Kd = TRUE, then the argument Conc should be set to 4 if the file was formatted by the default formatting function vFormat.

**Examples**

```r
protEx <- data.frame(Peptides = c("PWRGPWARVGSG","GYNRVGQGSG","PWRGPWARVGSG"),
C_6uM = c(65011.48,47462.24,24778), C_3uM = c(62637.81,31899.85,21313.67),
C_1.5uM = c(57893.22,25911.35,10397.99))

exKdA <- KdA(protEx,c(6,3,1.5),2:4)
```
QCKd

Value

A data.frame will be returned with columns for the mean, standard error, and population of peptides at each length/charge combination that can be exported for further analysis. Also utilizes ggplot2 and reshape2 to create a heat map plot that shows the signal distribution with corresponding populations that can be exported.

Author(s)

Cody Moore

References

Plot generation utilizes ggplot2 created by Hadley Wickham [aut, cre] and Winston Chang [aut] and reshape2 created by Hadley Wickham

Examples

```r
protEx.lcScan <- data.frame(Peptides = c("PWRGPWARVGSG","GYNRVQGSG","PNGYRSGVKGSG","GSG"),
                          Length = c(12,10,12,3),Charge = c(2,1,2,8),Kd = c(0.2572361,2.8239730,3.3911868,281.3058),
                          C_6uM = c(65011.48,47462.24,24778,2613.03),C_6uM2 = c(62637.81,20723.85,21313.67,2300.216))

## Signal length/charge Analysis ##

lcScanEx <- lcScan(protEx.lcScan)

## Kd length/charge Analysis ##

lcScanEx <- lcScan(protEx.lcScan, Conc = 4, Kd = TRUE)
```

QCKd

Quality Control of Peptides Based on Reproducibility and Kd

Description

Filter out peptides based on reproducibility between replicate concentrations and relative dissociation constants (Kd). Peptides must have a signal ratio between 0.5 and 2.0. A second reference file may be loaded with the same peptides referenced against another sample. Peptides are then compared based upon relative Kd value which must be at least one log10 apart.

Usage

```r
QCKd(File1, File2 = NULL, Kd = FALSE, QC = TRUE, ColSet1 = NULL, 
     ColSet2 = NULL, ColSet3 = NULL)
```
Arguments

File1: An R object, usually a data.frame generally created by the function FLoad().

File2: An R object, usually a data.frame generally created by the function FLoad().

Kd: A logical value, if Kd = TRUE then peptides will be filtered by Kd against the argument File2.

QC: A logical value, if QC = TRUE then peptides will be filtered by ratios of signal between replicates. Ratios must be between 0.5 to 2.0 to remain in the dataset.

ColSet1: A sequence value, represents the two columns that are replicates at a single concentration. Peptides must fit QC criteria in all given ColSets to remain in the dataset. ColSets may be omitted if less than three concentrations are to be compared. Ex: 2:3

ColSet2: A sequence value, represents the two columns that are replicates at a single concentration. Peptides must fit QC criteria in all given ColSets to remain in the dataset. ColSets may be omitted if less than three concentrations are to be compared. Ex: 6:7

ColSet3: A sequence value, represents the two columns that are replicates at a single concentration. Peptides must fit QC criteria in all given ColSets to remain in the dataset. ColSets may be omitted if less than three concentrations are to be compared. Ex: 4:5

Details

Either the QC or Kd filter may be applied by itself or both simultaneously.

Value

A data.frame will be returned with peptides filtered out that do not meet the given criteria for either the QC or Kd filters.

Author(s)

Cody Moore

Examples

```r
protEx.QCKd <- data.frame(Peptides = c("PWGRGPRWRGSG","GYNRVRGQGSG","PNGYRSGQKGSG","GSG"),
Length = c(12,10,12,3),Charge = c(2,1,2,0),Kd = c(0.2572361,2.8239730,3.3911868,281.3058),
C_6uM = c(65011.48,47462.24,24778,2613.03),C_6uM2 = c(62637.81,20723.85,21313.67,2300.216))

## All peptides filtered out due to same Kd value between files ##
QCKdEx <- QCKd(protEx.QCKd, protEx.QCKd.Kd = TRUE, QC = TRUE, ColSet1 = 5:6)

## QC control only ##
QCKdEx <- QCKd(protEx.QCKd, QC = TRUE, ColSet1 = 5:6)
```
QCon

*Subsetting for VDAP function QCKd*

**Description**
A sub-function of QCKd, subsets data for replicate control

**Usage**

```r
QCon(File1, ColSet)
```

**Arguments**

- `File1` Input File.
- `ColSet` ColSet (Same as QCKd)

**Author(s)**

Cody Moore

**See Also**

QCKd

**Examples**

```r
## The function is currently defined as
function(File1, ColSet){
  Sig <- File1[, min(ColSet)]  # Column Calls
  Sig2 <- File1[, max(ColSet)]
  FVar1 <- File1[Sig/Sig2 > 0.5 & Sig/Sig2 < 2.0,]
  FVar1 <- na.omit(FVar1)
  return(FVar1)
}
```

**resSep**

*Select Peptides with the Specified Amino Acid Residue(s) at an Indicated Position*

**Description**

Allows the experimenter to subset peptide data based on a selected amino acid residue or sequence a specified position(s). Requires the experimenter to select the residue(s) and position(s) of interest at a given length or length/charge combination.
Usage

```r
resSep(File, Length, Charge = NULL, Pos, Res)
```

Arguments

- **File**
  - An object, generally a `data.frame`, the `vformat` object with peptide and signal data.
- **Length**
  - An integer, the desired length of the peptides to separate.
- **Charge**
  - An integer, the desired charge of the peptides to separate. Defaults to `Charge = NULL`, which carries out length separation only.
- **Pos**
  - An integer or sequence, the position(s) to check for the residue(s) of interest.
- **Res**
  - A character input. The residue(s) to check for at the given position(s). The lengths of the arguments `Pos` and `Res` must match. Multiple residues are entered as a single character string. Ex: `Res = "RA"`.

Details

The lengths of the arguments `Pos` and `Res` must match.

Sequence Positions are read from right to left.

Ex: The residue "R" in 5-mer sequence "RSGSG" is at position 5.

When typing in a sequence of interest, it will be in reverse with regard to the displayed sequence.

Ex: Sequence "SR" at positions 4:5 in the 5-mer"RSGSG"

Value

A `data.frame` of the same format as the argument `File` containing only peptides that contain the specified residue(s) at the indicated position(s).

Author(s)

Cody Moore

See Also

`vsep`

Examples

```r
## Example data.frame ##

protEx.resSep <- data.frame(Peptides = c("PWPGPWARVGSG","GYRVSQGQGSG","PNGYRSVQKGSG","GSG"),
                           Length = c(12,10,12,3),Charge = c(2,1,2,0),Kd = c(0.2572361,2.8239730,3.3911868,281.3058),
                           C_6uM = c(65011.48,47462.24,24778,2613.03),C_6uM2 = c(62637.81,20723.85,21313.67,2300.216))

## Single Residue Separation ##

resSepEx1 <- resSep(protEx.resSep,12,2,5,"R")
```
### Positional Sequence Separation

```r
resSepEx2 <- resSep(resSepEx12, 2, 5:6, c("RA"))
```

---

**Description**

Generates the probability of each amino acid to appear in each position within a peptide of a specific length or length/charge combination. Can either be the raw probability or the ratio between the probabilities of 2 peptide sets.

Weights are centered at 1, meaning that there is no change in probability or signal from the global set. Weights above 1 indicate higher probability at the given position while weights below 1 indicate lower probability at the given position.

**Usage**

```r
vComp.lc(Prot, ProtG, Length, Charge)
```

```r
vComp.l(Prot, ProtG, Length)
```

**Arguments**

- **Prot**: An R object, generally a data.frame. Contains peptides that are considered "hits" or selected peptides with their length, charge, and signal information.
- **ProtG**: An R object, generally a data.frame. Contains the set of peptides from which the argument Prot were selected with their corresponding length, charge, and signal information.
- **Length**: An integer value, indicating the desired peptide length to analyze
- **Charge**: An integer value, indicating the desired charge to analyze

**Details**

If raw probabilities are desired, the same object can be loaded into both the Prot and ProtG arguments.

**Value**

Returns a data.frame that shows weights for each amino acid at each position within the peptide of the selected length. Also output a positional heatmap using the package `ggplot2`

**Author(s)**

Cody Moore
See Also

vMotif, genPep

Examples

protEx.Motif <- data.frame(Peptides = c("PWPGWAVGS", "GYNRVGVGSG", "PNGYRSGVKGSG", "GSG"),
Length = c(12, 10, 12, 3), Charge = c(2, 1, 2, 0), Kd = c(0.257361, 2.8239730, 3.3911868, 281.3058),
C_6uM = c(65011.48, 47462.24, 24778.03, 62133.06), C_6uM2 = c(62637.81, 20723.85, 21313.67, 23000.216))

## Length/Charge Example ##

vComp.lcEx <- vComp.lc(protEx.Motif, protEx.Motif, 12, 2)

## Length Example ##

vComp.lEx <- vComp.l(protEx.Motif, protEx.Motif, 12)

---

vFormat

Length/Charge/Kd Peptide Calculations and File Assembly

Description

Calculates the length, charge, and dissociation rate constant (Kd) for each peptide and assembles
the file into a universal format for subsequent VDAP Functions.

Usage

vFormat(x, Kd = FALSE, Concs, Cols)

Arguments

x An R object, usually a data.frame generally created by the function FLoad()

Kd Toggle to specify if dissociation rate constants (Kd) values should be calculated.
If Kd = FALSE, the nonlinear regression package drc will not be used.

Concs The concentrations of each column used for Kd calculations, separated by com-
mas. The order must match the relative position of the columns.

Cols The columns used for Kd calculations, expressed as a sequence. Ex: Columns 2
through 4 = 2:4

Details

The order of concentrations should not matter, as long as they are identical between the Concs and
Cols arguments. However, the columns must all be adjacent.
vFormat

Value

A data frame will be returned with the Length, charge, and Kd if Kd = TRUE characteristics placed in columns 2 - 4, followed by the signal at each concentration from the x argument. This is followed by quality values such as std.error, p-value, and t-value from the Kd of each peptide. Peptides will remain in column 1.

Note

Uses the R Package: stringr created by Hadley Wickham and drc created by Christian Ritz and Jens C. Strebig

Author(s)

Cody Moore

See Also

Dups, Attrib, KdA.

Examples

```r
## vFormat on example data set ##

protEx <- data.frame(Peptides = c("PWRGPWARVGSG","GYNRVQGQGSG","PNGYRSGVKGSG"),
                      C_6uM = c(65011.48,47462.24,24778),
                      C_3uM = c(62637.81,31899.85,21313.67),
                      C_1.5uM = c(57893.22,25911.35,10397.99))

## Preformatted protEx ##

Peptides      C_6uM    C_3uM    C_1.5uM
#1  PWGPRWARVGSG 65011.48 62637.81 57893.22
#2     GYNRVQGQGSG  47462.24 31899.85 25911.35
#3       PNGYRSGVKGSG   24778.00 21313.67 10397.99

formatEx <- vFormat(protEx,Kd = TRUE, c(6,3,1.5), 2:4)

## Formatted output ##

Peptide Length Charge     Kd  C_6uM  C_3uM  C_1.5uM Std..Dev t.value  p.value
#1  PWGPRWARVGSG        12  2  0.2572361 65011.48 62637.81 57893.22 0.008441968 30.471112 0.02088587
#2     GYNRVQGQGSG       10  1 2.8239730  47462.24 31899.85 25911.35 1.619385359 1.743855 0.33146423
#3       PNGYRSGVKGSG      12  2 3.3911868 24778.00 21313.67 10397.99 2.522251940 1.344508 0.48711826
```
Description

Generate signal weighted amino acid composition maps by postion at specific length or length/charge combinations. Weights are compared to the global distribution of peptides at the particular length or length/charge.

Weights are centered at 1, meaning that there is no change in probability or signal from the global set. Weights above 1 indicate higher probability at the given position and/or signal while weights below 1 indicate lower probability at the given position and/or signal.

When Kd = TRUE, weighting by Kd instead of signal is performed. Weights are generated using (1/Kd) since lower Kd values generally indicate higher affinity interactions, and would correlate with higher signal.

Usage

vmotifNlcHprotL protgL lengthL chargeL sigcolL kd ] falseI

vmotifNlHprotL protgL lengthL sigcolL kd ] falseI

Arguments

Prot An R object, generally a data.frame. Contains peptides that are considered "hits" or selected peptides with their length,charge, and signal/Kd attributes.
ProtG An R object, generally a data.frame. Contains the set of peptides from which the argument Prot were selected with their corresponding length, charge, and signal information.
Charge An integer value, indicating the desired charge to analyze
Length An integer value, indicating the desired peptide length to analyze
SigCol An Integer value, indicating the column that contains the desired signal data at a given concentration
Kd An logical value, indicating if weights should be generated using signal or Kd data. Effects signal weighting behavior. If Kd = TRUE, weights are generated using l/SigCol.

Value

Returns a data.frame that shows weights for each amino acid at each position within the peptide of the selected length. Also output a positional heatmap using the package ggplot2

Author(s)

Cody Moore
vSep

See Also
vComp
Examples

```r
protEx.Motif <- data.frame(Peptides = c("PWPGPKWAVGSG","GYNRYGQGSG","PNGYRGYKGSG","GSG"),
Length = c(12,10,12,3),Charge = c(2,1,2,0),Kd = c(0.2572361,2.8239730,3.3911868,281.3058),
C_6uM = c(65011.48,47462.24,24778,2613.03),C_6uM2 = c(62637.81,20723.85,21313.67,2300.216))

## vMotif Length/Charge and Length Signal Examples ##
vMotif.lcEx <- vMotif.lc(protEx.Motif,protEx.Motif, 12,2,5,Kd = FALSE)
vMotif.lEx <- vMotif.l(protEx.Motif,protEx.Motif, Length = 12,SigCol = 5,Kd = FALSE)
## vMotif Length/Charge Kd Example ##
vMotif.lcEx <- vMotif.lc(protEx.Motif,protEx.Motif, Length = 12,Charge = 2, SigCol = 5,Kd = TRUE)
```

---

vSep  

Select Peptides of a Particular Length/Charge Combination

Description

Select Peptides that have a specified length/charge combination, a subfunction for lcScan, and all methods of LCMotif and LcComp

Usage

```
vSep(File, Length = NULL, Charge = NULL)
```

Arguments

- **File**: An R object, usually a `data.frame` generally created by the function `FLoad()`
- **Length**: An integer value, specifies the desired length to subset.
- **Charge**: An integer value, specified the desired charge to subset.

Value

Returns a `data.frame` with peptides of the selected Length/Charge combination.

Author(s)

Cody Moore
Examples

protExChargeSep <- data.frame(Peptides = c("PWRGPWARVGSG","GYNRVGQGSG","PWRGPWARVGSG"),
                              Length = c(12,10,12), Charge = c(2,1,2))

## Length/Charge Combination ##
hitSelEx <- vSep(protExChargeSep,10,1)

## Charge only ##
hitSelEx <- vSep(protExChargeSep,Charge = 1)

## Length Only ##
hitSelEx <- vSep(protExChargeSep,Length = 12)
Index

*Topic \textasciitildeComposition
  vComp, 13
*Topic \textasciitildeComp
  vComp, 13
*Topic \textasciitildeFormatting
  vFormat, 14
*Topic \textasciitildeKd
  hitSel, 6
  QCKd, 9
*Topic \textasciitildeMotif
  vMotif, 16
*Topic \textasciitildePeptides
  genPep, 5
*Topic \textasciitildeReplicates
  QCKd, 9
*Topic \textasciitildeResidue
  resSep, 11
*Topic \textasciitildeSeparation
  resSep, 11
  vSep, 17
*Topic \textasciitildeVDAP
  vFormat, 14
*Topic \textasciitildeattributes
  Attrib, 3
*Topic \textasciitildecomposition
  aaDist, 2
  aaStruct, 3
*Topic \textasciitildeDistributions
  aaDist, 2
*Topic \textasciitildeDuplicate
  Dups, 4
*Topic \textasciitildeFiltering
  vSep, 17
*Topic \textasciitildeFormatting
  Attrib, 3
  Dups, 4
  KdA, 7
*Topic \textasciitildeDegenerate
  genPep, 5

*Topic \textasciitildeHits
  hitSel, 6
*Topic \textasciitildeKinetics
  KdA, 7
*Topic \textasciitildePlot
  lcScan, 8
*Topic \textasciitildeScan
  lcScan, 8
*Topic \textasciitildeSub-function
  QCon, 11
*Topic \textasciitildeSubfunction
  aaStruct, 3
*Topic \textasciitildeSubset
  QCon, 11

aaDist, 2
aaStruct, 3
Attrib, 3, 15
Dups, 4, 15
genPep, 5
hitSel, 6
KdA, 7, 15
lcScan, 8
QCKd, 9
QCon, 11
resSep, 11
vComp, 13
vFormat, 14
vMotif, 16
vSep, 17