The PEP.db User Guide

Renan Sauteraud*

April 23, 2014

Contents

1 Introduction

`PEP.db` is an R package that stores peptide collections and provide functions for their analysis. It is intended to be used in conjunction with other packages for peptide analysis and visualisation: `HIV.db`, `pepStat` anf `Pviz`.

As with any R package, it should first be loaded in the session

```
library(PEP.db)
```

2 Peptide collections

2.1 Information

For each peptide, the following collections display information about the position relative to the reference sequence, the alignment, the trimmed alignment, the zSums for the physicochemical properties as well as

*rsautera@fhcrc.org

the clades where they can be found.

2.2 Structure

The datasets in this package are `RangedData` objects. For more information about the class, its accessors and setters, please refer to `IRanges` documentation.

2.3 pep_hxb2

This collection is based on the alignment of the reference HIV sequence hxb2 and seven subtypes (clades) A, B, C, D, M, CRF01 and CRF02.

2.4 pep_hxb2JPT

This collection adds a few more clades to the previous one: CM244, CON_01_AE, LAI_A04321 and MN_DD328842.

2.5 pep_mac239

This collection is for SIV, with the clades mac239 and E660.

2.6 pep_m239smE543

This collection is for SIV, with the clades mac239 and E543.

3 Loading the collections

The peptide collections can be loaded like any other R dataset.

```
data(pep_hxb2)
head(pep_hxb2)
```

```
## RangedData with 6 rows and 10 value columns across 1 space
##                    space    ranges |           names             aligned
##                 <factor> <IRanges> |     <character>            <factor>
## MRVKETQMNWPNLWK    gp160   [1, 16] | MRVKETQMNWPNLWK MRVKETQMNWPNL----WK
## MRVMGIQKNYPLLWR    gp160   [1, 16] | MRVMGIQKNYPLLWR MRVMGIQKNYPLL----WR
## MRVMGIQRNCQHLWR    gp160   [1, 16] | MRVMGIQRNCQHLWR MRVMGIQRNCQHL----WR
## MRVKGIRKNYQHLWR    gp160   [1, 16] | MRVKGIRKNYQHLWR MRVKGIRKNYQHL----WR
## MRVRGILRNWQQWWI    gp160   [1, 16] | MRVRGILRNWQQWWI MRVRGILRNWQQW----WI
## MRVRGIERNYQHLWR    gp160   [1, 16] | MRVRGIERNYQHLWR MRVRGIERNYQHL----WR
##                          trimmed     seqNb       clade        z1        z2
##                         <factor> <integer> <character> <numeric> <numeric>
## MRVKETQMNWPNLWK MRVKenWPNL----WK         1       CRF01     -3.14      9.87
## MRVMGIQKNYPLLWR MRVMgnYPLL----WR         1       CRF02    -13.12      3.19
## MRVMGIQRNCQHLWR MRVMgnCQHL----WR         1           A      1.65      4.17
## MRVKGIRKNYQHLWR MRVKgnYQHL----WR         1           B      3.95      9.78
```

```
## MRVRGILRNWQQWWI MRVRgnWQQW----WI         1           C    -11.42     10.26
## MRVRGIERNYQHLWR MRVRgnYQHL----WR         1           D      6.00     10.76
##                        z3        z4        z5
##                   <numeric> <numeric> <numeric>
## MRVKETQMNWPNLWK     -8.14      6.72      1.36
## MRVMGIQKNYPLLWR    -11.96      7.09      0.76
## MRVMGIQRNCQHLWR    -11.22     10.31     -2.99
## MRVKGIRKNYQHLWR    -18.51     12.55     -0.87
## MRVRGILRNWQQWWI    -16.19     10.33     -2.37
## MRVRGIERNYQHLWR    -17.14      8.52     -1.91
```

4 Create collections

While the package comes with datasets for HIV and SIV. It is possible to create new collections for different organisms or proteins.

4.1 Requirements

Minimum information required to create a peptide collection:

- start
- end or width
- peptide sequence

4.2 Constructors

Peptide collections being `RangedData` objects, the RangedData contructor should be used to generate a collection with at least the required information.

```
library(IRanges)
AA <- c("A", "C", "D", "E", "F", "G", "H", "I", "K", "L", "M", "N", "P", "Q",
    "R", "S", "T", "V", "W", "Y")
starts <- seq(1, 30, 3)
ends <- starts + 14
peptides <- sapply(1:10, function(x) {
    paste0(AA[floor(runif(15, 1, 20))], collapse = "")
})
data <- data.frame(start = starts, end = ends, peptide = peptides)
newRD <- RangedData(data)
```

Then the function `create_db` in `pepStat` should be used to add the Zscore information to the newly constructed RangedData object. This is required for the normalization of `peptideSet`.

```
library(pepStat)
new_pep <- create_db(newRD)
```

Any additional information will be passed as a new column in the `RangedData` objects and later in the `peptideSet` object.

```
data <- data.frame(start = starts, end = ends, peptide = peptides, var1 = rnorm(10))
newDB <- RangedData(data)
new_pep <- create_db(newDB)
```