Alignments

VJC

August 14, 2007

Contents

1 Introduction

In this document we illustrate import of FASTA files, simple alignments, and consensus matrix calculation.

2 Data import

The *Biostrings* package includes FASTA files encoding the DNA sequence for Yeast chromosome I, along with some ORFs. Here we load the `someORF.fsa` file into a *BStringViews* object using the `BStringViews` method:

```
> library(Biostrings)
> file <- system.file("Exfiles", "someORF.fsa", package = "Biostrings")
> orf <- BStringViews(file(file), "DNAString")
> orf

  Views on a 26339-letter DNAString subject
Subject: ACTTGTAAATATATCTTTTATTTTCCGAGAGGAA...TATACATAGGGCTAAGGAAGAAAAAAAAATCAC
Views:
    start   end width
[1]     1  5573  5573 [ACTTGTAAATATATCTTTTATTTTCC...ACGCTTATCGACCTTATTGTTGATAT]
```

```
[2]  5574 11398  5825 [TTCCAAGGCCGATGAATTCGACTCTT...CAGAGTAAATTTTTTTCTATTCTCTT]
[3] 11399 14385  2987 [CTTCATGTCAGCCTGCACTTCTGGGT...CGATGGTACTCATGTAGCTGCCTCAT]
[4] 14386 18314  3929 [CACTCATATCGGGGGTCTTACTTCCC...ACGTGTCCCGAAACACGAAAAAGTAC]
[5] 18315 20962  2648 [AGAGAAAGAGTTTCACTTCTTGATTA...AAAATATAATTTATGTGTGAACATAG]
[6] 20963 23559  2597 [GTGTCCGGGCCTCGCAGGCGTTCTAC...TTCAAGTTTTGGCAGAATGTACTTTT]
[7] 23560 26339  2780 [CAAGATAATGTCAAAGTTAGTGGTCG...AGGGCTAAGGAAGAAAAAAAAATCAC]
```

```
> desc(orf)
```

```
[1] ">YAL001C TFC3 SGDID:S0000001, Chr I from 152168-146596, reverse complement, Verifi
[2] ">YAL002W VPS8 SGDID:S0000002, Chr I from 142709-148533, Verified ORF"
[3] ">YAL003W EFB1 SGDID:S0000003, Chr I from 141176-144162, Verified ORF"
[4] ">YAL005C SSA1 SGDID:S0000004, Chr I from 142433-138505, reverse complement, Verifi
[5] ">YAL007C ERP2 SGDID:S0000005, Chr I from 139347-136700, reverse complement, Verifi
[6] ">YAL008W FUN14 SGDID:S0000006, Chr I from 135916-138512, Verified ORF"
[7] ">YAL009W SPO7 SGDID:S0000007, Chr I from 134856-137635, Verified ORF"
```

Note that some of the ORF sequences are represented in reverse complement form.

3 Alignments

A very simple implementation of the Needleman-Wunsch global alignment algorithm is provided. This takes space proportional to the square of the string length, uses a simple scalar gap penalty, and requires a substitution scoring matrix appropriate for the alphabets from which the strings to be aligned are constructed. Pattern alphabet concepts are not currently used.

The BLOSUM50 matrix is available in this package as a matrix:

```
> data(blosum50)
> blosum50[1:4, 1:4]
```

```
   A  R  N  D
A  5 -2 -1 -2
R -2  7 -1 -2
N -1 -1  7  2
D -2 -2  2  8
```

A canonical example from Durbin, Eddy et al:

```
> nwdemo <- needwunsQS("PAWHEAE", "HEAGAWGHEE", blosum50)
> nwdemo
```

```
     [,1]
[1,] "--P-AW-HEAE"
[2,] "HEAGAWGHE-E"
Score:  1
```

The score (final element of dynamic programming score matrix) can be accessed:

```
> alignScore(nwdemo)
```

```
[1] 1
```

4 Computing alignment consensus matrices

Here is some provisional code for computing a consensus matrix for a group of equal-length strings assumed to be aligned.

```
> consmat <- function(Lmers, freq = FALSE) {
+     lens <- nchar(Lmers)
+     if (!all(lens == lens[1]))
+         stop("must have equal number of bases in each element")
+     str <- as.character(Lmers)
+     N <- length(Lmers)
+     pure <- unlist(lapply(str, function(x) strsplit(x, "")[[1]]))
+     pos <- as.numeric(t(col(matrix(0, nr = N, nc = lens[1]))))
+     table(codes = pure, pos = pos)/ifelse(freq, N, 1)
+ }
```

To illustrate, the following application assumes the ORF data to be aligned for the first 10 positions (patently false):

```
> orf10 <- views(subject(orf), start(orf), start(orf) + 9)
> consmat(orf10)
```

```
     pos
codes 1 2 3 4 5 6 7 8 9 10
    A 2 2 2 0 4 3 3 3 2  1
    C 3 1 2 2 2 1 0 0 2  3
    G 1 1 1 2 1 0 3 3 1  2
    T 1 3 2 3 0 3 1 1 2  1
```

The information content as defined by Hertz and Stormo 1995 is computed as follows:

```
> infContent <- function(Lmers) {
+     zlog <- function(x) ifelse(x == 0, 0, log(x))
+     co <- consmat(Lmers, freq = TRUE)
+     lets <- rownames(co)
+     fr <- alphabetFrequency(Lmers)[lets]
+     sum(co * zlog(co/fr))
+ }
> infContent(orf10)

[1] -40.31777
```