

Vectorizing the `DNAString` function (work in progress)

Hervé Pagès

December 1, 2007

Contents

1	Introduction	1
2	DNAString vs BStringViews	1
3	The BStringViews generic function	2
4	Performance	2
5	Loading a FASTA file into a <i>BStringViews</i> object	3
6	Switching between DNA and RNA views	3

1 Introduction

This is a short tour on the `DNAString` function vectorization feature.

Feel free to add your own comments.

2 DNAString vs BStringViews

The `Biostrings2Classes` vignette presents a proposal for 2 new classes (*BString* and *BStringViews*) as a replacement for the *BioString* class currently defined in the *Biostrings* 1 (*Biostrings* v 1.4.x) package.

It also shows how to use the `DNAString` function to create a *DNAString* object (a *DNAString* object is just a particular case of a *BString* object):

```
> d <- DNAString("TTGAAAA-CTC-N")
```

However this function is NOT vectorized: it always returns a *DNAString* object (which can only represent a *single* string).

In *Biostrings* 1, the `DNAString` function IS vectorized. Its vectorized form does the following: (1) concatenates the elements of its `src` argument into a single big string, (2) stores the offsets of all these elements in the `offsets` slot.

This behaviour is not immediately obvious to the user, until he looks at the `offsets` slot.

It always returns a *BioString* object (with has as many values as the number of elements passed in the `src` argument).

3 The BStringViews generic function

The feature described in the previous section (provided by the vectorized form of the `DNASTring` function in *Biostrings* 1) is provided in *Biostrings* 2 via the `BStringViews` generic function:

```
> v <- BStringViews(c("TTGAAAA-C", "TC-N"), "DNASTring")
> v
```

Views on a 13-letter `DNASTring` subject

Subject: TTGAAAA-CTC-N

Views:

	start	end	width	
[1]	1	9	9	[TTGAAAA-C]
[2]	10	13	4	[TC-N]

4 Performance

The following example was provided by Wolfgang:

```
> library(hgu95av2probe)

> system.time(z <- BStringViews(hgu95av2probe$sequence, "DNASTring"))
```

user	system	elapsed
5.489	0.117	6.125

```
> z
```

Views on a 5045000-letter `DNASTring` subject

Subject: TGGCTCCTGCTGAGGTCCCCTTTCCGGCTGTGAA...AAGCCCTCGTGCTCCTTGTCAACAGCGCACCCA

Views:

	start	end	width	
[1]	1	25	25	[TGGCTCCTGCTGAGGTCCCCTTTCC]
[2]	26	50	25	[GGCTGTGAATTCCTGTACATATTC]
[3]	51	75	25	[GCTTCAATTCCATTATGTTTAAATG]
[4]	76	100	25	[GCCGTTTGACAGAGCATGCTCTGCG]
[5]	101	125	25	[TGACAGAGCATGCTCTGCGTTGTTG]
[6]	126	150	25	[CTCTGCGTTGTTGGTTTCACCAGCT]
[7]	151	175	25	[GGTTTCACCAGCTTCTGCCCTCACA]
[8]	176	200	25	[TTCTGCCCTCACATGCACAGGGATT]
[9]	201	225	25	[CCTCACATGCACAGGGATTTAACAA]
...
[201792]	5044776	5044800	25	[GAGTGCCAATTCGATGATGAGTCAG]

```
[201793] 5044801 5044825    25 [ACACTGACACTTGTGCTCCTTGTCA]
[201794] 5044826 5044850    25 [CAATTCGATGATGAGTCAGCAACTG]
[201795] 5044851 5044875    25 [GACTTCTGAGGAGATGGATAGCCT]
[201796] 5044876 5044900    25 [AGATGGATAGCCTTCTGTCAAAGCA]
[201797] 5044901 5044925    25 [ATAGCCTTCTGTCAAAGCATCATCT]
[201798] 5044926 5044950    25 [TTCTGTCAAAGCATCATCTCAACAA]
[201799] 5044951 5044975    25 [CAAAGCATCATCTCAACAAGCCCTC]
[201800] 5044976 5045000    25 [GTGCTCCTTGTCAACAGCGACCCCA]
```

With *Biostrings* 1, the call to `DNAString(hgu95av2probe$sequence)` takes about 20 minutes... (the implementation of the vectorization feature is quadratic in time, as reported by Wolfgang).

5 Loading a FASTA file into a *BStringViews* object

The `read.BStringViews` function can be used to load a FASTA file in a *BStringViews* object:

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

Views on a 26339-letter DNAString subject

Subject: ACTTGTAATATATCTTTTATTTTCCGAGAGGAA...TATACATAGGGCTAAGGAAGAAAAAAAATCAC
Views:

	start	end	width	
[1]	1	5573	5573	[ACTTGTAATATATCTTTTATTTTCC...ACGCTTATCGACCTTATTGTTGATAT]
[2]	5574	11398	5825	[TTCCAAGGCCGATGAATTCGACTCTT...CAGAGTAAATTTTTTCTATTCTCTT]
[3]	11399	14385	2987	[CTTCATGTCAGCCTGCACTTCTGGGT...CGATGGTACTCATGTAGCTGCCTCAT]
[4]	14386	18314	3929	[CACTCATATCGGGGCTTACTTCCC...ACGTGTCCCGAAACACGAAAAAGTAC]
[5]	18315	20962	2648	[AGAGAAAGAGTTTCACTTCTTGATTA...AAATATAATTTATGTGTGAACATAG]
[6]	20963	23559	2597	[GTGTCCGGGCCTCGCAGGCGTTCTAC...TTCAAGTTTGGCAGAATGTACTTTT]
[7]	23560	26339	2780	[CAAGATAATGTCAAAGTTAGTGGTCG...AGGGCTAAGGAAGAAAAAAAATCAC]

```
> desc(orf)
```

```
[1] "YAL001C TFC3 SGDID:S0000001, Chr I from 152168-146596, reverse complement, Verified ORF"
[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, Verified ORF"
[5] "YAL007C ERP2 SGDID:S0000005, Chr I from 139347-136700, reverse complement, Verified ORF"
[6] "YAL008W FUN14 SGDID:S0000006, Chr I from 135916-138512, Verified ORF"
[7] "YAL009W SP07 SGDID:S0000007, Chr I from 134856-137635, Verified ORF"
```

6 Switching between DNA and RNA views

The `BStringViews` function can also be used to switch between “DNA” and “RNA” views on the same string:

```
> orf2 <- BStringViews(orf, "RNAString")
```

These conversions are very fast because no string data needs to be copied:

```
> orf[[0]]@data
```

26339-byte XRaw object (starting at address 0x1008018)

```
> orf2[[0]]@data
```

26339-byte XRaw object (starting at address 0x1008018)