

Vectorizing the `DNAStr` function (work in progress)

Hervé Pagès

June 9, 2006

Contents

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

1 Introduction

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

Feel free to add your own comments.

2 `DNAStr` 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 `DNAStr` function to create a *DNAStr* object (a *DNAStr* object is just a particular case of a *BString* object):

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

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

In *Biostrings* 1, the `DNAStr` 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 `DNAStrng` function in *Biostrings* 1) is provided in *Biostrings* 2 via the `BStringViews` generic function:

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

Views on a 13-letter `DNAStrng` subject

Subject: TTGAAAA-CTC-N

Views:

	first	last	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, "DNAStrng"))
```

```
[1] 6.06 0.00 6.07 NA NA
```

```
> z
```

Views on a 4977100-letter `DNAStrng` subject

Subject: TCTCCTTTGCTGAGGCCTCCAGCTTAGGCCTCCA...GTGAAACCCAGCCTGGCCAACATGGTGAACCC

Views:

	first	last	width	
[1]	1	25	25	TCTCCTTTGCTGAGGCCTCCAGCTT
[2]	26	50	25	AGGCCTCCAGCTTCAGGCAGGCCAA
[3]	51	75	25	CCAGCTTCAGGCAGGCCAAGGCCTT
[4]	76	100	25	AGCTCAGGTGGCCCCAGTTCAATCT
[5]	101	125	25	AGTTCTGGAATGGAAGGGTTCTGGC
[6]	126	150	25	TAGGGACTCAGGGCCATGCCTGCCC
[7]	151	175	25	TTCCCTGAAGGAACATTCCTTAGTC
[8]	176	200	25	GAAGGAACATTCCTTAGTCTCAAGG
[9]	201	225	25	CTTAGTCTCAAGGGCTAGCATCCCT
...
[199076]	4976876	4976900	25	TTCAAGACCAGCCTGGCCAACATGG
[199077]	4976901	4976925	25	TCAAGACCAGCCTGGCCAACATGGT

```
[199078] 4976926 4976950    25 |CAAGACCAGCCTGGCCAACATGGTG|
[199079] 4976951 4976975    25 |AAGACCAGCCTGGCCAACATGGTGA|
[199080] 4976976 4977000    25 |AGACCAGCCTGGCCAACATGGTGAA|
[199081] 4977001 4977025    25 |GACCAGCCTGGCCAACATGGTGAAA|
[199082] 4977026 4977050    25 |ACCAGCCTGGCCAACATGGTGAAAC|
[199083] 4977051 4977075    25 |CCAGCCTGGCCAACATGGTGAAACC|
[199084] 4977076 4977100    25 |CAGCCTGGCCAACATGGTGAAACCC|
```

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 in a *BStringViews* object

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

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

Views on a 26339-letter `DNAString` subject

Subject: ACTTGTAATATATCTTTTATTTTCCGAGAGGAA...TATACATAGGGCTAAGGAAGAAAAAAAATCAC
Views:

	first	last	width	
[1]	1	5573	5573	ACTTGTAATATATCTTTTATTTTCC...ACGCTTATCGACCTTATTGTTGATAT
[2]	5574	11398	5825	TTCCAAGGCCGATGAATTCGACTCTT...CAGAGTAAATTTTTTCTATTCTCTT
[3]	11399	14385	2987	CTTCATGTCAGCCTGCACTTCTGGGT...CGATGGTACTCATGTAGCTGCCTCAT
[4]	14386	18314	3929	CACTCATATCGGGGTCTTACTTCCC...ACGTGTCCCGAAACACGAAAAAGTAC
[5]	18315	20962	2648	AGAGAAAGAGTTTCACTTCTTGATTA...AAAATATAATTTATGTGTGAACATAG
[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 buffer (starting at address 056CD4A0)

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

26339-byte buffer (starting at address 056CD4A0)