Biostrings Quick Overview

Hervé Pagès
Fred Hutchinson Cancer Research Center
Seattle, WA

March 13, 2024

Most but not all functions defined in the Biostrings package are summarized here.

Function	Description
`length`	Return the number of sequences in an object.
`names`	Return the names of the sequences in an object.
`[`	Extract sequences from an object.
`head`, `tail`	Extract the first or last sequences from an object.
`rev`	Reverse the order of the sequences in an object.
`c`	Combine in a single object the sequences from 2 or more objects.
`width`, `nchar`	Return the sizes (i.e. number of letters) of all the sequences in an object.
`==`, `!=`	Element-wise comparison of the sequences in 2 objects.
`match`, `%in%`	Analog to `match` and `%in%` on character vectors.
`duplicated`, `unique`	Analog to `duplicated` and `unique` on character vectors.
`sort`, `order`	Analog to `sort` and `order` on character vectors, except that the ordering of DNA or Amino Acid sequences doesn't depend on the locale.
`relist`, `split`, `extractList`	Analog to `relist` and `split` on character vectors, except that the result is a *DNAStringSetList* or *AAStringSetList* object. `extractList` is a generalization of `relist` and `split` that supports *arbitrary* groupings.

Table 1: **Low-level manipulation of *DNAStringSet* and *AAStringSet* objects.**

Function	Description
`alphabetFrequency` `letterFrequency`	Tabulate the letters (all the letters in the alphabet for `alphabetFrequency`, only the specified letters for `letterFrequency`) in a sequence or set of sequences.
`uniqueLetters`	Extract the unique letters from a sequence or set of sequences.
`letterFrequencyInSlidingView`	Specialized version of `letterFrequency` that tallies the requested letter frequencies for a fixed-width view that is conceptually slid along the input sequence.
`consensusMatrix`	Computes the consensus matrix of a set of sequences.

`dinucleotideFrequency`
`trinucleotideFrequency`
`oligonucleotideFrequency`
	Fast 2-mer, 3-mer, and k-mer counting for DNA or RNA.

`nucleotideFrequencyAt`	Tallies the short sequences formed by extracting the nucleotides found at a set of fixed positions from each sequence of a set of DNA or RNA sequences.

Table 2: **Counting / tabulating.**

Function	Description

`reverse`
`complement`
`reverseComplement`
	Compute the reverse, complement, or reverse-complement, of a set of DNA sequences.

`translate`	Translate a set of DNA sequences into a set of Amino Acid sequences.
`chartr` `replaceAmbiguities`	Replace letters in a sequence or set of sequences.
`subseq`, `subseq<-` `extractAt`, `replaceAt`	Extract/replace arbitrary substrings from/in a string or set of strings.
`replaceLetterAt`	Replace the letters specified by a set of positions by new letters.
`padAndClip`, `stackStrings`	Pad and clip strings.
`strsplit`, `unstrsplit`	`strsplit` splits the sequences in a set of sequences according to a pattern. `unstrsplit` is the reverse operation i.e. a fast implementation of `sapply(x, paste0, collapse=sep)` for collapsing the list elements of a *DNAStringSetList* or *AAStringSetList* object.

Table 3: **Sequence transformation and editing.**

Function	Description
`matchPattern` `countPattern`	Find/count all the occurrences of a given pattern (typically short) in a reference sequence (typically long). Support mismatches and indels.
`vmatchPattern` `vcountPattern`	Find/count all the occurrences of a given pattern (typically short) in a set of reference sequences. Support mismatches and indels.

`matchPDict`
`countPDict`
`whichPDict`
	Find/count all the occurrences of a set of patterns in a reference sequence. (`whichPDict` only identifies which patterns in the set have at least one match.) Support a small number of mismatches.

`vmatchPDict`
`vcountPDict`
`vwhichPDict`
	[Note: `vmatchPDict` not implemented yet.] Find/count all the occurrences of a set of patterns in a set of reference sequences. (`whichPDict` only identifies for each reference sequence which patterns in the set have at least one match.) Support a small number of mismatches.

`pairwiseAlignment`	Solve (Needleman-Wunsch) global alignment, (Smith-Waterman) local alignment, and (ends-free) overlap alignment problems.
`matchPWM` `countPWM`	Find/count all the occurrences of a Position Weight Matrix in a reference sequence.
`trimLRPatterns`	Trim left and/or right flanking patterns from sequences.
`matchLRPatterns`	Find all paired matches in a reference sequence i.e. matches specified by a left and a right pattern, and a maximum distance between them.
`matchProbePair`	Find all the amplicons that match a pair of probes in a reference sequence.
`findPalindromes`	Find palindromic regions in a sequence.

Table 4: **String matching / alignments.**

Function	Description

`readBStringSet`
`readDNAStringSet`
`readRNAStringSet`
`readAAStringSet`
	Read ordinary/DNA/RNA/Amino Acid sequences from files (FASTA or FASTQ format).

`writeXStringSet`	Write sequences to a file (FASTA or FASTQ format).
`writePairwiseAlignments`	Write pairwise alignments (as produced by `pairwiseAlignment`) to a file ("pair" format).

`readDNAMultipleAlignment`
`readRNAMultipleAlignment`
`readAAMultipleAlignment`
	Read multiple alignments from a file (FASTA, "stockholm", or "clustal" format).

`write.phylip`	Write multiple alignments to a file (Phylip format).

Table 5: **I/O functions.**

Function	Description
`stringDist`	Computes the matrix of Levenshtein edit distances, or Hamming distances, or pairwise alignment scores, for a set of strings.

Table 6: **Miscellaneous.**