

# Biostrings Quick Overview

Hervé Pagès  
Fred Hutchinson Cancer Research Center  
Seattle, WA

August 25, 2022

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

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

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

| Function   | Description   |
|--|---|
| <code>alphabetFrequency</code><br><code>letterFrequency</code>   | Tabulate the letters (all the letters in the alphabet for <code>alphabetFrequency</code> , only the specified letters for <code>letterFrequency</code> ) in a sequence or set of sequences. |
| <code>uniqueLetters</code>   | Extract the unique letters from a sequence or set of sequences.   |
| <code>letterFrequencyInSlidingView</code>  | Specialized version of <code>letterFrequency</code> that tallies the requested letter frequencies for a fixed-width view that is conceptually slid along the input sequence.                |
| <code>consensusMatrix</code>   | Computes the consensus matrix of a set of sequences.  |
| <code>dinucleotideFrequency</code><br><code>trinucleotideFrequency</code><br><code>oligonucleotideFrequency</code> | Fast 2-mer, 3-mer, and k-mer counting for DNA or RNA.   |
| <code>nucleotideFrequencyAt</code>   | 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<br>complement<br>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<br>replaceAmbiguities               | Replace letters in a sequence or set of sequences.   |
| subseq, subseq<-<br>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                       | <code>strsplit</code> splits the sequences in a set of sequences according to a pattern. <code>unstrsplit</code> is the reverse operation i.e. a fast implementation of <code>sapply(x, paste0, collapse=sep)</code> for collapsing the list elements of a <i>DNASetList</i> or <i>AASetList</i> object. |

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

| Function                                  | Description  |
|---|--|
| matchPattern<br>countPattern              | Find/count all the occurrences of a given pattern (typically short) in a reference sequence (typically long). Support mismatches and indels.   |
| vmatchPattern<br>vcountPattern            | Find/count all the occurrences of a given pattern (typically short) in a set of reference sequences. Support mismatches and indels.  |
| matchPDict<br>countPDict<br>whichPDict    | Find/count all the occurrences of a set of patterns in a reference sequence. ( <code>whichPDict</code> only identifies which patterns in the set have at least one match.) Support a small number of mismatches.   |
| vmatchPDict<br>vcountPDict<br>vwhichPDict | [Note: <code>vmatchPDict</code> not implemented yet.] Find/count all the occurrences of a set of patterns in a set of reference sequences. ( <code>whichPDict</code> 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<br>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.**

| <b>Function</b>   | <b>Description</b>  |
|---|---|
| readBStringSet<br>readDNAStringSet<br>readRNAStringSet<br>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<br>readRNAMultipleAlignment<br>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.**

| <b>Function</b> | <b>Description</b>   |
|-----------------|--|
| stringDist      | Computes the matrix of Levenshtein edit distances, or Hamming distances, or pairwise alignment scores, for a set of strings. |

Table 6: **Miscellaneous.**