

# Biostrings Quick Overview

Hervé Pagès  
Fred Hutchinson Cancer Research Center  
Seattle, WA

April 3, 2013

Please note that *most* but *not all* the functionalities provided by the Biostrings package are listed in this document.

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>	Put 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>split, relist</code>	Analog to <code>split</code> and <code>relist</code> on character vectors, except that the result is a <i>DNASetList</i> or <i>AASetList</i> object.

Table 1: Low-level manipulation of *DNASetList* or *AASetList* objects.

Function	Description
<code>subseq, subseq&lt;-</code>	Extract or replace subsequences in a set of sequences.
<code>reverse</code> <code>complement</code> <code>reverseComplement</code>	Compute the reverse, complement, or reverse-complement, of a set of DNA sequences.
<code>translate</code>	Translate a set of DNA sequences into a set of Amino Acid sequences.
<code>chartr</code>	Translate the letters in a set of sequences.
<code>replaceLetterAt</code>	Replace the letters specified by a set of positions by new letters.

Table 2: Basic transformations of sequences.

Function	Description
<code>alphabetFrequency</code> <code>letterFrequency</code>	Tabulate the letters (all the letters in the alphabet for <code>alphabetFrequency</code> , only the specified letters for <code>letterFrequency</code> ) of 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> <code>trinucleotideFrequency</code> <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 3: Counting / tabulating.

Function	Description
<code>matchPattern</code> <code>countPattern</code>	Find/count all the occurrences of a given pattern (typically short) in a reference sequence (typically long). Support mismatches and indels.
<code>vmatchPattern</code> <code>vcountPattern</code>	Find/count all the occurrences of a given pattern (typically short) in a set of reference sequences. Support mismatches and indels.
<code>matchPDict</code> <code>countPDict</code> <code>whichPDict</code>	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.
<code>vmatchPDict</code> <code>vcountPDict</code> <code>vwhichPDict</code>	[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.
<code>pairwiseAlignment</code>	Solve (Needleman-Wunsch) global alignment, (Smith-Waterman) local alignment, and (ends-free) overlap alignment problems.
<code>matchPWM</code> <code>countPWM</code>	Find/count all the occurrences of a Position Weight Matrix in a reference sequence.
<code>trimLRPatterns</code>	Trim left and/or right flanking patterns from sequences.
<code>matchLRPatterns</code>	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.
<code>matchProbePair</code>	Find all the amplicons that match a pair of probes in a reference sequence.
<code>findPalindromes</code> <code>findComplementedPalindromes</code>	Find palindromic or complemented palindromic regions in a sequence.

Table 4: String matching / alignments.

Function	Description
<code>readBStringSet</code> <code>readDNAStrngSet</code> <code>readRNAStrngSet</code> <code>readAAStringSet</code>	Read ordinary/DNA/RNA/Amino Acid sequences from files (FASTA or FASTQ format).
<code>writeXStringSet</code>	Write sequences to a file (FASTA or FASTQ format).
<code>writePairwiseAlignments</code>	Write pairwise alignments (as produced by <code>pairwiseAlignment</code> ) to a file (“pair” format).
<code>readDNAMultipleAlignment</code> <code>readRNAMultipleAlignment</code> <code>readAAMultipleAlignment</code>	Read multiple alignments from a file (FASTA, “stockholm”, or “clustal” format).
<code>write.phylip</code>	Write multiple alignments to a file (Phylip format).

Table 5: I/O functions.

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

Table 6: Miscellaneous.