

# Benchmarking

*Michiel Coghe*

*4 juni 2019*

## Benchmarking of the different functions in the bendDNA package

In this R Markdown file all the different functions included in the bendDNA package are benchmarked. This will be done by looking at every function individually, testing and timing the functions for different parameter values.

The testing is done  $x$  times (depending on which function) and summary statistics are given in a summary table. These are then plotted using violin plots comparing every different parameter value. The CRAN microbenchmark package is used for making both the summary table and plot.

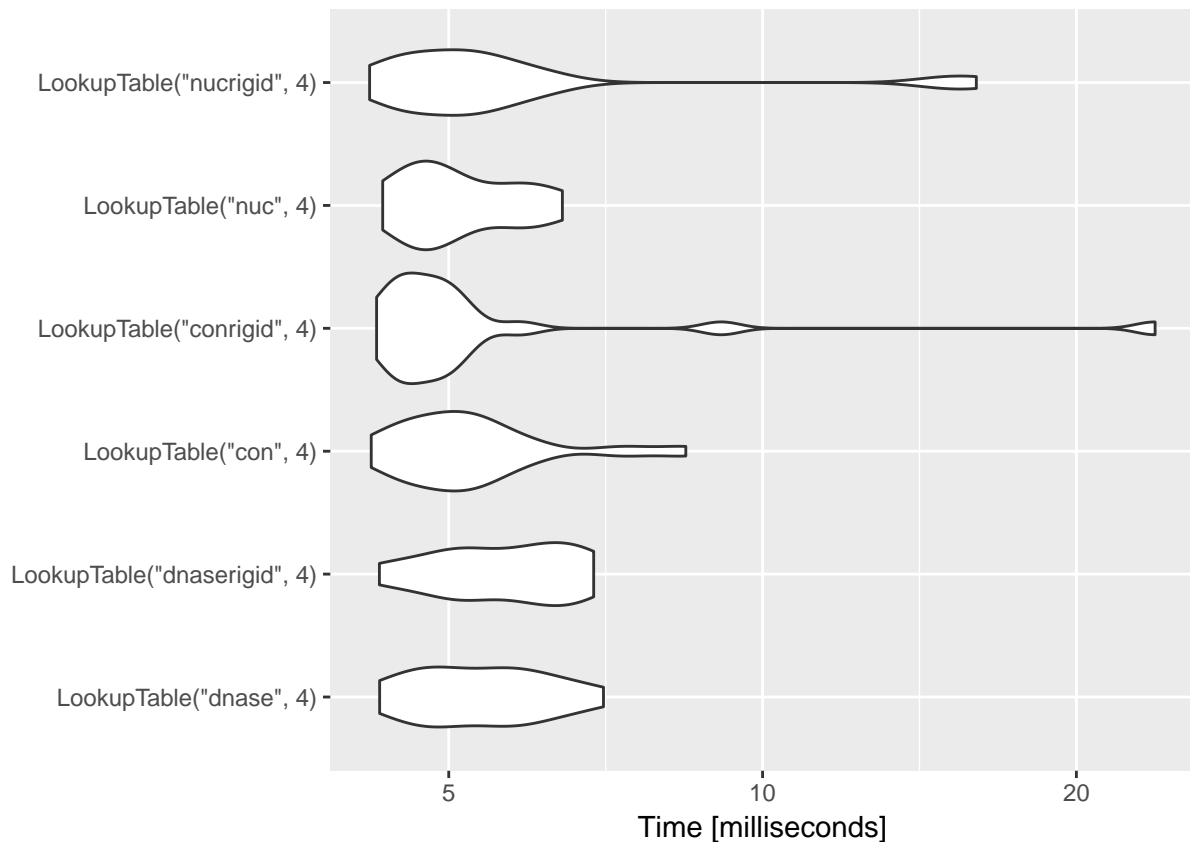
### LookupTable

Function for producing all possible k-mers and calculating their average bendability coefficients.

Usage: `LookupTable(scale, k, sequence.out = F)`

Parameters that can be used include: - Scale - Number of consecutive trinucleotides - Whether to output the sequences

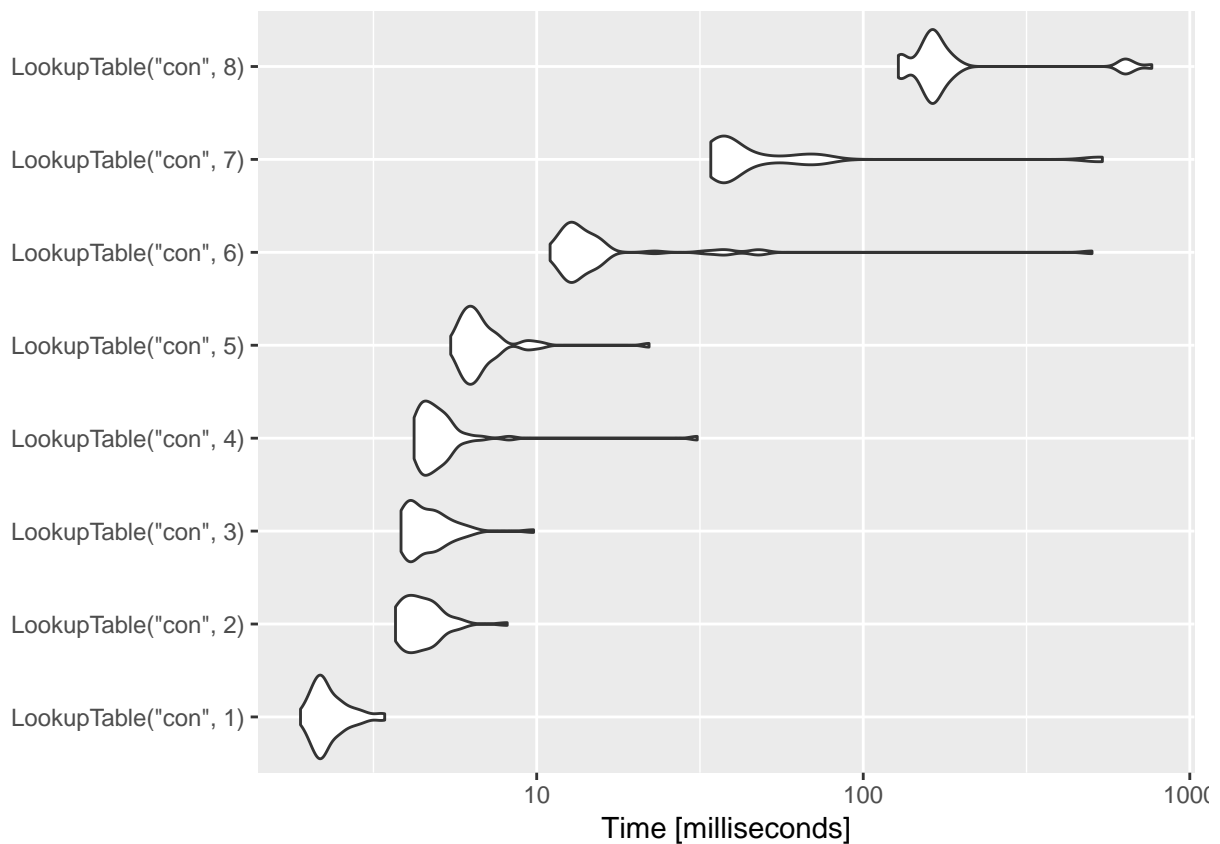
### Different lookup tables



FALSE Unit: milliseconds

FALSE		expr	min	lq	mean	median	uq
FALSE		LookupTable("dnase", 4)	4.292085	4.813949	5.418130	5.378200	5.969988
FALSE		LookupTable("dnaserigid", 4)	4.289925	5.080146	5.699121	5.821772	6.513721
FALSE		LookupTable("con", 4)	4.213252	4.632796	5.320857	5.071777	5.484840
FALSE		LookupTable("conrigid", 4)	4.263467	4.506177	5.944880	4.793702	5.082575
FALSE		LookupTable("nuc", 4)	4.321243	4.669782	5.143078	4.904391	5.718910
FALSE		LookupTable("nucrigid", 4)	4.198134	4.614707	6.130975	5.188406	5.690563
FALSE		max neval cld					
FALSE	7.037204	20	a				
FALSE	6.886558	20	a				
FALSE	8.441082	20	a				
FALSE	23.798957	20	a				
FALSE	6.426519	20	a				
FALSE	16.034437	20	a				

### Number of consecutive trinucleotides



FALSE Unit: milliseconds

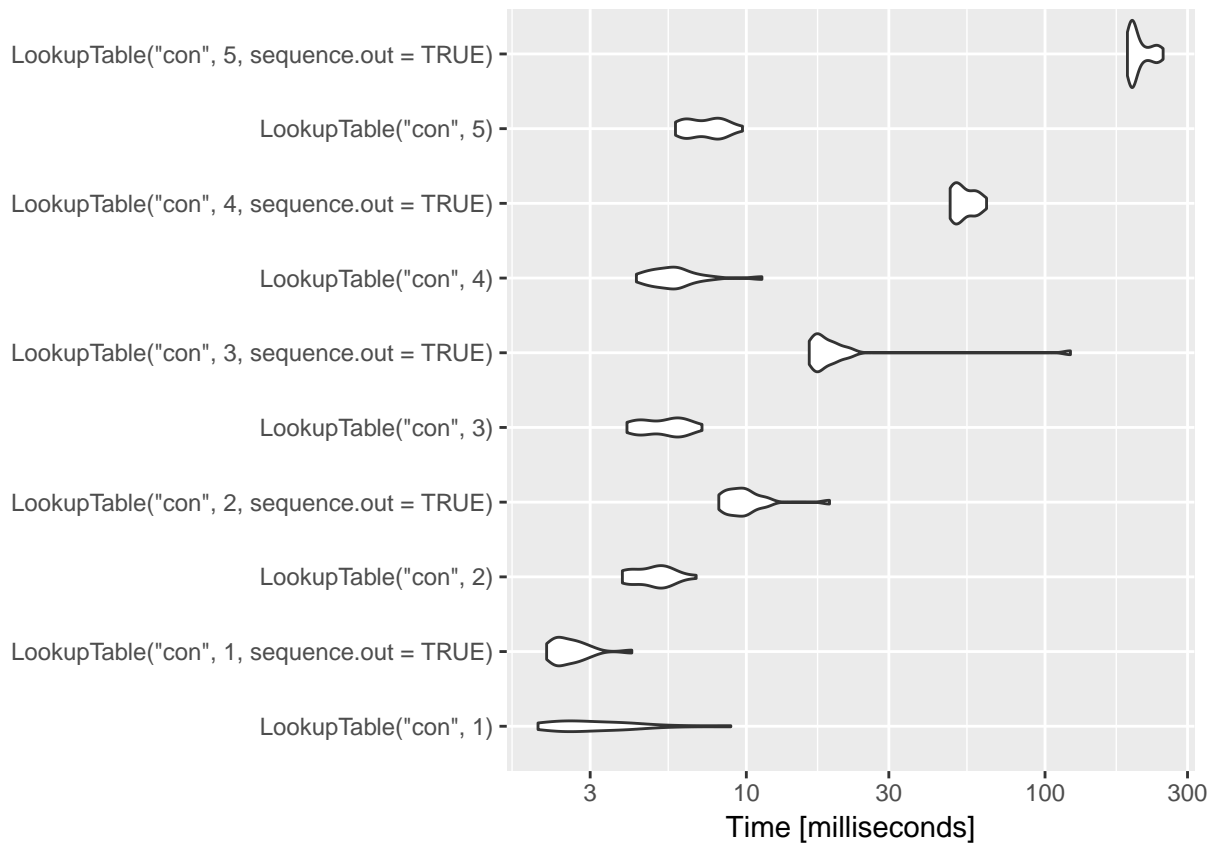
FALSE		expr	min	lq	mean	median
FALSE		LookupTable("con", 1)	1.890916	2.123095	2.332046	2.221096
FALSE		LookupTable("con", 2)	3.694897	3.949755	4.502591	4.319353
FALSE		LookupTable("con", 3)	3.841225	4.098242	4.706693	4.525615
FALSE		LookupTable("con", 4)	4.211632	4.478369	5.471018	4.823398
FALSE		LookupTable("con", 5)	5.452984	5.988617	7.013674	6.424898
FALSE		LookupTable("con", 6)	10.989119	12.490727	26.079987	13.374630
FALSE		LookupTable("con", 7)	34.123934	36.389575	73.677770	39.034803

```

FALSE LookupTable("con", 8) 128.094605 154.314169 220.778861 163.652921
FALSE      uq      max neval cld
FALSE    2.481623    3.423841    50 a
FALSE    4.868755    8.123589    50 a
FALSE    5.003743    9.783944    50 a
FALSE    5.210005   31.020284    50 a
FALSE    6.995089   22.087307    50 a
FALSE   15.414031  501.953010    50 a
FALSE   51.730712  539.972701    50 b
FALSE  178.923595  765.127047    50 c

```

## Sequence Out



```

## Unit: milliseconds
##
##          expr      min      lq
## LookupTable("con", 1)  2.007005  2.349066
## LookupTable("con", 1, sequence.out = TRUE)  2.144693  2.297500
## LookupTable("con", 2)  3.846084  4.223511
## LookupTable("con", 2, sequence.out = TRUE)  8.087953  8.774233
## LookupTable("con", 3)  3.984852  4.469460
## LookupTable("con", 3, sequence.out = TRUE) 16.241238 17.145660
## LookupTable("con", 4)  4.284526  5.013462
## LookupTable("con", 4, sequence.out = TRUE) 48.124907 49.737476
## LookupTable("con", 5)  5.791534  6.308539
## LookupTable("con", 5, sequence.out = TRUE) 188.739394 193.942380
##      mean      median      uq      max neval  cld

```

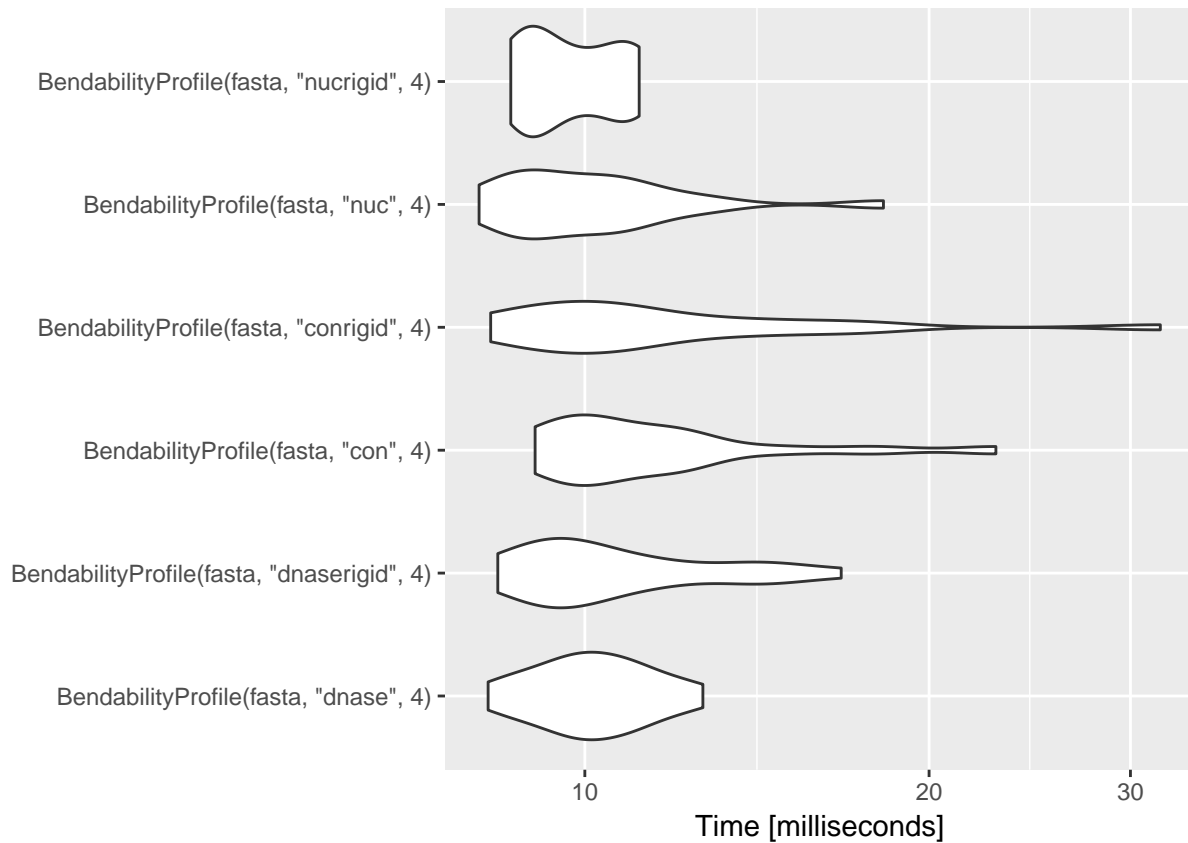
##	3.396709	2.801006	3.897650	8.874664	20	a
##	2.605273	2.510782	2.764558	4.136578	20	a
##	4.911897	4.994023	5.302337	6.793147	20	a
##	10.088991	9.660565	10.384911	18.991218	20	a
##	5.316673	5.504009	6.019664	7.106319	20	a
##	23.602279	17.928591	19.759840	121.661608	20	b
##	5.876900	5.680843	6.115775	11.278533	20	a
##	53.808692	52.072232	57.876723	63.721441	20	c
##	7.335178	7.389794	8.129800	9.707811	20	a
##	206.505108	197.811140	215.563436	248.547798	20	d

## BendabilityProfile

Splits sequence into k-mers with two-nucleotide overlap and calculates average bendability per k-mer according to the chosen scale.

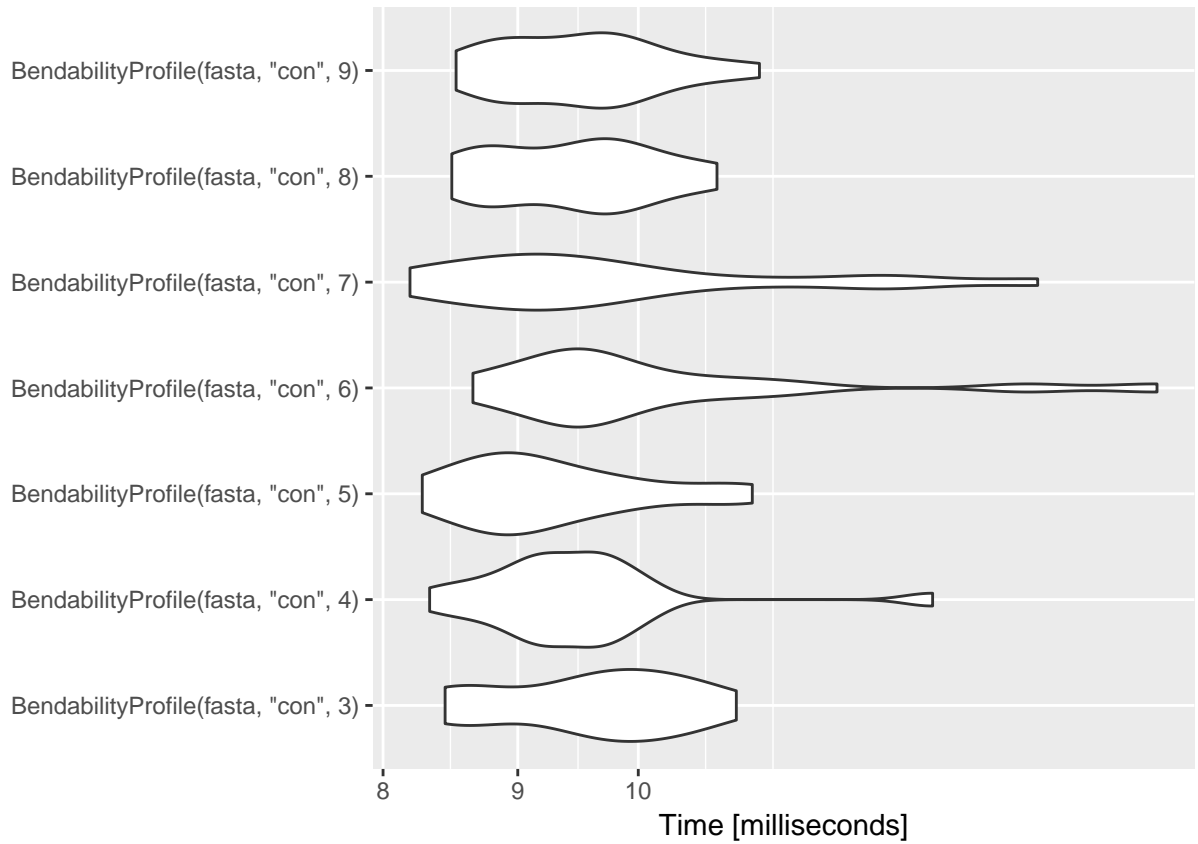
Parameters that can be used include: - Scale - Number of consecutive trinucleotides

### Scale



```
## Unit: milliseconds
##
##          expr      min      lq      mean
##  BendabilityProfile(fasta, "dnase", 4) 8.230500 9.382759 10.217823
##  BendabilityProfile(fasta, "dnaserigid", 4) 8.392486 9.207814 10.846193
##  BendabilityProfile(fasta, "con", 4) 9.047449 9.775305 11.755015
##  BendabilityProfile(fasta, "conrigid", 4) 8.273157 9.411107 12.277932
##  BendabilityProfile(fasta, "nuc", 4) 8.082014 8.837677 10.325841
##  BendabilityProfile(fasta, "nucrigid", 4) 8.614946 8.979685 9.752087
##    median      uq      max neval cld
## 10.125194 10.99533 12.68349    20   a
##  9.868717 11.81093 16.75743    20   a
## 10.509641 12.21346 22.88104    20   a
## 10.634909 13.31523 31.86099    20   a
##  9.872496 10.94619 18.24500    20   a
##  9.482381 10.67973 11.15650    20   a
```

## Number of consecutive trinucleotides



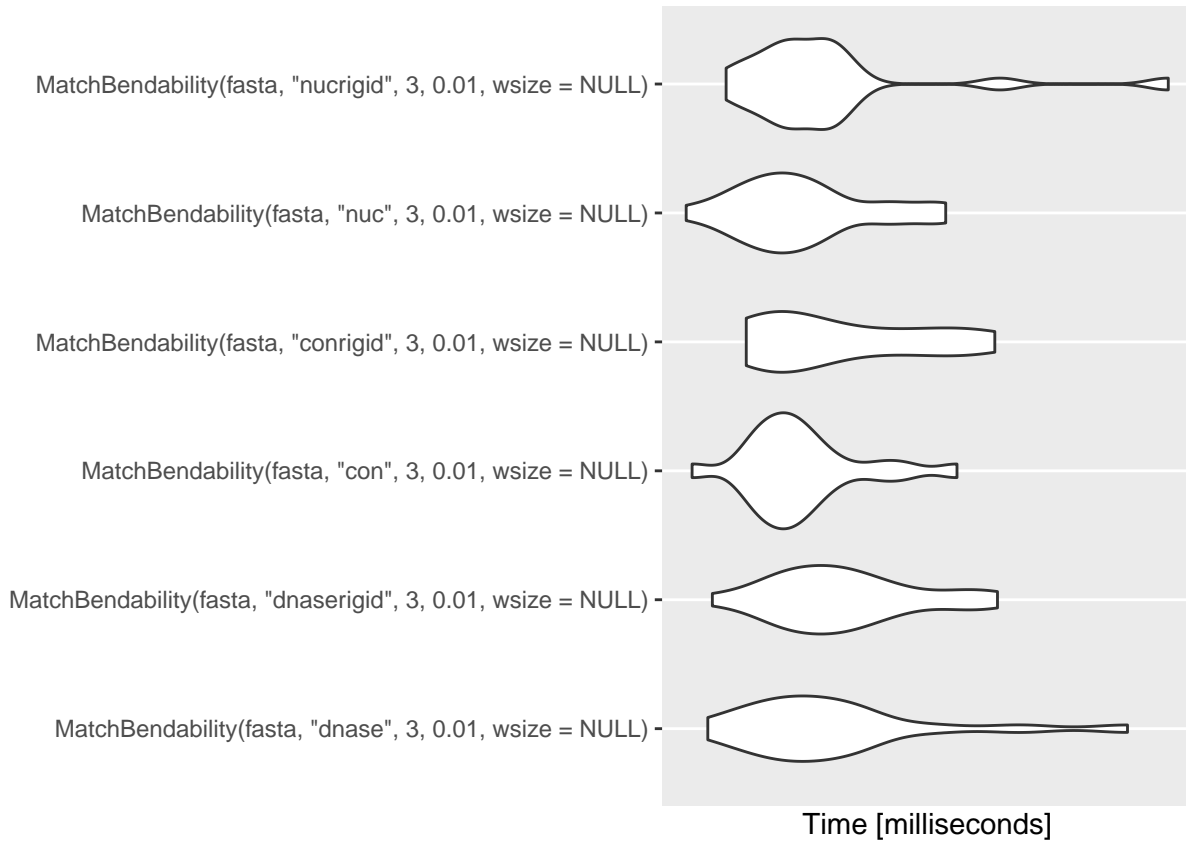
```
## Unit: milliseconds
##               expr      min      lq      mean  median
## BendabilityProfile(fasta, "con", 3) 8.445941 8.976174  9.633783 9.727249
## BendabilityProfile(fasta, "con", 4) 8.332551 9.058518  9.498147 9.349282
## BendabilityProfile(fasta, "con", 5) 8.279096 8.757494  9.283732 9.091185
## BendabilityProfile(fasta, "con", 6) 8.653824 9.328224 10.218498 9.596580
## BendabilityProfile(fasta, "con", 7) 8.190544 8.899501  9.837265 9.339294
## BendabilityProfile(fasta, "con", 8) 8.495617 8.799340  9.478196 9.580112
## BendabilityProfile(fasta, "con", 9) 8.527475 8.908410  9.526657 9.589022
##      uq      max neval cld
## 10.201328 10.89517   20   a
##  9.715640 12.93511   20   a
##  9.694313 11.04851   20   a
## 10.447006 15.73908   20   a
## 10.216716 14.17970   20   a
##  9.862777 10.71266   20   a
##  9.908943 11.11763   20   a
```

## MatchBendability

Calculates bendability profile of query sequence and finds all other sequences which have similar (or same) profile.

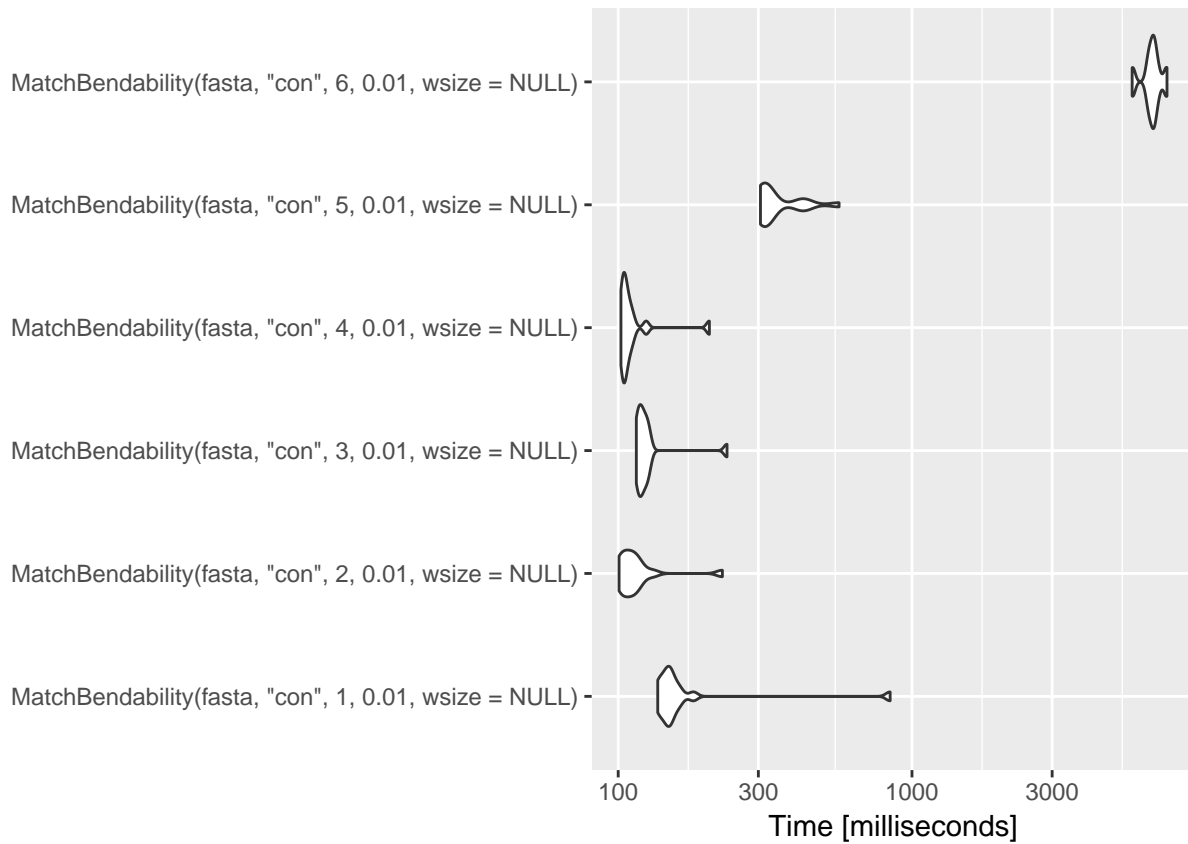
Parameters that can be used include: - Scale - Number of consecutive trinucleotides - Tolerance - wsize - output.list - random.out

### Scale



```
## Unit: milliseconds
##
##                               expr      min
##      MatchBendability(fasta, "dnase", 3, 0.01, wsize = NULL) 114.9711
##      MatchBendability(fasta, "dnaserigid", 3, 0.01, wsize = NULL) 115.3928
##      MatchBendability(fasta, "con", 3, 0.01, wsize = NULL) 113.5326
##      MatchBendability(fasta, "conrigid", 3, 0.01, wsize = NULL) 118.5736
##      MatchBendability(fasta, "nuc", 3, 0.01, wsize = NULL) 112.9964
##      MatchBendability(fasta, "nucrigid", 3, 0.01, wsize = NULL) 116.6724
##      lq      mean  median    uq      max neval cld
## 120.8552 127.4844 124.6192 129.9718 160.9756    20  a
## 122.8568 128.3426 127.5922 132.0101 145.0351    20  a
## 120.5018 124.0066 123.1019 125.4445 140.4044    20  a
## 120.7553 128.1322 124.0185 135.6507 144.7257    20  a
## 119.7089 124.2088 122.3860 126.7034 139.1410    20  a
## 121.4329 126.7266 123.7366 127.0838 166.3265    20  a
```

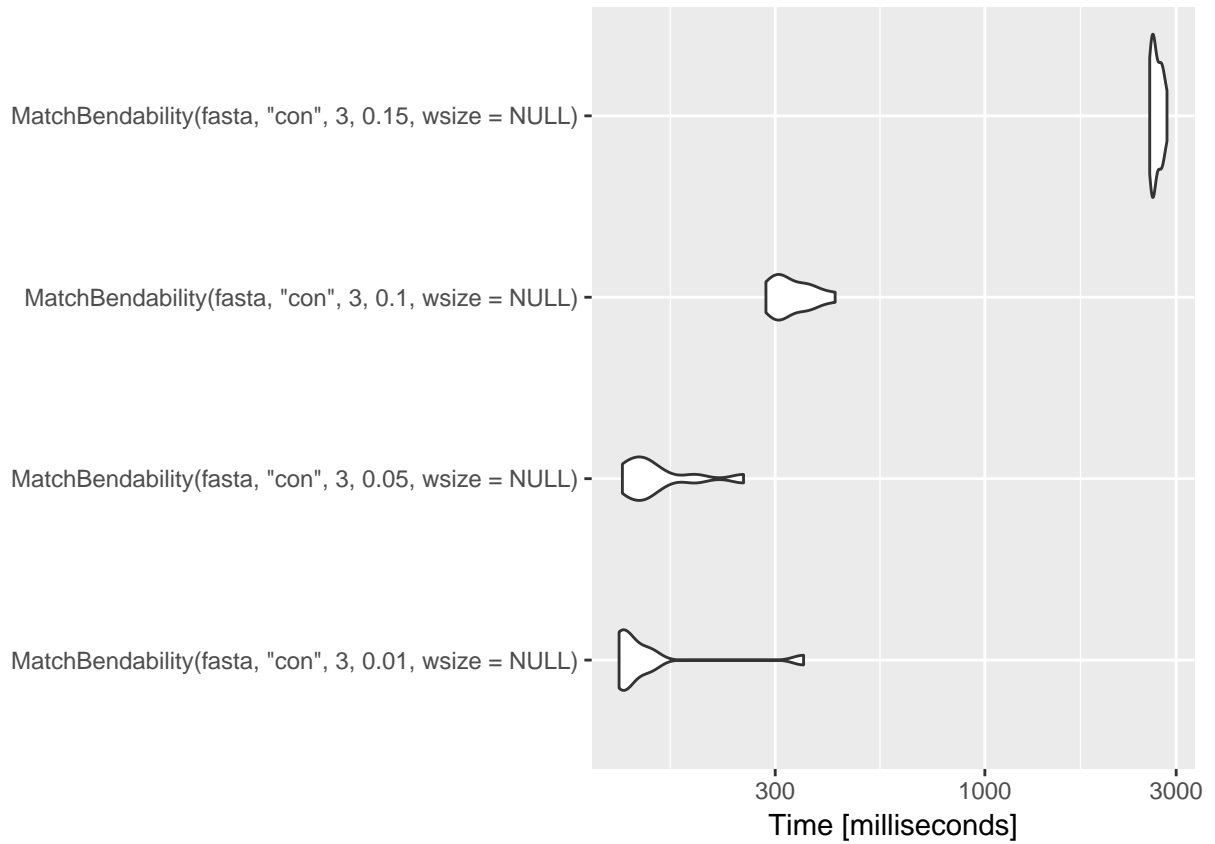
## Number of consecutive sequences



```
## Unit: milliseconds
##
## expr min lq
## MatchBendability(fasta, "con", 1, 0.01, wsize = NULL) 136.3505 143.5043
## MatchBendability(fasta, "con", 2, 0.01, wsize = NULL) 100.7104 103.9174
## MatchBendability(fasta, "con", 3, 0.01, wsize = NULL) 115.2734 117.7915
## MatchBendability(fasta, "con", 4, 0.01, wsize = NULL) 102.1936 103.8869
## MatchBendability(fasta, "con", 5, 0.01, wsize = NULL) 304.9831 311.8894
## MatchBendability(fasta, "con", 6, 0.01, wsize = NULL) 5615.8026 6407.8541
## mean median uq max neval cld
## 196.3076 148.7365 156.3571 842.6724 15 ab
## 118.3250 110.4430 116.8007 226.3120 15 a
## 128.4085 120.1011 125.1027 234.6483 15 ab
## 113.8949 105.8901 110.0024 204.3278 15 a
## 354.9495 316.0867 376.2482 564.7463 15 b
## 6572.8606 6614.7778 6748.9592 7379.3902 15 c
```

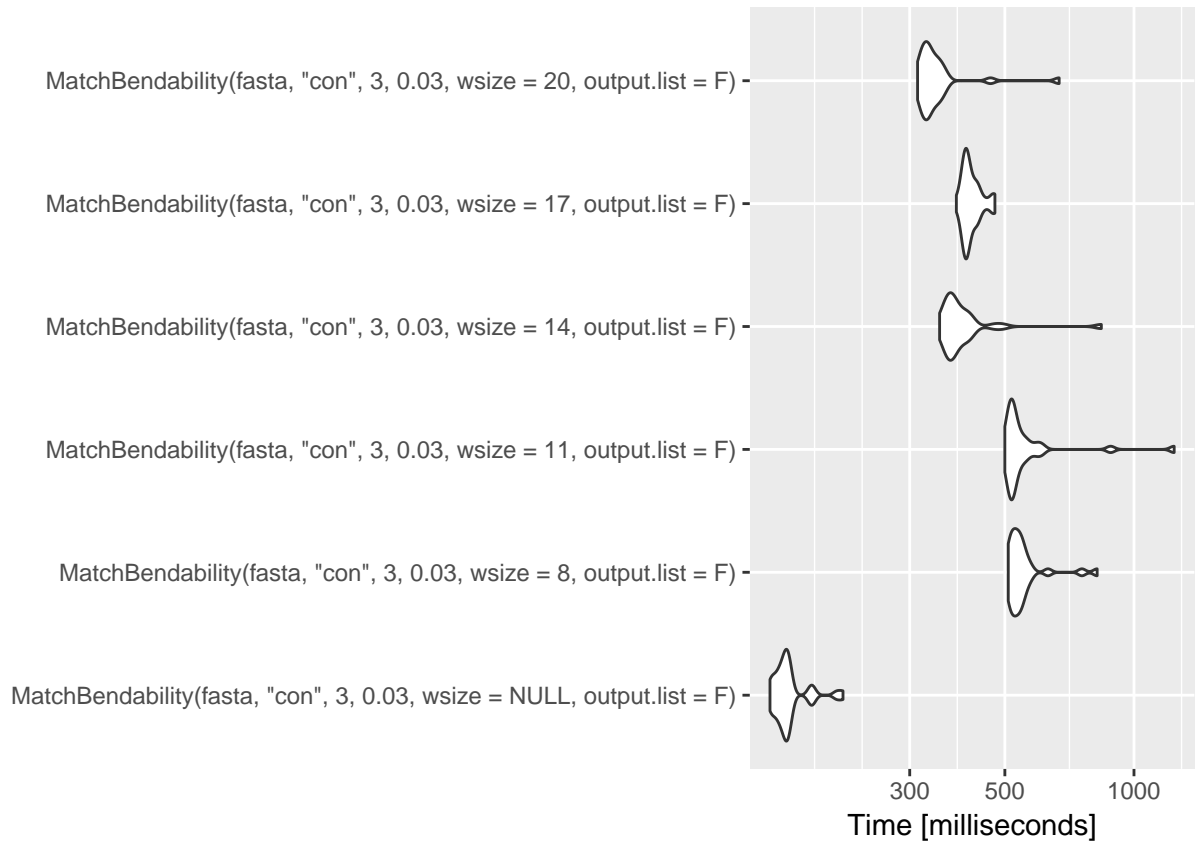


## Tolerance



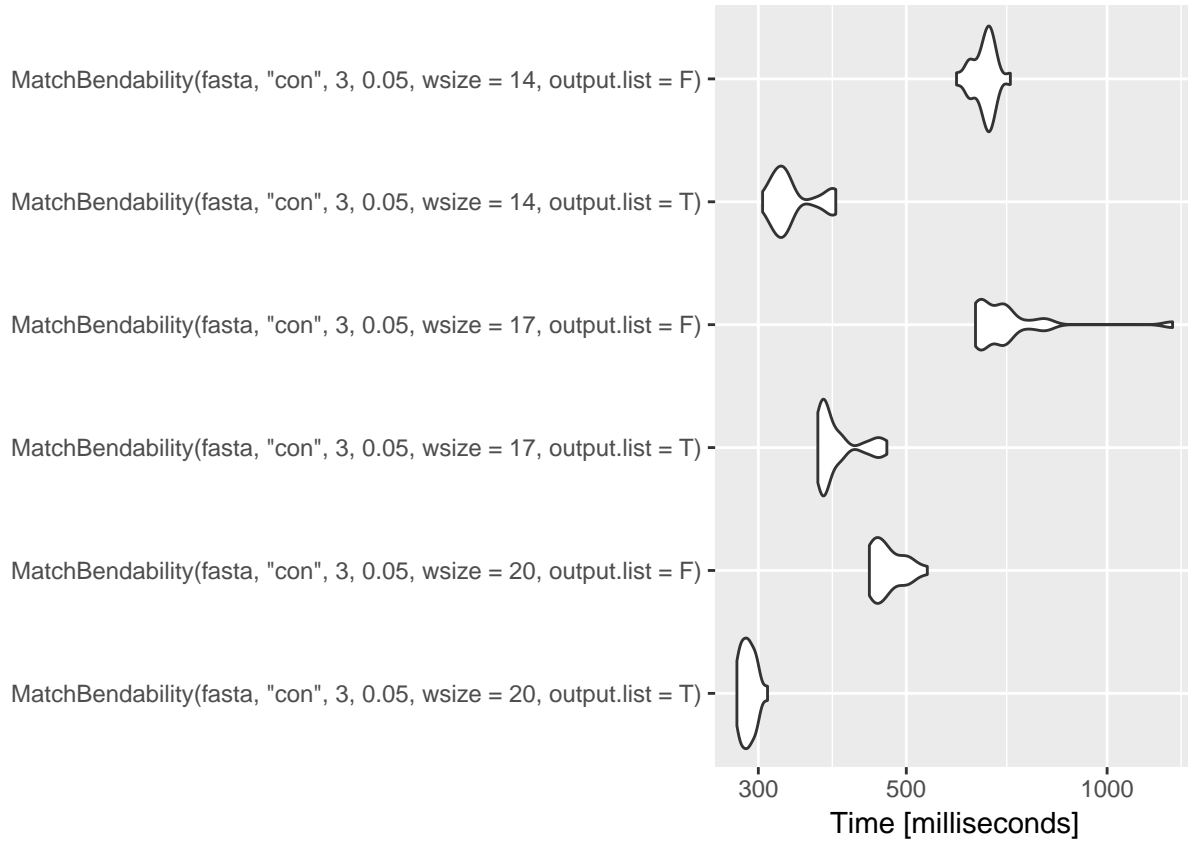
```
## Unit: milliseconds
##
##      expr      min      lq
## MatchBendability(fasta, "con", 3, 0.01, wsize = NULL) 122.3981 124.3398
## MatchBendability(fasta, "con", 3, 0.05, wsize = NULL) 124.7285 130.3154
## MatchBendability(fasta, "con", 3, 0.1, wsize = NULL) 284.4374 302.4286
## MatchBendability(fasta, "con", 3, 0.15, wsize = NULL) 2577.4519 2613.6536
##      mean    median      uq      max neval cld
## 152.8669 126.7723 145.8477 353.0816   10 a
## 154.7367 141.4765 157.9086 250.1115   10 a
## 332.3620 311.5900 359.3769 423.2662   10 b
## 2683.8485 2650.7823 2752.3150 2846.1032   10 c
```

## Window size



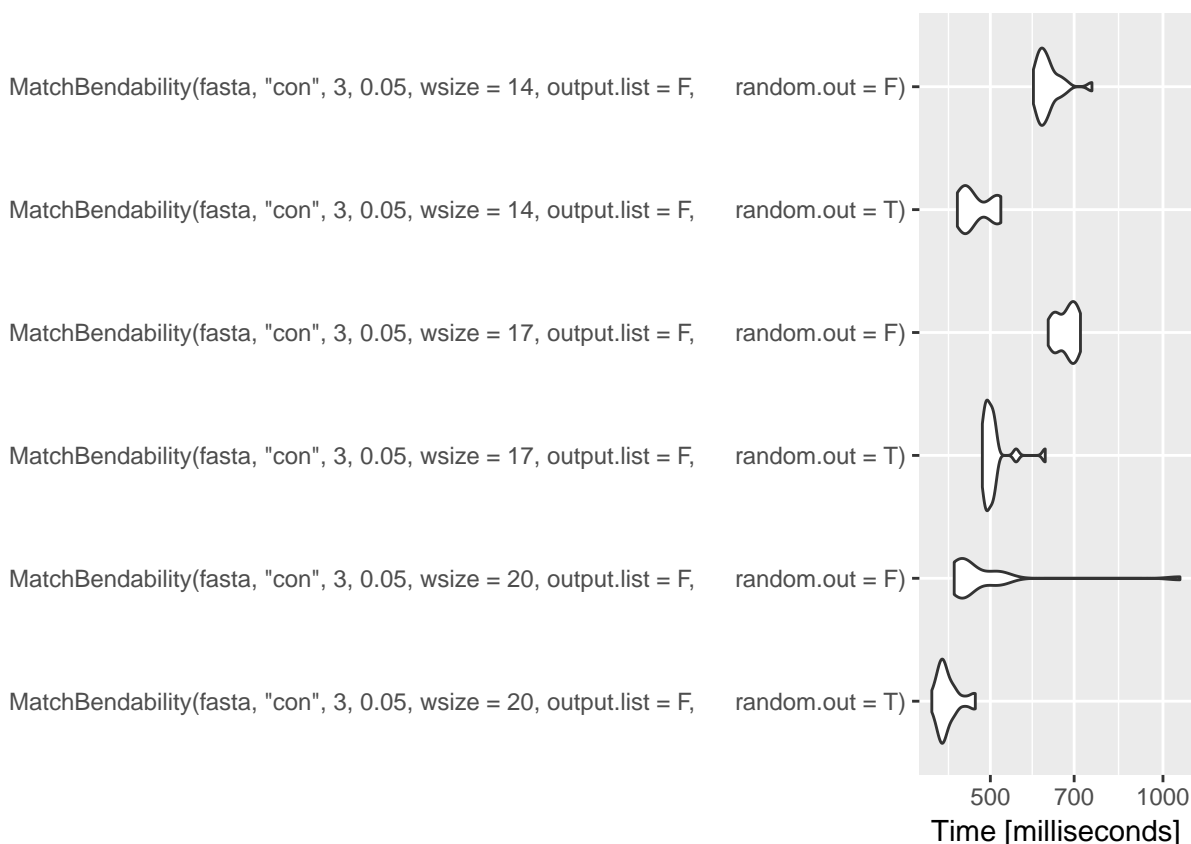
```
## Unit: milliseconds
##
##                                     expr
## MatchBendability(fasta, "con", 3, 0.03, wsize = NULL, output.list = F)
##   MatchBendability(fasta, "con", 3, 0.03, wsize = 8, output.list = F)
##   MatchBendability(fasta, "con", 3, 0.03, wsize = 11, output.list = F)
##   MatchBendability(fasta, "con", 3, 0.03, wsize = 14, output.list = F)
##   MatchBendability(fasta, "con", 3, 0.03, wsize = 17, output.list = F)
##   MatchBendability(fasta, "con", 3, 0.03, wsize = 20, output.list = F)
##      min      lq      mean  median      uq      max neval  cld
## 141.6831 148.7219 157.7439 154.5134 157.7337 209.6577   30  a
## 509.1230 520.3589 554.6345 535.9301 554.0109 820.6655   30  d
## 499.9293 516.5436 568.1056 522.1224 552.6513 1240.8940   30  d
## 352.2176 369.3050 404.1424 378.4839 409.7113 839.2960   30 bc
## 385.3670 403.8144 418.1357 409.5882 429.8699 473.9894   30  c
## 312.9512 325.0812 350.2192 333.0960 349.9444 669.1618   30  b
```

## Output List



```
## Unit: milliseconds
##
##                                     expr
## MatchBendability(fasta, "con", 3, 0.05, wsize = 20, output.list = T)
## MatchBendability(fasta, "con", 3, 0.05, wsize = 20, output.list = F)
## MatchBendability(fasta, "con", 3, 0.05, wsize = 17, output.list = T)
## MatchBendability(fasta, "con", 3, 0.05, wsize = 17, output.list = F)
## MatchBendability(fasta, "con", 3, 0.05, wsize = 14, output.list = T)
## MatchBendability(fasta, "con", 3, 0.05, wsize = 14, output.list = F)
##      min      lq      mean  median      uq      max neval  cld
## 278.6118 282.7778 289.7833 288.9573 296.2394 309.6677   20  a
## 440.1111 447.1011 469.4727 462.0227 492.1450 537.6050   20  c
## 368.4249 375.3738 394.0707 376.8281 399.1125 467.5229   20  b
## 634.9120 647.0359 717.2018 696.2763 706.7447 1253.6698   20  e
## 304.4367 318.8715 338.1512 328.9160 351.0883 391.9328   20  a
## 594.5911 638.8537 655.3690 659.7666 670.9795 716.0702   20  d
```

## Random Out



```
## Unit: milliseconds
##
## MatchBendability(fasta, "con", 3, 0.05, wsize = 20, output.list = F, random.out = T)
## MatchBendability(fasta, "con", 3, 0.05, wsize = 20, output.list = F, random.out = F)
## MatchBendability(fasta, "con", 3, 0.05, wsize = 17, output.list = F, random.out = T)
## MatchBendability(fasta, "con", 3, 0.05, wsize = 17, output.list = F, random.out = F)
## MatchBendability(fasta, "con", 3, 0.05, wsize = 14, output.list = F, random.out = T)
## MatchBendability(fasta, "con", 3, 0.05, wsize = 14, output.list = F, random.out = F)
##      min      lq      mean      median      uq      max      neval      cld
## 395.2287 409.1902 421.6810 413.9224 430.7457 470.6519      20      a
## 432.4120 442.3250 494.6190 452.2158 502.4967 1070.7376      20      b
## 484.1702 491.0497 506.3746 497.6709 506.8615 623.2982      20      b
## 630.7226 650.5915 678.5830 685.2977 702.6332 717.9503      20      c
## 437.9513 448.1384 473.9801 458.5621 510.2918 521.4480      20      ab
## 594.4005 609.1412 631.4784 621.2251 639.9047 751.8879      20      c
```

## ClusterAndConsensus

Intended to present output of function MatchBendability in a more human-friendly format. The output can contain very large numbers of highly similar sequences, so clustering and calculating a consensus can help provide a clearer picture. DISCLAIMER: due to the nature of bendability problem and limitations of IUPAC extended genetic alphabet, output of this function is not intended for further machine use. Not all sequences which can be inferred from consensus output will have satisfied the initial bendability requirements (for example AA/TT will be marked as WW, same as AT/TA).

Parameters that can be used include: - Cutoff - sample.seq

### Cutoff

```
## =====
##
## Time difference of 0.33 secs
##
## =====
##
## Time difference of 0.17 secs
##
## =====
##
## Time difference of 0.12 secs
##
## =====
##
## Time difference of 0.26 secs
##
## =====
##
## Time difference of 0.24 secs
##
## =====
##
## Time difference of 0.29 secs
##
## =====
##
## Time difference of 0.16 secs
##
## =====
##
## Time difference of 0.25 secs
##
## =====
##
## Time difference of 0.12 secs
##
## =====
##
## Time difference of 0.28 secs
##
## =====
##
## Time difference of 0.12 secs
##
## =====
##
## Time difference of 0.33 secs
##
## =====
##
```

```

## Time difference of 0.17 secs
##
## =====
##
## Time difference of 0.12 secs
##
## =====
##
## Time difference of 0.32 secs
##
## =====
##
## Time difference of 0.3 secs
##
## =====
##
## Time difference of 0.27 secs
##
## =====
##
## Time difference of 0.27 secs
##
## =====
##
## Time difference of 0.16 secs
##
## =====
##
## Time difference of 0.25 secs
##
## =====
##
## Time difference of 0.26 secs
##
## =====
##
## Time difference of 0.24 secs
##
## =====
##
## Time difference of 0.12 secs
##
## =====
##
## Time difference of 0.25 secs
##
## =====
##
## Time difference of 0.13 secs
##
## =====
##
## Time difference of 0.28 secs
##

```

```

## =====
##
## Time difference of 0.17 secs
##
## =====
##
## Time difference of 0.27 secs
##
## =====
##
## Time difference of 0.16 secs
##
## =====
##
## Time difference of 0.3 secs
##
## =====
##
## Time difference of 0.29 secs
##
## =====
##
## Time difference of 0.29 secs
##
## =====
##
## Time difference of 0.27 secs
##
## =====
##
## Time difference of 0.25 secs
##
## =====
##
## Time difference of 0.12 secs
##
## =====
##
## Time difference of 0.31 secs
##
## =====
##
## Time difference of 0.29 secs
##
## =====
##
## Time difference of 0.32 secs
##
## =====
##
## Time difference of 0.12 secs
##
## =====
##

```

```

## Time difference of 0.3 secs
##
## =====
##
## Time difference of 0.12 secs
##
## =====
##
## Time difference of 0.26 secs
##
## =====
##
## Time difference of 0.3 secs
##
## =====
##
## Time difference of 0.16 secs
##
## =====
##
## Time difference of 0.29 secs
##
## =====
##
## Time difference of 0.25 secs
##
## =====
##
## Time difference of 0.31 secs
##
## =====
##
## Time difference of 0.14 secs
##
## =====
##
## Time difference of 0.26 secs
##
## =====
##
## Time difference of 0.3 secs
##
## =====
##
## Time difference of 0.12 secs
##
## =====
##
## Time difference of 0.13 secs
##
## =====
##
## Time difference of 0.16 secs
##

```



```

## =====
##
## Time difference of 0.26 secs
##
## =====
##
## Time difference of 0.27 secs
##
## =====
##
## Time difference of 0.29 secs
##
## =====
##
## Time difference of 0.28 secs
##
## =====
##
## Time difference of 0.12 secs
##
## =====
##
## Time difference of 0.13 secs
##
## =====
##
## Time difference of 0.12 secs
##
## =====
##
## Time difference of 0.14 secs
##
## =====
##
## Time difference of 0.2 secs
##
## =====
##
## Time difference of 0.28 secs
##
## =====
##
## Time difference of 0.33 secs
##
## =====
##
## Time difference of 0.16 secs
##
## =====
##
## Time difference of 0.36 secs
##
## =====
##

```

```

## Time difference of 0.3 secs
##
## =====
##
## Time difference of 0.16 secs
##
## =====
##
## Time difference of 0.17 secs
##
## =====
##
## Time difference of 0.3 secs
##
## =====
##
## Time difference of 0.28 secs
##
## =====
##
## Time difference of 0.16 secs
##
## =====
##
## Time difference of 0.3 secs
##
## =====
##
## Time difference of 0.26 secs
##
## =====
##
## Time difference of 0.16 secs
##
## =====
##
## Time difference of 0.29 secs
##
## =====
##
## Time difference of 0.12 secs
##
## =====
##
## Time difference of 0.15 secs
##
## =====
##
## Time difference of 0.3 secs
##
## =====
##
## Time difference of 0.26 secs
##

```

```

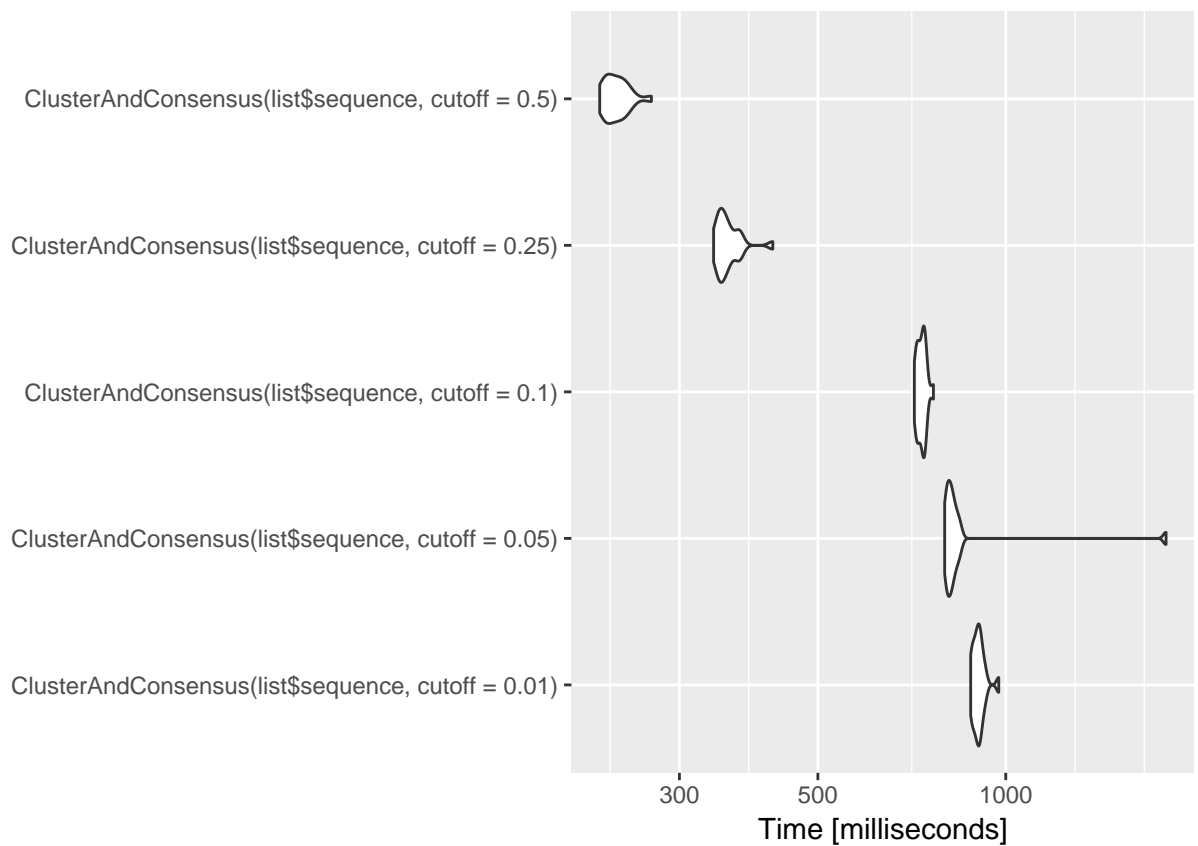
## =====
##
## Time difference of 0.16 secs
##
## =====
##
## Time difference of 0.12 secs
##
## =====
##
## Time difference of 0.17 secs
##
## =====
##
## Time difference of 0.12 secs
##
## =====
##
## Time difference of 0.29 secs
##
## =====
##
## Time difference of 0.16 secs
##
## =====
##
## Time difference of 0.3 secs
##
## =====
##
## Time difference of 0.17 secs
##
## =====
##
## Time difference of 0.24 secs
##
## =====
##
## Time difference of 0.16 secs
##
## =====
##
## Time difference of 0.26 secs
##
## =====
##
## Time difference of 0.26 secs
##
## =====
##
## Time difference of 0.26 secs
##
## =====
##

```

```

## Time difference of 0.25 secs
##
## =====
##
## Time difference of 0.26 secs
##
## =====
##
## Time difference of 0.31 secs
##
## =====
##
## Time difference of 0.13 secs
##
## =====
##
## Time difference of 0.26 secs
##
## =====
##
## Time difference of 0.3 secs
##
## =====
##
## Time difference of 0.25 secs

```



```
## Unit: milliseconds
```

```
##                               expr      min      lq
## ClusterAndConsensus(list$sequence, cutoff = 0.01) 878.5830 891.7041
## ClusterAndConsensus(list$sequence, cutoff = 0.05) 798.2650 806.8557
## ClusterAndConsensus(list$sequence, cutoff = 0.1) 713.7797 721.7999
## ClusterAndConsensus(list$sequence, cutoff = 0.25) 340.3495 348.0929
## ClusterAndConsensus(list$sequence, cutoff = 0.5) 223.5798 229.4834
##      mean      median      uq      max neval  cld
## 905.2323 902.9964 913.2434 974.5034    20    d
## 867.0222 816.8246 829.2678 1806.5150    20    d
## 733.1836 734.0177 741.3867 765.7809    20    c
## 359.4651 353.6879 366.8091 423.4012    20    b
## 238.0616 237.3189 245.6086 270.5406    20    a
```

### Sample.seq

```
## =====
##
## Time difference of 0.02 secs
##
## =====
##
## Time difference of 0.08 secs
##
## =====
##
## Time difference of 0.03 secs
##
## =====
##
## Time difference of 0.02 secs
##
## =====
##
## Time difference of 0.01 secs
##
## =====
##
## Time difference of 0.02 secs
##
## =====
##
## Time difference of 0.05 secs
##
## =====
##
## Time difference of 0.02 secs
##
## =====
##
## Time difference of 0.03 secs
##
## =====
##
## Time difference of 0.02 secs
```

```

##
## =====
##
## Time difference of 0.05 secs
##
## =====
##
## Time difference of 0.04 secs
##
## =====
##
## Time difference of 0.02 secs
##
## =====
##
## Time difference of 0.03 secs
##
## =====
##
## Time difference of 0.02 secs
##
## =====
##
## Time difference of 0.04 secs
##
## =====
##
## Time difference of 0.06 secs
##
## =====
##
## Time difference of 0.03 secs
##
## =====
##
## Time difference of 0.03 secs
##
## =====
##
## Time difference of 0.57 secs
##
## =====
##
## Time difference of 0.04 secs
##
## =====
##
## Time difference of 0.01 secs
##
## =====
##
## Time difference of 0.04 secs
##
## =====

```

```

##
## Time difference of 0.03 secs
##
## =====
##
## Time difference of 0.01 secs
##
## =====
##
## Time difference of 0.05 secs
##
## =====
##
## Time difference of 0.03 secs
##
## =====
##
## Time difference of 0.05 secs
##
## =====
##
## Time difference of 0.06 secs
##
## =====
##
## Time difference of 0.04 secs
##
## =====
##
## Time difference of 0.03 secs
##
## =====
##
## Time difference of 0.04 secs
##
## =====
##
## Time difference of 0.03 secs
##
## =====
##
## Time difference of 0.02 secs
##
## =====
##
## Time difference of 0.03 secs
##
## =====
##
## Time difference of 0.07 secs
##
## =====
##
## Time difference of 0.04 secs

```

```

##
## =====
##
## Time difference of 0.01 secs
##
## =====
##
## Time difference of 0.03 secs
##
## =====
##
## Time difference of 0.03 secs
##
## =====
##
## Time difference of 0.04 secs
##
## =====
##
## Time difference of 0.04 secs
##
## =====
##
## Time difference of 0.02 secs
##
## =====
##
## Time difference of 0.03 secs
##
## =====
##
## Time difference of 0.01 secs
##
## =====
##
## Time difference of 0.06 secs
##
## =====
##
## Time difference of 0.02 secs
##
## =====
##
## Time difference of 0.04 secs
##
## =====
##
## Time difference of 0.06 secs
##
## =====
##
## Time difference of 0.01 secs
##
## =====

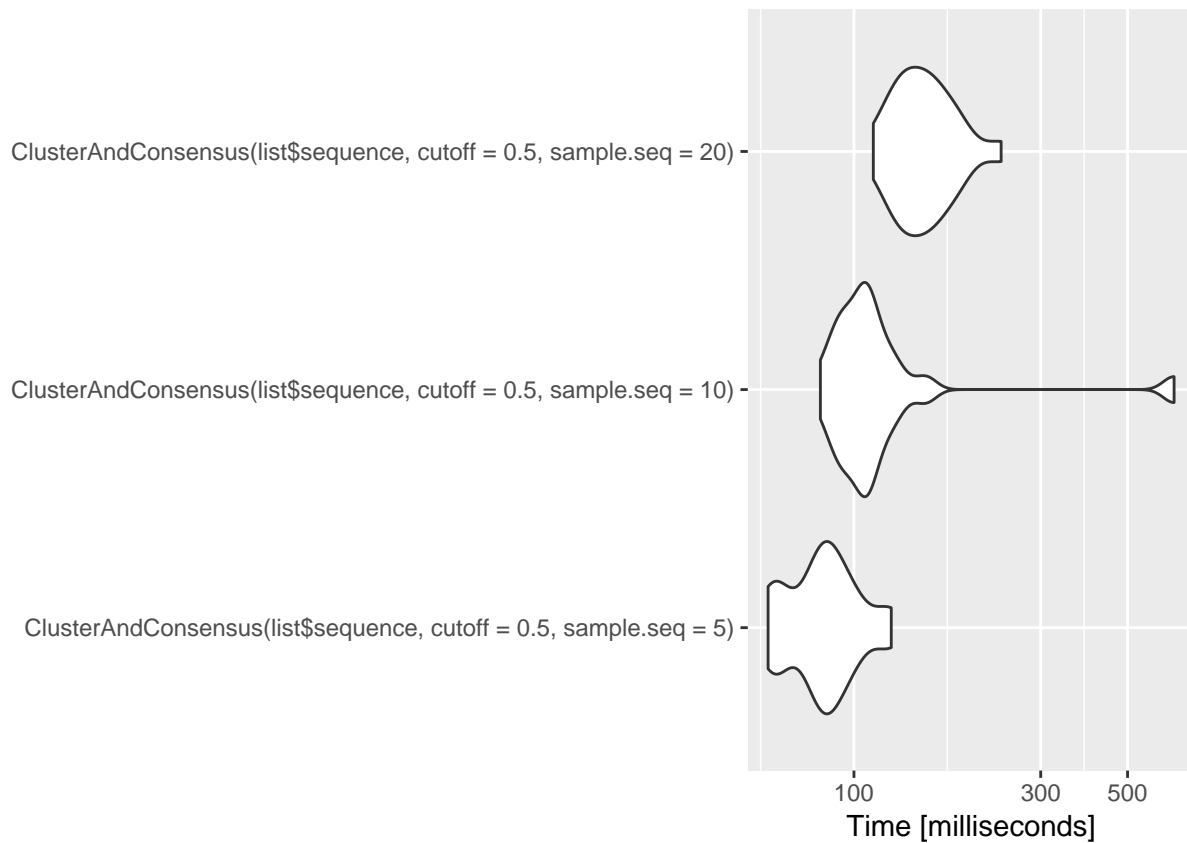
```



```

##
## Time difference of 0.01 secs
##
## =====
##
## Time difference of 0.04 secs
##
## =====
##
## Time difference of 0.07 secs
##
## =====
##
## Time difference of 0.04 secs
##
## =====
##
## Time difference of 0.02 secs
##
## =====
##
## Time difference of 0.02 secs
##
## =====
##
## Time difference of 0.01 secs
##
## =====
##
## Time difference of 0.01 secs
##
## =====
##
## Time difference of 0.04 secs
##
## =====
##
## Time difference of 0.02 secs

```



```
## Unit: milliseconds
##
##      expr
## ClusterAndConsensus(list$sequence, cutoff = 0.5, sample.seq = 5)
## ClusterAndConsensus(list$sequence, cutoff = 0.5, sample.seq = 10)
## ClusterAndConsensus(list$sequence, cutoff = 0.5, sample.seq = 20)
##      min      lq      mean      median      uq      max neval cld
## 60.34674 70.86475 84.73964 83.83441 93.15534 124.5525    20  a
## 82.07551 95.51521 134.22925 106.82965 116.47752 657.4119    20 ab
## 112.06989 131.28195 153.07841 148.38117 166.14075 237.6640    20  b
```