wisard: A package to perform the weighted interval scheduling algorithm

Robert Buecking

January 24, 2022

Contents

1 Installation

requires devtools package

```
> library(devtools)
> setwd(wisard/..)
> install("wisard")
```

2 wisard

2.1 Algorithm

The wisard package contains functions to perform the wighted interval scheduling algorithm on a set of intervals. Primarily, this package is intended to remove overlapping alignments from whole genome alignments created with BLAST, but it can be applied to any other form of weighted intervals to find the highest scoring set of non-overlapping intervals. The weighted interval scheduling algorithm is a dynamic programming algorithm that finds the highest scoring path of compatible intervals through a set of weighted intervals as depicted in figure 1. The functions are easiy to modify with regard to overlap criteria and weighting scheme.

The package additionally contains a function to parse BLAST xml-output with a single subject and query sequence.

The workflow is demonstrated in a short example below. Input can be a BLAST-result as XML, as explained below or any other format transformed into a GRanges object.

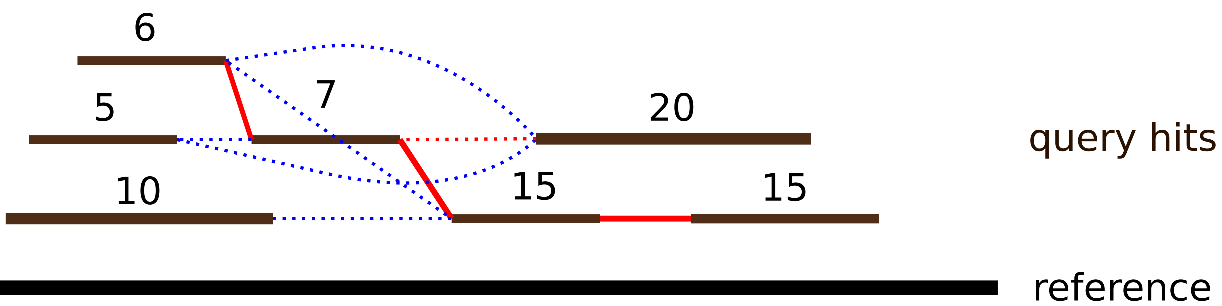

Figure 1: Highest scoring path of compatible intervals

2.2 Reading a BLAST xml-output file

The function read_blast_xml can be used to parse a BLAST xml-output file generated by NCBI-BLAST using -outfmt 5 or AB-BLAST using -mformat=7.

```
> library(wisard)
> # Reading a BLAST xml-file
> xml_file <- system.file("extdata", "example.xml", package = 'wisard')
> blast_result <- read_blast_xml(xml_file, grange_output = T, keep_sequences = F)

[1] "file is from blastn"

> # Metadata contained in xml-file
> blast_result$metadata

  Statistics_db.num Statistics_db.len Statistics_hsp.len Statistics_eff.space
1                 1          10327335                 68          5.60758e+13
  Statistics_kappa Statistics_lambda Statistics_entropy program
1            0.104             0.988                0.4  blastn
                                                        query_name
1 Lactobacillus zhachilii strain HBUAS52074 chromosome, complete genome
      query_ID query_len Parameters_expect Parameters_sc.match
1 NZ_CP031933.2   2714973                10                   1
  Parameters_sc.mismatch Parameters_gap.open Parameters_gap.extend
1                     -1                   1                     2

> # Alignments contained in the xml file
> blast_result$alignments

GRanges object with 549 ranges and 20 metadata columns:
           seqnames           ranges strand |   Hsp_num Hsp_bit.score Hsp_score
              <Rle>        <IRanges>  <Rle> | <numeric>     <numeric> <numeric>
    1 NZ_CP011125.1 8624113-8625463      + |         1       1085.13       759
    2 NZ_CP011125.1 8618246-8619596      + |         2       1085.13       759
    3 NZ_CP011125.1 8620573-8621704      + |         3       921.212       644
    4 NZ_CP011125.1 8614706-8615837      + |         4       921.212       644
    5 NZ_CP011125.1 8621992-8622990      + |         5       537.784       375
  ...           ...             ...    ... .       ...           ...       ...
  545 NZ_CP011125.1 5009294-5009466      - |       545       43.1761        28
  546 NZ_CP011125.1 5009285-5009457      - |       546       43.1761        28
  547 NZ_CP011125.1 6349092-6349267      - |       547       43.1761        28
```

```
548 NZ_CP011125.1 9910129-9910337      - |       548       43.1761        28
549 NZ_CP011125.1 4320937-4321200      - |       549       43.1761        28
      Hsp_evalue Hsp_query.from Hsp_query.to Hsp_hit.from.1 Hsp_query.frame
       <numeric>      <numeric>    <numeric>      <numeric>       <numeric>
  1            0        1993721      1995068        8624113               1
  2            0        1993721      1995068        8618246               1
  3 2.72951e-264        1990664      1991793        8620573               1
  4 2.72951e-264        1990664      1991793        8614706               1
  5  7.2365e-149        1992046      1993036        8621992               1
...          ...            ...          ...            ...             ...
545      5.64262         527497       527669        5009294               1
546      5.64262         527425       527597        5009285               1
547      5.64262        2096151      2096326        6349092               1
548      5.64262         431628       431836        9910129               1
549      5.64262        2039609      2039869        4320937               1
    Hsp_hit.frame Hsp_identity Hsp_positive  Hsp_gaps Hsp_align.len   seq_len
        <numeric>    <numeric>    <numeric> <numeric>     <numeric> <numeric>
  1             1         1063         1063         7          1353  10327335
  2             1         1063         1063         7          1353  10327335
  3             1          892          892         4          1133  10327335
  4             1          892          892         4          1133  10327335
  5             1          716          716        28          1009  10327335
...           ...          ...          ...       ...           ...       ...
545            -1          103          103         2           174  10327335
546            -1          103          103         2           174  10327335
547            -1          102          102      <NA>           176  10327335
548            -1          121          121         2           210  10327335
549            -1          151          151         5           265  10327335
    query_len      query_id raw_score        ID filter_pass
    <numeric>   <character> <numeric> <integer>   <logical>
  1   2714973 NZ_CP031933.2       759         1        TRUE
  2   2714973 NZ_CP031933.2       759         2        TRUE
  3   2714973 NZ_CP031933.2       644         3        TRUE
  4   2714973 NZ_CP031933.2       644         4        TRUE
  5   2714973 NZ_CP031933.2       375         5        TRUE
...       ...           ...       ...       ...         ...
545   2714973 NZ_CP031933.2        28       545        TRUE
546   2714973 NZ_CP031933.2        28       546        TRUE
547   2714973 NZ_CP031933.2        28       547        TRUE
548   2714973 NZ_CP031933.2        28       548        TRUE
549   2714973 NZ_CP031933.2        28       549        TRUE
                   query_ranges
                      <GRanges>
  1 NZ_CP031933.2:1993721-1995068
  2 NZ_CP031933.2:1993721-1995068
  3 NZ_CP031933.2:1990664-1991793
  4 NZ_CP031933.2:1990664-1991793
  5 NZ_CP031933.2:1992046-1993036
...                           ...
545   NZ_CP031933.2:527497-527669
546   NZ_CP031933.2:527425-527597
```

```
547 NZ_CP031933.2:2096151-2096326
548   NZ_CP031933.2:431628-431836
549 NZ_CP031933.2:2039609-2039869
-------
seqinfo: 1 sequence from an unspecified genome
```

2.3 Caclulating the BLAST sum score

```
> # calculating the BLAST sum score of a set of alignments:
> sum_score(blast_result$alignments$raw_score, blast_result$metadata)
```

```
[1] 25028.66
```

2.4 Filter BLAST results

The function filter_blast can be used to filter the blast results

```
> # filter records:
> blast_result$alignments <- filter_blast(blast_result$alignments,
+                                         min_len = 10,
+                                         max_e = 1,
+                                         min_identity = 0)
```

```
[1] "Number of alignments filtered out"
[1] 92
[1] "Number of alignments that passed filter"
[1] 457
```

2.5 Run the WIS algorithm

The function run_get_wis runs the WIS algorithm

```
> # run wis
> wis_results <- run_get_wis(blast_result$alignments, is_circular = T,
+                            use_strand = T,
+                            score_col = "raw_score")
```

```
[1] "strand yes, frame no"
[1] "analyzing + strand"
calculate wis for start alignment  1
max score is:  11617
[1] "analyzing - strand"
calculate wis for start alignment  1
max score is:  10515
```

2.6 Run the greedy algorithm

The function greedy_algorithm runs the WIS algorithm

```
> # run greedy
> greedy_results <- greedy_algorithm(blast_result$alignments,
+                                    max_score = "raw_score",
+                                    is_circular = T,
+                                    use_strand = T)
```

```
[1] "in greedy"

> length(blast_result$alignments)

[1] 457

> length(greedy_results)

[1] 291
```