

# Template Matching Workflow

GLMRosa

## Template Matching Workflow

### Preparando o ambiente

Carregando os pacotes necessários

```
library(dplyr)
```

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

`filter`, `lag`

The following objects are masked from 'package:base':

`intersect`, `setdiff`, `setequal`, `union`

```
library(dplyr)
library(progressr)
library(here)
```

`here()` starts at `/home/grosa/R_repos/monitoraSom`

```
library(collapse)
```

collapse 1.9.5, see `?`collapse-package`` or `?`collapse-documentation``

Attaching package: 'collapse'

The following object is masked from 'package:stats':

D

```
library(purrr)
library(furrr)
```

Loading required package: future

```
library(av)
library(tuneR)
```

Attaching package: 'tuneR'

The following object is masked from 'package:av':

sine

```
library(seewave)
library(data.table)
```

Attaching package: 'data.table'

The following object is masked from 'package:purrr':

transpose

The following objects are masked from 'package:dplyr':

between, first, last

```
library(dtwclust)
```

Loading required package: proxy

Attaching package: 'proxy'

The following objects are masked from 'package:stats':

```
as.dist, dist
```

The following object is masked from 'package:base':

```
as.matrix
```

Loading required package: dtw

Loaded dtw v1.23-1. See ?dtw for help, citation("dtw") for use in publication.

dtwclust:

Setting random number generator to L'Ecuyer-CMRG (see RNGkind()).

To read the included vignettes type: browseVignettes("dtwclust").

See news(package = "dtwclust") after package updates.

```
library(slider)
library(ggplot2)
library(farver)
library(parallel)
library(doParallel)
```

Loading required package: foreach

Attaching package: 'foreach'

The following objects are masked from 'package:purrr':

```
accumulate, when
```

Loading required package: iterators

Para o pacote here funcionar corretamente, é necessário que este script esteja em uma sessão baseada no projeto localizado na raiz do repositório

```
path_base <- here()
path_soundscapes <- here("example", "soundscapes")
path_templates <- here("example", "roi_cuts")
path_data <- here("example", "data")
path_backup <- here("example", "backup")
path_plots <- here("example", "plots")
path_scripts <- here("R/")
# path_scripts <- "C:/R_repos/monitoraSom/R/"

c(
  dir.exists(path_soundscapes), dir.exists(path_templates),
  dir.exists(path_data), dir.exists(path_backup),
  dir.exists(path_plots), dir.exists(path_scripts)
)
```

```
[1] TRUE TRUE TRUE TRUE TRUE TRUE
```

Enquanto ainda não exist

```
invisible(
  list.files(path_scripts, full.names = TRUE) %>%
    gsub("//", "/", .) %>%
    map(~ source(.x))
)
```

1. Get template metadata 1.a. Get metadata from standalone cuts

```
df_templates_A <- fetch_template_metadata(
  path = here("example", "roi_cuts"), method = "standalone"
)
```

Template metadata successfully extracted

```
glimpse(df_templates_A)
```

Rows: 6

Columns: 10

```
$ template_path      <chr> "/home/grosa/R_repos/monitoraSom/example/roi_cuts~
$ template_file      <chr> "W05048607S0586408_20210812_091000_013.173-015.83~
$ template_label     <chr> "Lipaugus vociferans", "Lipaugus vociferans", "Li~
$ template_start     <dbl> 0, 0, 0, 0, 0, 0
$ template_end       <dbl> 2.657146, 2.155792, 2.306208, 6.166563, 6.266833,~
$ template_sample_rate <int> 48000, 48000, 48000, 48000, 48000, 48000
$ template_min_freq  <dbl> 1.186, 1.186, 1.186, 1.186, 1.186, 1.186
$ template_max_freq  <dbl> 5.567, 5.567, 5.567, 5.567, 5.567, 5.567
$ template_wl        <dbl> 2048, 2048, 2048, 2048, 2048, 2048
$ template_ovlp      <dbl> 0, 0, 0, 0, 0, 0
```

1.b. Get metadata from ROI tables

```
df_templates_B <- fetch_template_metadata(
  path = here("example", "roi_tables"), method = "roi_table"
)
```

Template metadata successfully extracted

```
glimpse(df_templates_B)
```

Rows: 6

Columns: 10

```
$ template_path      <chr> "/home/grosa/R_repos/MonitoraSomAlt/example/sound~
$ template_file      <chr> "W05048607S0586408_20210812_091000.wav", "W050486~
$ template_label     <chr> "Lipaugus vociferans", "Lipaugus vociferans", "Li~
$ template_start     <dbl> 13.17317, 15.37910, 35.08203, 49.57095, 48.26745,~
$ template_end       <dbl> 15.83031, 17.53489, 37.38822, 55.73752, 54.53429,~
$ template_sample_rate <int> 48000, 48000, 48000, 48000, 48000, 48000
$ template_min_freq  <dbl> 1.1855005, 0.8992885, 0.9433211, 1.0754190, 1.097~
$ template_max_freq  <dbl> 5.566746, 5.368599, 5.258517, 1.757924, 1.669859,~
$ template_wl        <int> 2048, 2048, 2048, 2048, 2048, 2048
$ template_ovlp      <int> 0, 0, 0, 0, 0, 0
```

2. Get soundscape metadata

```
df_soundscape_metadata <- fetch_soundscape_metadata(
  path = path_soundscape, ncores = 1
)
```

```
)
```

Soundscape metadata successfully extracted

```
glimpse(df_soundscape)
```

Rows: 34

Columns: 6

```
$ soundscape_path      <chr> "/home/grosa/R_repos/monitoraSom/example/sounds~
$ soundscape_file      <chr> "W05030625S0609383_20221012_080000.wav", "W0504~
$ soundscape_duration  <dbl> 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, ~
$ soundscape_sample_rate <int> 48000, 48000, 48000, 48000, 48000, 48000, 48000~
$ soundscape_codec     <chr> "pcm_s16le", "pcm_s16le", "pcm_s16le", "pcm_s16~
$ soundscape_layout    <chr> "1 channels", "1 channels", "1 channels", "1 ch~
```

3. Get match grid

```
df_grid <- fetch_match_grid(
  soundscape_data = df_soundscape, template_data = df_templates_A
)
```

All files locally available, are compatible and resulted in a grid of 204 matchings

```
glimpse(df_grid)
```

Rows: 204

Columns: 16

```
$ soundscape_path      <chr> "/home/grosa/R_repos/monitoraSom/example/sounds~
$ soundscape_file      <chr> "W05030625S0609383_20221012_080000.wav", "W0503~
$ soundscape_duration  <dbl> 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, ~
$ soundscape_sample_rate <int> 48000, 48000, 48000, 48000, 48000, 48000, 48000~
$ soundscape_codec     <chr> "pcm_s16le", "pcm_s16le", "pcm_s16le", "pcm_s16~
$ soundscape_layout    <chr> "1 channels", "1 channels", "1 channels", "1 ch~
$ template_path        <chr> "/home/grosa/R_repos/monitoraSom/example/roi_cu~
$ template_file        <chr> "W05048607S0586408_20210812_091000_013.173-015.~
$ template_label       <chr> "Lipaugus vociferans", "Lipaugus vociferans", "~
$ template_start       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ template_end         <dbl> 2.657146, 2.155792, 2.306208, 6.166563, 6.26683~
```

```
$ template_sample_rate <int> 48000, 48000, 48000, 48000, 48000, 48000, 48000~
$ template_min_freq <dbl> 1.186, 1.186, 1.186, 1.186, 1.186, 1.186, 1.186~
$ template_max_freq <dbl> 5.567, 5.567, 5.567, 5.567, 5.567, 5.567, 5.567~
$ template_wl <dbl> 2048, 2048, 2048, 2048, 2048, 2048, 2048, 2048,~
$ template_ovlp <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
```

#### 4. Match templates to soundscape 4.a. Match templates to soundscape using correlation

```
df_matches_cor <- match_n(
  df_grid = df_grid, score_method = "cor",
  ncores = 8, par_strat = "future",
  save_res =
    "/home/grosa/R_repos/MonitoraSomDev/example/data/matches/matches_cor.rds"
)
```

Warning in supportsMulticoreAndRStudio(...): [ONE-TIME WARNING] Forked processing ('multicore') is not supported when running R from RStudio because it is considered unstable. For more details, how to control forked processing or not, and how to silence this warning in future R sessions, see `?parallelly::supportsMulticore`

Template matching completed

```
glimpse(df_matches_cor)
```

Rows: 204

Columns: 19

```
$ soundscape_path <chr> "/home/grosa/R_repos/monitoraSom/example/sounds~
$ soundscape_file <chr> "W05030625S0609383_20221012_080000.wav", "W0503~
$ soundscape_duration <dbl> 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60,~
$ soundscape_sample_rate <int> 48000, 48000, 48000, 48000, 48000, 48000, 48000~
$ soundscape_codec <chr> "pcm_s16le", "pcm_s16le", "pcm_s16le", "pcm_s16~
$ soundscape_layout <chr> "1 channels", "1 channels", "1 channels", "1 ch~
$ template_path <chr> "/home/grosa/R_repos/monitoraSom/example/roi_cu~
$ template_file <chr> "W05048607S0586408_20210812_091000_013.173-015.~
$ template_label <chr> "Lipaugus vociferans", "Lipaugus vociferans", "~
$ template_start <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ template_end <dbl> 2.657146, 2.155792, 2.306208, 6.166563, 6.26683~
$ template_sample_rate <int> 48000, 48000, 48000, 48000, 48000, 48000, 48000~
$ template_min_freq <dbl> 1.186, 1.186, 1.186, 1.186, 1.186, 1.186, 1.186~
```

```

$ template_max_freq      <dbl> 5.567, 5.567, 5.567, 5.567, 5.567, 5.567, 5.567~
$ template_wl            <dbl> 2048, 2048, 2048, 2048, 2048, 2048, 2048, 2048,~
$ template_ovlp          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ score_sliding_window   <int> 62, 50, 54, 144, 146, 186, 62, 50, 54, 144, 146~
$ score_method           <chr> "cor", "cor", "cor", "cor", "cor", "cor", "cor",~
$ score_vec              <list> [<data.frame[1406 x 2]>], [<data.frame[1406 x ~

```

## 5. Get detections

```

df_detections <- fetch_score_peaks_n(
  tib_match = df_matches_cor, buffer_size = "template"
)

```

Detections extracted from scores

```
glimpse(df_detections)
```

Rows: 1,208

Columns: 18

```

$ soundscape_path      <chr> "/home/grosa/R_repos/monitoraSom/example/soundsc~
$ soundscape_file      <chr> "W05030625S0609383_20221012_080000.wav", "W05030~
$ template_path        <chr> "/home/grosa/R_repos/monitoraSom/example/roi_cut~
$ template_file         <chr> "W05048607S0586408_20210812_091000_013.173-015.8~
$ template_min_freq    <dbl> 1.186, 1.186, 1.186, 1.186, 1.186, 1.186, 1.186,~
$ template_max_freq    <dbl> 5.567, 5.567, 5.567, 5.567, 5.567, 5.567, 5.567,~
$ detection_start       <dbl> 3.288256, 10.676157, 14.092527, 24.597865, 29.89~
$ detection_end         <dbl> 5.935943, 13.323843, 16.740214, 27.245552, 32.54~
$ detection_wl          <dbl> 2048, 2048, 2048, 2048, 2048, 2048, 2048, 2048, ~
$ detection_ovlp        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ detection_sample_rate <int> 48000, 48000, 48000, 48000, 48000, 48000, 48000,~
$ detection_buffer      <int> 62, 62, 62, 62, 62, 62, 62, 62, 62, 62, 50, 50, ~
$ detec_min_score       <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
$ detec_min_quant       <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
$ detec_top_n           <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
$ peak_index            <dbl> 109, 282, 362, 608, 732, 896, 964, 1062, 1171, 1~
$ peak_score            <dbl> 0.02810765, 0.04885051, 0.03755142, 0.06236631, ~
$ peak_quant            <dbl> 0.967, 0.998, 0.988, 1.000, 0.991, 0.936, 0.954,~

```

## 6. Whole workflow in a single pipeline



```

#|
df_detections <- fetch_match_grid(
  template_data = fetch_template_metadata(
    path = here("example", "roi_cuts"), method = "standalone"
  ),
  soundscape_data = fetch_soundscape_metadata(
    path = path_soundscales, ncores = 6
  )
) %>%
  match_n(score_method = "cor", ncores = 8, par_strat = "foreach") %>%
  fetch_score_peaks_n(buffer_size = "template") %>%
  glimpse()

```

Soundscape metadata successfully extracted

Template metadata successfully extracted

All files locally available, are compatible and resulted in a grid of 204 matchings

Template matching completed

Detections extracted from scores

Rows: 1,208

Columns: 18

```

$ soundscape_path      <chr> "/home/grosa/R_repos/monitoraSom/example/soundsc~
$ soundscape_file      <chr> "W05030625S0609383_20221012_080000.wav", "W05030~
$ template_path        <chr> "/home/grosa/R_repos/monitoraSom/example/roi_cut~
$ template_file        <chr> "W05048607S0586408_20210812_091000_013.173-015.8~
$ template_min_freq    <dbl> 1.186, 1.186, 1.186, 1.186, 1.186, 1.186, 1.186, ~
$ template_max_freq    <dbl> 5.567, 5.567, 5.567, 5.567, 5.567, 5.567, 5.567, ~
$ detection_start      <dbl> 3.288256, 10.676157, 14.092527, 24.597865, 29.89~
$ detection_end        <dbl> 5.935943, 13.323843, 16.740214, 27.245552, 32.54~
$ detection_wl         <dbl> 2048, 2048, 2048, 2048, 2048, 2048, 2048, 2048, ~
$ detection_ovlp       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ detection_sample_rate <int> 48000, 48000, 48000, 48000, 48000, 48000, 48000, ~
$ detection_buffer     <int> 62, 62, 62, 62, 62, 62, 62, 62, 62, 62, 62, 50, 50, ~
$ detec_min_score      <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
$ detec_min_quant      <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
$ detec_top_n          <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~

```

```
$ peak_index      <dbl> 109, 282, 362, 608, 732, 896, 964, 1062, 1171, 1~
$ peak_score      <dbl> 0.02810765, 0.04885051, 0.03755142, 0.06236631, ~
$ peak_quant      <dbl> 0.967, 0.998, 0.988, 1.000, 0.991, 0.936, 0.954,~
```

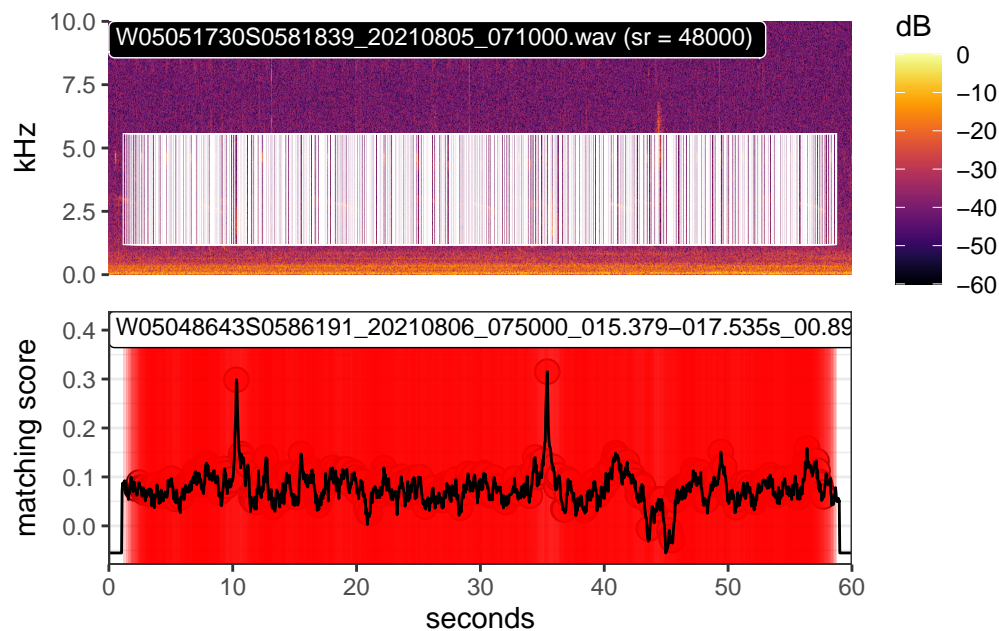
## 7. Whole workflow in a single function

```
# df_detections <- template_matching(
#   path_soundscapes = here("example", "soundscapes"),
#   path_templates = here("example", "roi_cuts"),
#   template_type = "standalone", score_method = "cor",
#   buffer_size = "template", min_score = NA, min_quant = NA, top_n = NA,
#   ncores = 8, par_strat = "foreach" # todo Implementação pendente
# )
```

## 8. Plotting 8.a. Without filters

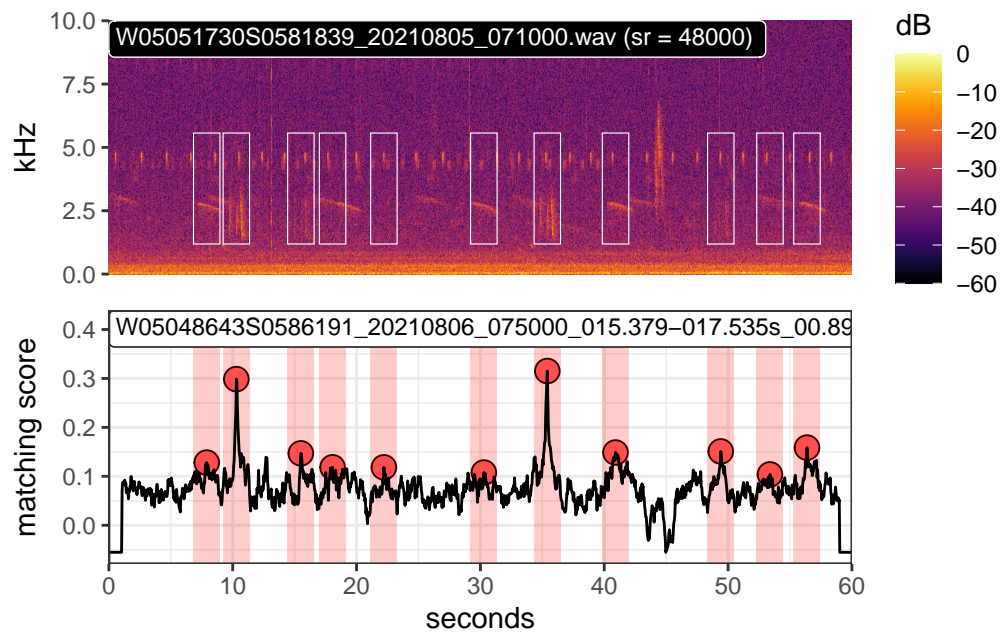
```
plot_match_i(match_i_res = df_matches_cor[188, ], buffer_size = 0)
```

Loading required package: patchwork



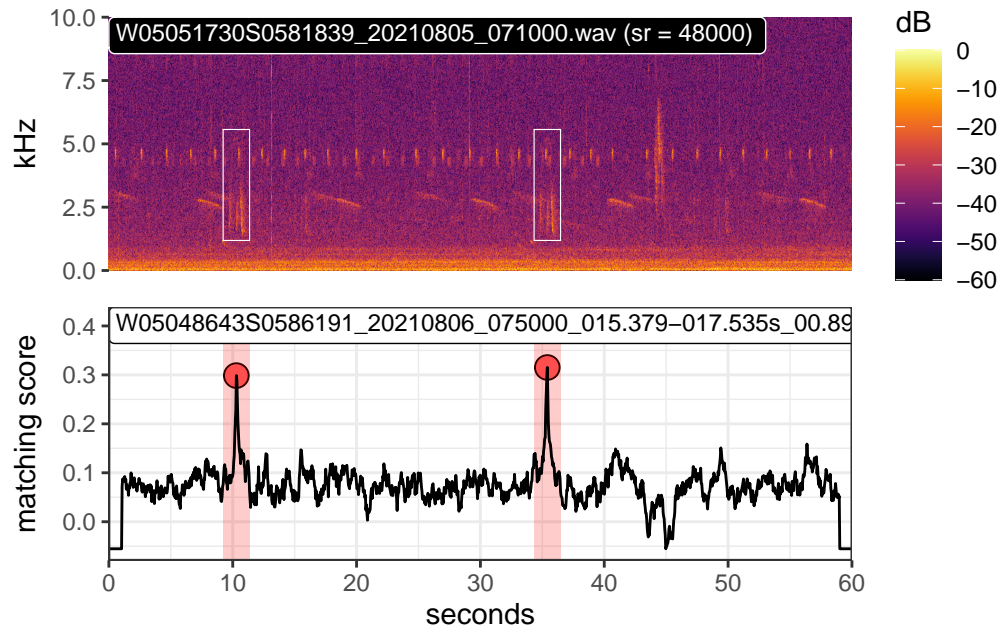
## 8.b. With template buffer

```
plot_match_i(df_matches_cor[188, ], buffer_size = "template")
```



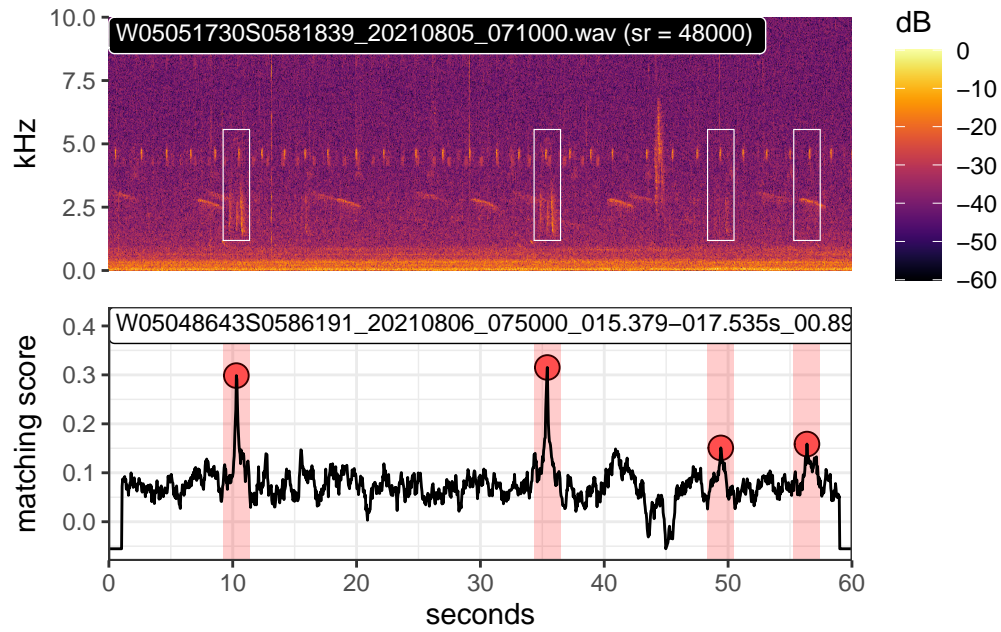
8.c. With min\_score (cutoff) filter

```
plot_match_i(df_matches_cor[188, ], buffer_size = 0, min_score = 0.2)
```



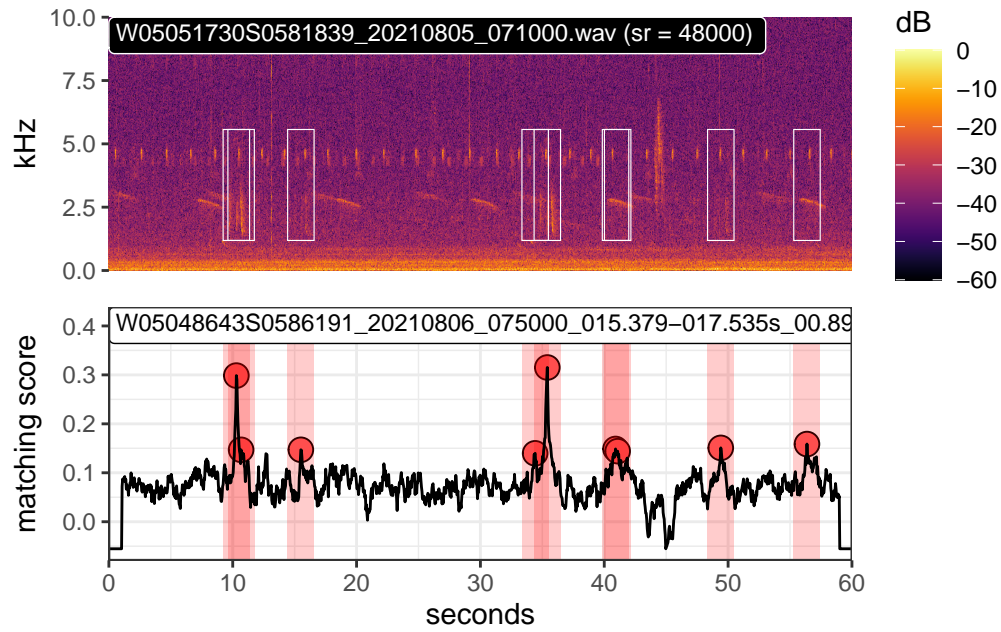
8.d. With top\_n filter

```
plot_match_i(df_matches_cor[188, ], buffer_size = 0, top_n = 4)
```



8.e. With quantile filter without buffer

```
plot_match_i(df_matches_cor[188, ], buffer_size = 0, min_quant = 0.975)
```



with buffer

```
plot_match_i(df_matches_cor[188, ], buffer_size = "template", min_quant = 0.975)
```

