

# benthic species-habitat dashboards

Tom Webb

16/03/2021

## Producing habitat summaries and visual dashboards for European marine benthic species

The processed EMODnet benthic numerical abundance product files, now matched to sediment properties and the EMODnet broadscale seabed habitat map, are here read in and used to feed functions to provide a summary of the habitat affinities of individual named benthic species. Where habitat affinity data is available for that species from Biotic, this is included too.

First, load required packages:

```
library(tidyverse)
library(ggribbles)
library(worrms)
library(ggtext)
library(patchwork)
```

Now, load the datasets (column types are fully specified for benth\_events to avoid parsing errors):

```
benth_events <- read_csv(here::here(
  "data", "derived_data/benthic_abundance_sampling_events_seabed_habs.csv"),
  col_types = cols(
    datasetid = col_double(),
    datecollected = col_datetime(format = ""),
    decimallongitude = col_double(),
    decimallatitude = col_double(),
    minimumdepthinmeters = col_double(),
    sampid = col_double(),
    eventNumber = col_double(),
    MudPercent = col_double(),
    SandPercent = col_double(),
    GravelPercent = col_double(),
    TotalD50 = col_double(),
    SandD50 = col_double(),
    GravelD50 = col_double(),
    Rock10cm = col_double(),
    Rock50cm = col_double(),
    log_D50 = col_double(),
    Biozone = col_character(),
    Energy = col_character(),
    Substrate = col_character(),
    Salinity = col_character(),
    Oxygen = col_character(),
    EUNIScomb = col_character(),
```

```

    EUNIScombD = col_character(),
    Allcomb = col_character(),
    AllcombD = col_character(),
    SalcombD = col_character(),
    MSFD_BBHT = col_character()
  ))
benth_abundances <- read_csv(here::here(
  "data", "derived_data/benthic_abundances_long.csv"))
benth_taxa <- read_csv(here::here(
  "data", "derived_data/benthic_taxa.csv"))
benth_substrate_prefs <- read_csv(here::here(
  "data", "derived_data/benthic_species_substratum_prefs.csv"))
benth_substrate_key <- read_csv(here::here(
  "data", "derived_data/substrate_values_key.csv"))

```

## Getting habitat summaries for a single species

This code loads a function which takes a species Aphia ID and returns basic summary information about the benthic habitats that it has been recorded in (from the EMODnet numerical abundance product). For continuous variables (e.g. ‘Percent Mud’) the function returns the arithmetic mean value of the variable from all sampling events the species was recorded in, weighted by its abundance in each event. For categorical variables (e.g. ‘Substrate’) the function returns the relative frequency of occurrence in each category (again weighted by abundance). Finally, the function checks if the species is one of those occurring in our Biotic substrate dataset, and adds relevant information about its recorded habitat preferences if so.

```
source(here::here("scripts", "get_species_habitats.R"))
```

An example of running for one species:

```
get_species_habitats(103228)
```

```

## # A tibble: 1 x 80
##   AphiaID total_occ total_ab mean_ab GravelD50 GravelPercent log_D50 MudPercent
##   <dbl>      <int>      <dbl>  <dbl>      <dbl>          <dbl>  <dbl>      <dbl>
## 1  103228      3002  154156.   51.4      8.06          22.5  -0.208    3.03
## # ... with 72 more variables: Rock10cm <dbl>, Rock50cm <dbl>, SandD50 <dbl>,
## #   SandPercent <dbl>, TotalD50 <dbl>, Energy_High energy <dbl>,
## #   Energy_Low energy <dbl>, Energy_Moderate energy <dbl>,
## #   Energy_No energy information <dbl>, Energy_NA <dbl>,
## #   Biozone_Arctic mid bathyal <dbl>, Biozone_Atlantic upper bathyal <dbl>,
## #   Biozone_Atlanto-Arctic upper bathyal <dbl>,
## #   Biozone_Deep circalittoral <dbl>, Biozone_Infralittoral <dbl>,
## #   Biozone_Shallow circalittoral <dbl>, Biozone_NA <dbl>,
## #   Substrate_Coarse substrate <dbl>, Substrate_Fine mud <dbl>,
## #   Substrate_Mixed sediment <dbl>, Substrate_Muddy sand <dbl>,
## #   Substrate_Rock or other hard substrata <dbl>, Substrate_Sand <dbl>,
## #   Substrate_Sandy mud <dbl>, Substrate_Sandy mud or Muddy sand <dbl>,
## #   Substrate_Seabed <dbl>, Substrate_Sediment <dbl>, Substrate_NA <dbl>,
## #   Salinity_NA <dbl>, Oxygen_NA <dbl>, EUNIScomb_A3.1 <dbl>,
## #   EUNIScomb_A4.1 <dbl>, EUNIScomb_A4.2 <dbl>, EUNIScomb_A4.27 <dbl>,
## #   EUNIScomb_A4.3 <dbl>, EUNIScomb_A5 <dbl>, EUNIScomb_A5.13 <dbl>,
## #   EUNIScomb_A5.14 <dbl>, EUNIScomb_A5.15 <dbl>,
## #   EUNIScomb_A5.23 or A5.24 <dbl>, EUNIScomb_A5.25 or A5.26 <dbl>,
## #   EUNIScomb_A5.27 <dbl>, EUNIScomb_A5.33 <dbl>, EUNIScomb_A5.35 <dbl>,

```

```
## # EUNIScomb_A5.36 <dbl>, EUNIScomb_A5.37 <dbl>, EUNIScomb_A5.43 <dbl>,
## # EUNIScomb_A5.44 <dbl>, EUNIScomb_A5.45 <dbl>, EUNIScomb_A6 <dbl>,
## # EUNIScomb_A6.3 or A6.4 <dbl>, EUNIScomb_A6.5 <dbl>, EUNIScomb_Na <dbl>,
## # EUNIScomb_NA <dbl>, MSFD_BBHT_Circalittoral coarse sediment <dbl>,
## # MSFD_BBHT_Circalittoral mixed sediment <dbl>,
## # MSFD_BBHT_Circalittoral mud <dbl>,
## # MSFD_BBHT_Circalittoral rock and biogenic reef <dbl>,
## # MSFD_BBHT_Circalittoral sand <dbl>,
## # MSFD_BBHT_Infralittoral coarse sediment <dbl>,
## # MSFD_BBHT_Infralittoral mixed sediment <dbl>,
## # MSFD_BBHT_Infralittoral mud <dbl>,
## # MSFD_BBHT_Infralittoral rock and biogenic reef <dbl>,
## # MSFD_BBHT_Infralittoral sand <dbl>, MSFD_BBHT_Na <dbl>,
## # MSFD_BBHT_Offshore circalittoral coarse sediment <dbl>,
## # MSFD_BBHT_Offshore circalittoral mixed sediment <dbl>,
## # MSFD_BBHT_Offshore circalittoral mud <dbl>,
## # MSFD_BBHT_Offshore circalittoral rock and biogenic reef <dbl>,
## # MSFD_BBHT_Offshore circalittoral sand <dbl>,
## # MSFD_BBHT_Upper bathyal sediment <dbl>, MSFD_BBHT_NA <dbl>
```

## Getting habitat summaries for all

This simply runs the above function over all species:

```
sp_habitat_summaries <- benth_taxa %>%
  mutate(aphia = AphiaID) %>%
  group_by(aphia) %>%
  group_modify(~ get_species_habitats(sp_id = .$AphiaID)) %>%
  ungroup() %>%
  dplyr::select(-aphia)
```

To neaten up this output, gather together similar columns:

```
sp_habitat_summaries <- sp_habitat_summaries %>%
  dplyr::select(AphiaID:mean_ab, inf_epi,
    TotalD50, log_D50, GravelD50, GravelPercent,
    MudPercent:SandPercent,
    starts_with("Energy"),
    starts_with("Biozone"),
    starts_with("Substrate"),
    starts_with("Salinity"),
    starts_with("EUNIS"),
    starts_with("MSFD"),
    everything())
```

For comparative purposes, it is also useful to have summaries of the distributions and frequencies of habitat types across all sampling events. This loads a function to do that:

```
source(here::here("scripts", "get_event_habitats.R"))
```

So means / frequencies of different habitat types are:

```
event_habitat_summaries <- get_event_habitats()
```

We can write these two data products to file:



## Reproducibility

Reproducibility receipt

```
## datetime
```

```
Sys.time()
```

```
## [1] "2021-03-31 09:39:50 BST"
```

```
## repository
```

```
git2r::repository()
```

```
## Local:      master /Users/tom/Google Drive/emodnet habitats/EMODnet_occs_habs
```

```
## Remote:     master @ origin (https://github.com/EMODnet/EMODnet-Biology-Benthic-Habitats-Occurrences-T
```

```
## Head:       [cda3de7] 2021-03-31: Merge branch 'master' of https://github.com/EMODnet/EMODnet-Biology-B
```

```
## session info
```

```
sessionInfo()
```

```
## R version 3.6.2 (2019-12-12)
```

```
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
```

```
## Running under: macOS Catalina 10.15.7
```

```
##
```

```
## Matrix products: default
```

```
## BLAS:      /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
```

```
## LAPACK:    /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
```

```
##
```

```
## locale:
```

```
## [1] en_GB.UTF-8/en_GB.UTF-8/en_GB.UTF-8/C/en_GB.UTF-8/en_GB.UTF-8
```

```
##
```

```
## attached base packages:
```

```
## [1] stats      graphics  grDevices  utils      datasets  methods   base
```

```
##
```

```
## other attached packages:
```

```
## [1] patchwork_1.0.0 ggtext_0.1.1    worrms_0.4.0    ggribes_0.5.3
```

```
## [5] forcats_0.4.0  stringr_1.4.0  dplyr_1.0.4    purrr_0.3.4
```

```
## [9] readr_1.3.1    tidyr_1.0.0    tibble_3.0.6   ggplot2_3.3.3
```

```
## [13] tidyverse_1.3.0
```

```
##
```

```
## loaded via a namespace (and not attached):
```

```
## [1] Rcpp_1.0.6      lubridate_1.7.4  here_0.1        assertthat_0.2.1
```

```
## [5] rprojroot_1.3-2 digest_0.6.27    utf8_1.1.4      R6_2.5.0
```

```
## [9] cellranger_1.1.0 plyr_1.8.6      backports_1.1.5  reprex_0.3.0
```

```
## [13] evaluate_0.14   highr_0.8       httr_1.4.2      pillar_1.5.0
```

```
## [17] rlang_0.4.10    curl_4.3        readxl_1.3.1    rstudioapi_0.13
```

```
## [21] rmarkdown_2.7   labeling_0.3     urltools_1.7.3  triebeard_0.3.0
```

```
## [25] munsell_0.5.0   gridtext_0.1.4  broom_0.7.2     compiler_3.6.2
```

```
## [29] modelr_0.1.5    xfun_0.21       pkgconfig_2.0.3  htmltools_0.5.1.1
```

```
## [33] tidyselect_1.1.0 httpcode_0.2.0  fansi_0.4.2     crayon_1.4.1
```

```
## [37] dbplyr_1.4.2    withr_2.1.2     crul_0.9.0      grid_3.6.2
```

```
## [41] jsonlite_1.7.2  gtable_0.3.0    lifecycle_1.0.0 DBI_1.1.1
```

```
## [45] git2r_0.26.1    magrittr_2.0.1  scales_1.1.0    cli_2.3.1
```

```
## [49] stringi_1.5.3   farver_2.0.3    fs_1.3.1        xml2_1.3.2
```

```
## [53] ellipsis_0.3.1  generics_0.1.0  vctrs_0.3.6     tools_3.6.2
```

```
## [57] glue_1.4.2      markdown_1.1    hms_0.5.3       yaml_2.2.1
```

```
## [61] colorspace_1.4-1 rvest_0.3.5     knitr_1.31      haven_2.2.0
```