EASI-FishR vignette

Introduction

Here, we provide example code and application of the Ecological Assessment of the Sustainable Impacts of Fisheries (EASI-Fish) ecological risk assessment approach developed by Griffiths et al. (2019) and detailed in Hill et al. (in prep). We apply EASI-Fish to two example species, Alopias vulpinus (ALV) and Carcharhinus longimanus (OCS) and two fisheries to show how this approach can be applied to multiple species and fisheries simultaneously.

Preparing life history data

We read in a csv document listing the various life history parameters for two species and explore these to make sure they look reasonable. This csv has the structure of a param column that names all the parameters you can input, and then each of the following columns represents a species information. For parameters that can be varied, there are rows to input standard deviation, standard error, and distribution type (e.g. 'fixed', 'normal').

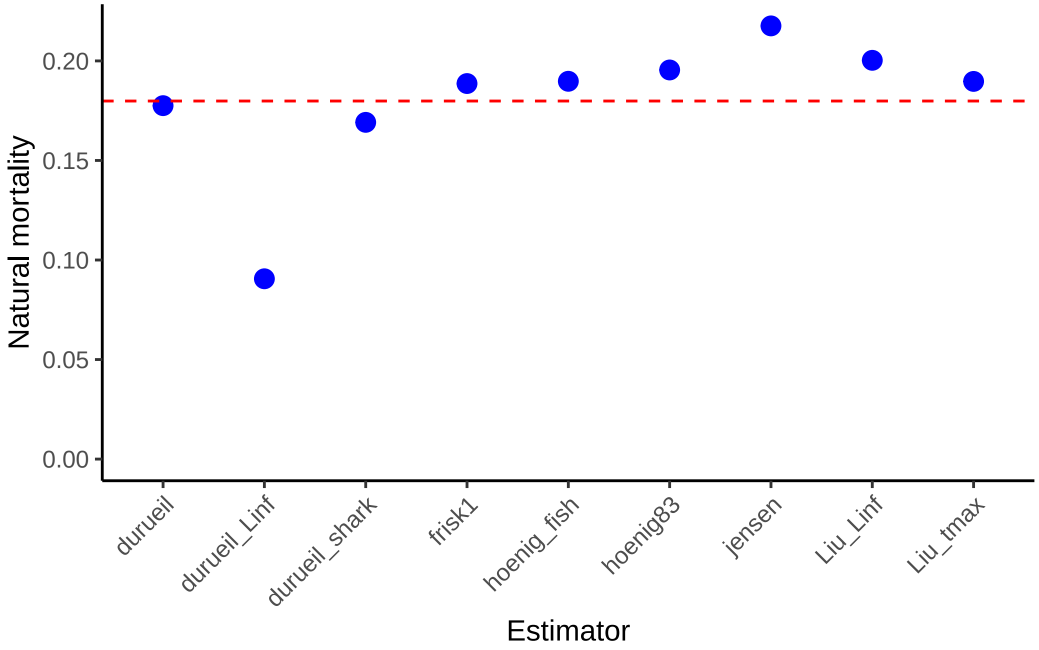
Note that you need the 'caret' and 'lattice' packages as well as the 'metadata234.Rdata' file to run the Liu natural mortality estimators. If not, you do not need these features.

```r
# load in raw life history csv
lh <- read.csv('data/lh_inputs.csv')
head(lh)
```

```
        param     ALV  OCS
1        nsim      10   10
2     len_int       4    4
3  length_type      UF   TL
4 growth_model Schnute VBL0
5   max_length     292  342
6      max_age      22   25
```

Now, we rearrange this data and estimate natural mortality taking the mean of three different estimators. A function is provided herein to explore a range of commonly available natural mortality functions.

```r
# Rearrange raw data and estimate M
lh_df <- lh |>
  pivot_longer(cols = -param, names_to = "spp",values_to = "value") |>
  pivot_wider(names_from = param ,values_from = value) |>
  type.convert(as.is=T) |>
  rowwise() |>
  mutate(M_normal = easiM_estimate(funcs = c('hoenig_fish', 'durueil_shark', 'Liu_tmax'),
         nsim = nsims) |>
  ungroup()

# Explore all M estimates for ALV
easiM_estimator_all(funcs = c("hoenig83", "Liu_tmax", "Liu_Linf",
                           "jensen", "pauly_sst","pauly_kt", "durueil",
                           "durueil_shark", "hoenig_fish", "durueil_Linf",
                           "frisk1","Then_tmax", "Then_Linf"),
                    CV = 0.1, tmax = 22, K = 0.136, L0 = 81, Linf = 245.8,
                    plot = T)
```



```
       func_name          M
1       hoenig83 0.19545455
2       Liu_tmax 0.18971364
3       Liu_Linf 0.20030704
4         jensen 0.21760000
5        durueil 0.17753348
6  durueil_shark 0.16915105
7    hoenig_fish 0.18976802
8   durueil_Linf 0.09056963
9         frisk1 0.18863485
10          mean 0.17985914
```

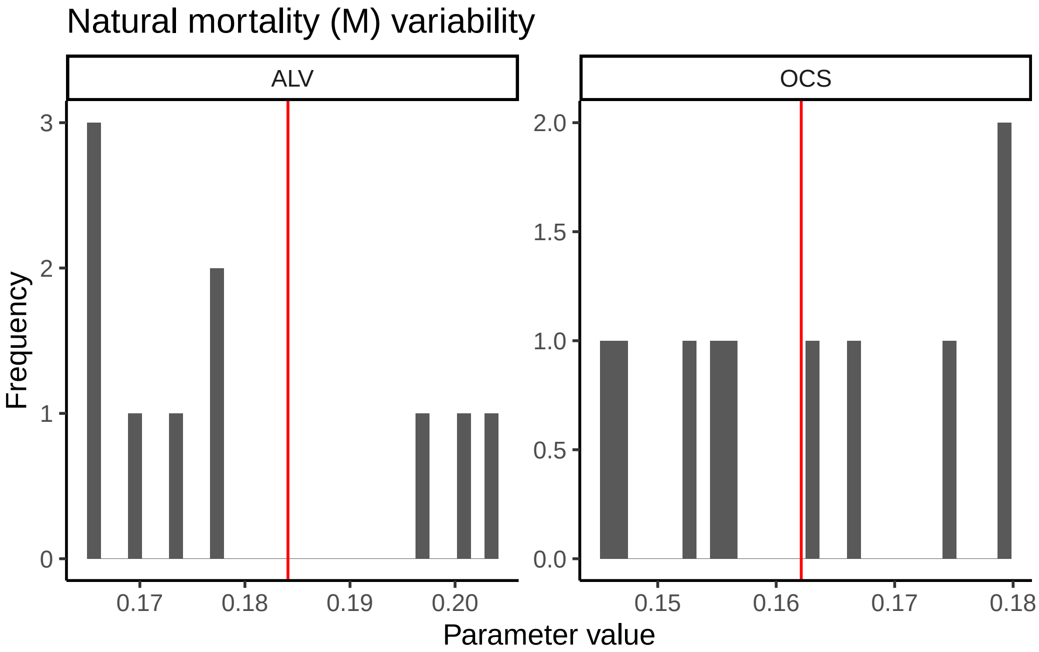
We then convert this data frame into a list by species, and vary parameters as in Hill et al. (in prep) by simulation.

The prep_life_history function takes in data in the format of the dataframe above and converts it into a list by species, and within each list element (df) expands the data by simulation and length from 0-Linf. For each simulation, you can vary parameters using the input csv values defined by using column names in the param_settings argument, or you can define them custom as below. They will default to fixed if nothing is defined. This structure makes it easy to apply EASI-Fish to many species. You can see in the snippet of the dataframe below that sim is 1 and length starts at 4 (0-4cm bin) and goes up in 4cm intervals.

```r
lh_list <- prep_life_history(lh_df,
                             param_settings = list(K = list(dist = "lognormal", cv = 0.1),
                             schnute_a = list(dist = "lognormal", cv = 0.05),
                             lwa = list(dist = "lognormal", cv = 0.05),
                             M_normal = list(dist = "uniform",min = function(x) x - 0.02,
                             L50 = list(dist = "normal", sd = 2),
                             Lm = list(dist = "normal", sd = 2)))
head(lh_list[['ALV']])
```

```
# A tibble: 6 x 31
  spp     sim length mn_length max_age  nsim len_int length_type growth_model
  <chr> <int>  <dbl>     <dbl>   <int> <dbl>   <int> <chr>       <chr>
1 ALV       1      4         2      22    10       4 UF          Schnute
2 ALV       1      8         6      22    10       4 UF          Schnute
3 ALV       1     12        10      22    10       4 UF          Schnute
4 ALV       1     16        14      22    10       4 UF          Schnute
5 ALV       1     20        18      22    10       4 UF          Schnute
6 ALV       1     24        22      22    10       4 UF          Schnute
# i 22 more variables: max_length <int>, Linf <dbl>, K <dbl>, t0 <dbl>,
#   lwa <dbl>, lwb <dbl>, Lm <dbl>, L50 <dbl>, r <dbl>, L0 <dbl>, M <dbl>,
#   min_depth <int>, max_depth <int>, schnute_len1 <dbl>, schnute_len2 <lgl>,
#   schnute_age1 <dbl>, schnute_age2 <lgl>, schnute_a <dbl>, schnute_b <lgl>,
#   logistic_alpha <lgl>, logistic_g <dbl>, M_normal <dbl>
```

Here, we can inspect how the parameters have been varied using natural mortality (M) as an example. As a uniform distribution was used and only 10 sims for the purposes of illustration, the estimates show a somewhat patchy distribution.



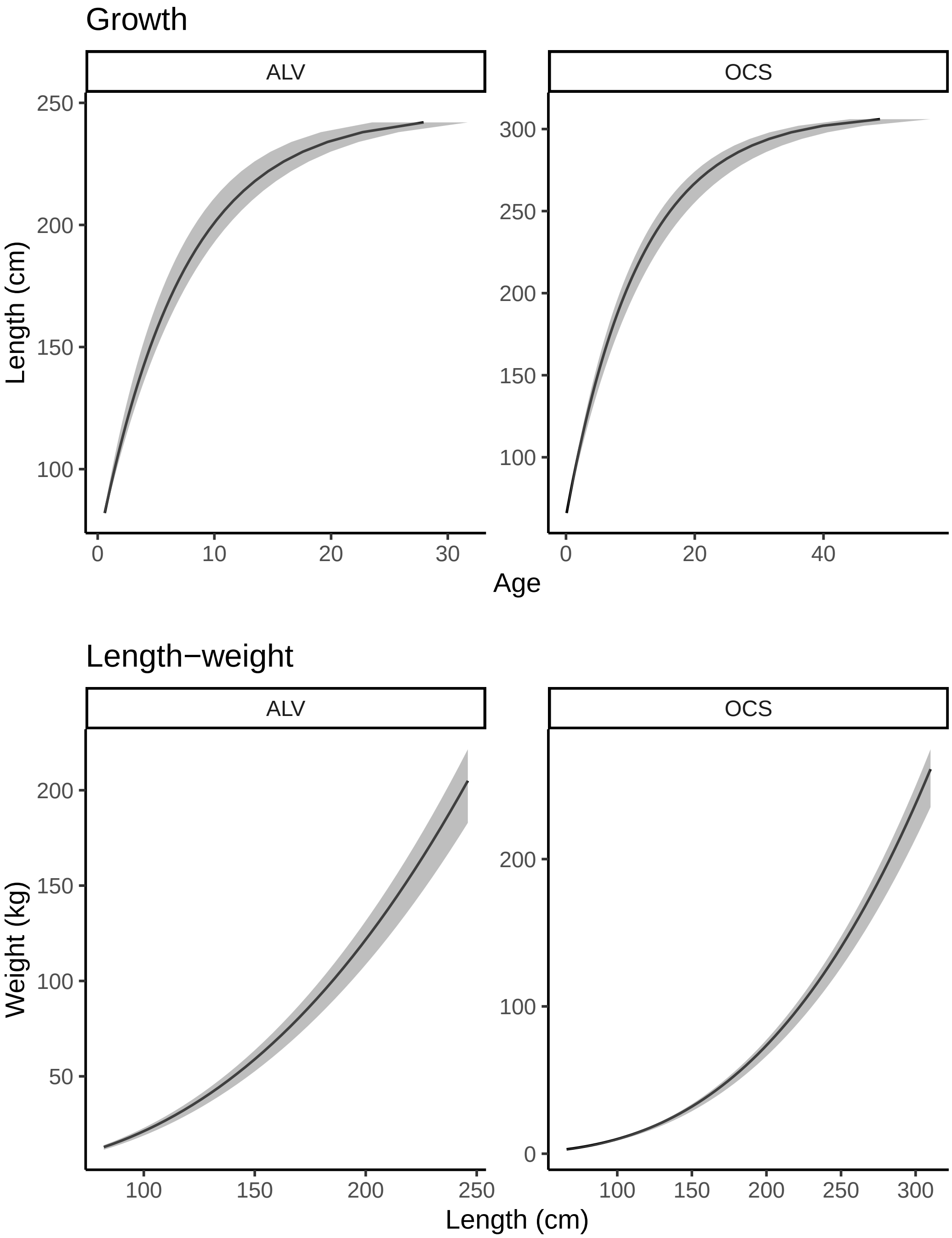
Lastly, now that we have the data in the correct format we can now estimate various processes including growth, maturity, length-weight, and delta-T. We also append selectivity data.

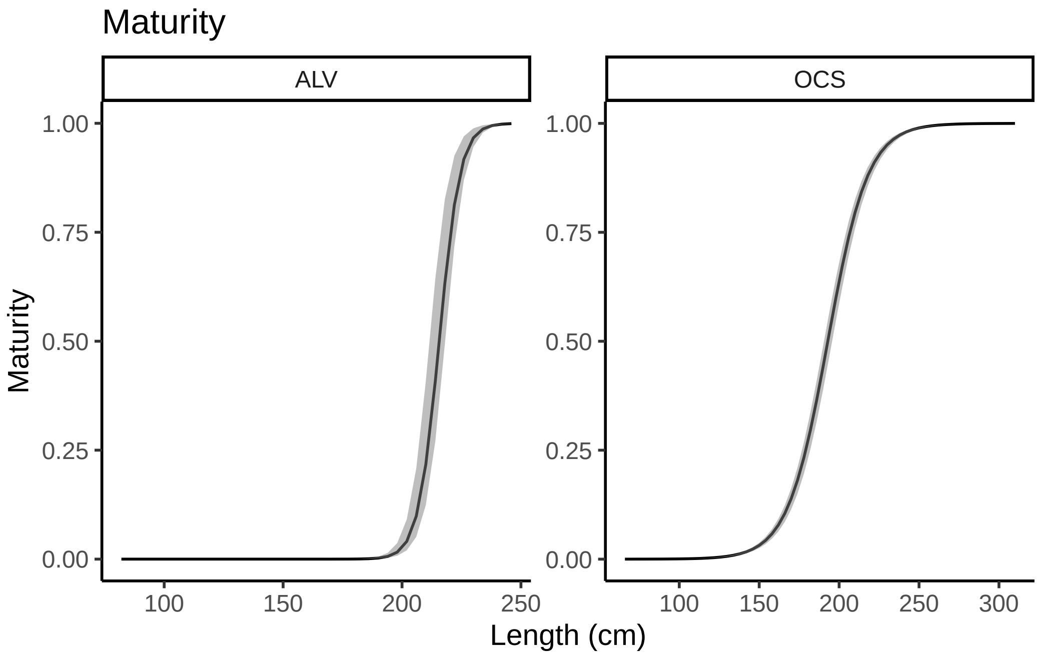
The functions applying each of these life history processes can be found in easi_funcs.R or by running easi_growth2 for example.

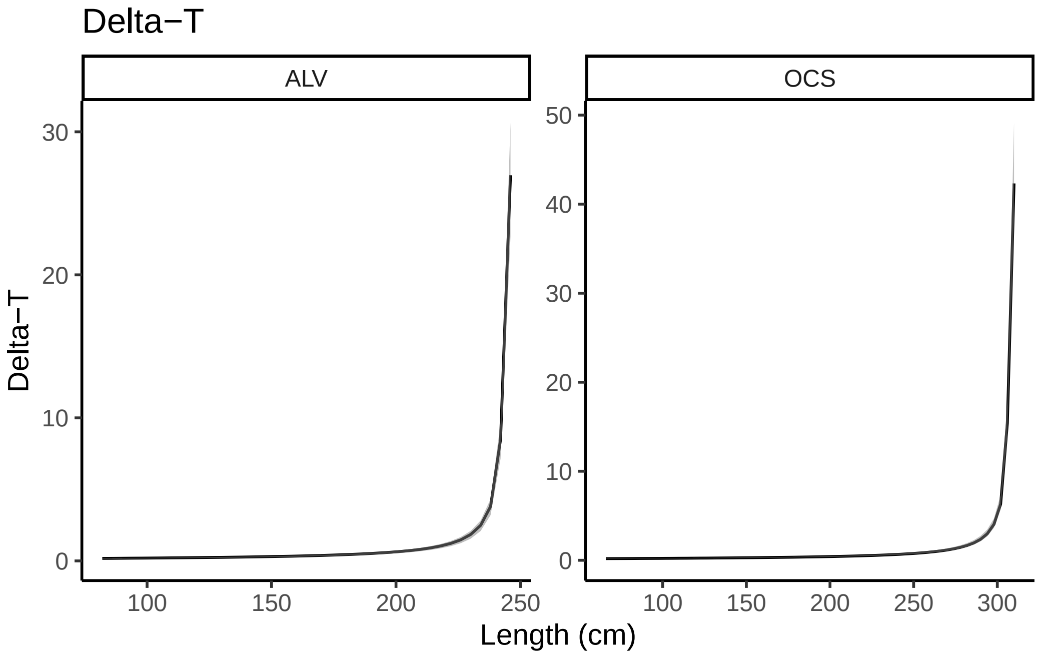
```
# Read in selectivity data from two fleets
sels <- read.csv('data/sels.csv')

# Estimate life history processes
lh_processes <- lh_list |>
  map(~ .x |>
        arrange(spp, sim, mn_length) |>
        mutate(age = easi_growth2(model = growth_model, len = mn_length,
                                  Linf = Linf, K = K, t0 = t0, L0 = L0,
                                  age1 = schnute_age1, age2 = schnute_age2,
                                  len1 = schnute_len1, len2 = schnute_len2,
                                  a = schnute_a, b = schnute_b,
                                  alpha = logistic_alpha, g = logistic_g),
        wt = lw(lwa, mn_length, lwb),
        mat = easi_mat2(len = mn_length, L50 = L50, r = r, Lm = Lm),
        dT = deltaT2(K = K, Linf = Linf, len = length, len_int = len_int))  |>
        dplyr::select(spp, sim, length, mn_length, age, wt, mat, dT,
          M_normal, L0, Linf, length_type, growth_model,
          min_depth, max_depth, max_age) |>
                left_join(sels, by = c('spp', 'mn_length')) |>
        pivot_longer(cols = fshrys,
    names_to = "fishery", values_to = "sel") |>
        dplyr::filter(mn_length >= 0 & length <= (Linf + 4)) |>
      dplyr::select(spp, sim, fishery, everything()))
head(lh_processes[['ALV']])
```

```
# A tibble: 6 x 18
  spp     sim fishery    length mn_length   age       wt      mat    dT M_normal
  <chr> <int> <chr>       <dbl>     <dbl> <dbl>    <dbl>    <dbl> <dbl>    <dbl>
1 ALV       1 LL_shallow      4         2 -1.90 0.000995 4.90e-23 0.100    0.169
2 ALV       1 PS_assoc        4         2 -1.90 0.000995 4.90e-23 0.100    0.169
3 ALV       1 LL_shallow      8         6 -1.80 0.0158   1.28e-22 0.102    0.169
4 ALV       1 PS_assoc        8         6 -1.80 0.0158   1.28e-22 0.102    0.169
5 ALV       1 LL_shallow     12        10 -1.70 0.0573   3.37e-22 0.104    0.169
6 ALV       1 PS_assoc       12        10 -1.70 0.0573   3.37e-22 0.104    0.169
# i 8 more variables: L0 <dbl>, Linf <dbl>, length_type <chr>,
#   growth_model <chr>, min_depth <int>, max_depth <int>, max_age <int>,
#   sel <dbl>
```

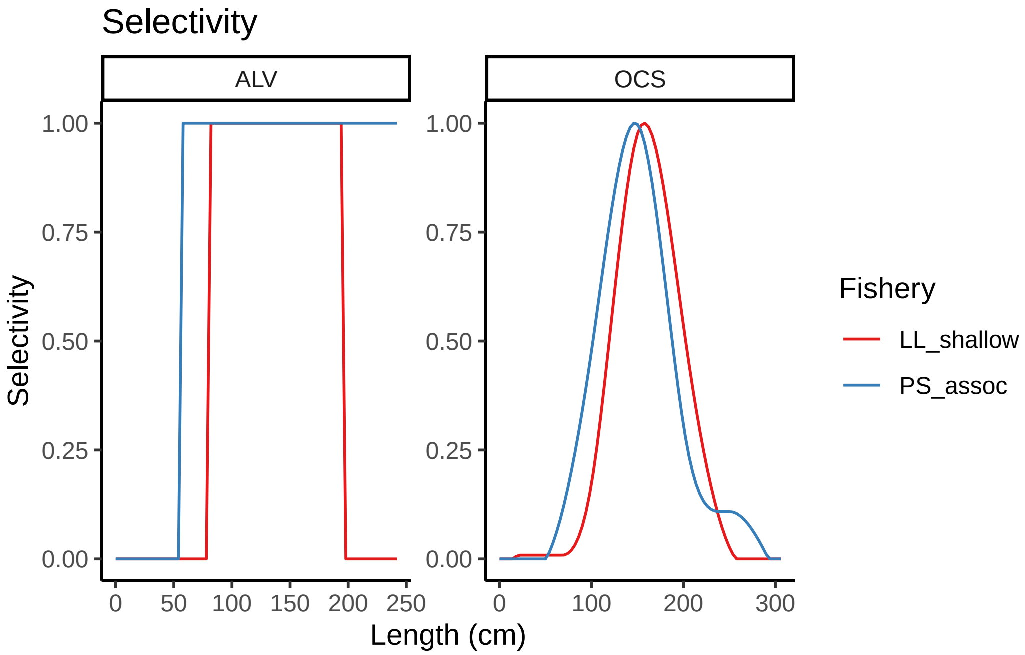
Now we can plot some of these life history processes to make sure they look reasonable.







Lastly, we plot selectivity data for the two fisheries.


Selectivity
ALV
OCS
1.00
0.75
0.50
0.25
0.00
Selectivity
0
50
100
150
200
250
0
100
200
300
Length (cm)
Fishery
LL_shallow
PS_assoc

Preparing fisheries data

Now we read in the fisheries data and prepare it for input into EASI-Fish with a similar pipeline to the life history data above. However, fisheries data is a little more complicated because some parameters are fishery related only, and some are fishery and species specific.

```
# Load in fishery constancts from csv
fshry <- read.csv('data/fishery_inputs.csv')

# Rearrange raw data into df
fshry_df <- expand_grid(spp = spps, nsim = nsims, fshry) |>
  pivot_longer(-c(spp, nsim, param), values_to = 'value', names_to = 'fishery') |>
  pivot_wider(names_from = 'param', values_from = 'value') |>
  dplyr::select(-matches("_(min|max|se|sd|dist|stdD)$")) |>
  dplyr::rename(min_depth_flt = min_depth, max_depth_flt = max_depth) |>
    mutate(across(.cols = -c(spp, fishery),~ as.numeric(.)))
head(fshry_df)
```

```
# A tibble: 4 x 10
  spp    nsim fishery    min_depth_flt max_depth_flt     Q effort season avail
  <chr> <dbl> <chr>                 <dbl>         <dbl> <dbl>  <dbl>  <dbl> <dbl>
1 ALV      10 LL_shallow             0           150     1      1  1         1
2 ALV      10 PS_assoc               0           200     1      1  0.751     1
3 OCS      10 LL_shallow             0           150     1      1  1         1
4 OCS      10 PS_assoc               0           200     1      1  0.751     1
# i 1 more variable: Lc <dbl>
```

Here, we append overlap, at-vessel mortality and post-release mortality values to the data frame by species and fishery. Two different estimates of overlap are provided including the original estimation of overlap using presence/absence at an annual timestep, and the updated approach using the cumulative sum at a monthly timestep. See Hill et al. (in prep) for details on how these were calculated. At-vessel mortality and post-release mortality estimates were extracted from the literature for each species and gear type (i.e. purse seine, longline).

```
# Load in olap, AVM and PRM csv
fshry_spp_df <- read.csv('data/spp_fishery_inputs.csv')

# Join to fishery df
fshry_df <- left_join(fshry_df, fshry_spp_df, by = c('spp', 'fishery'))
```

Here we append the life history dataframe to the fishery dataframe so that we can calculate encounterability which looks at the overlap of fishing gear and species distribution vertically in the water column. To calculate this you need the minimum and maximum depths of the gear, and the species.

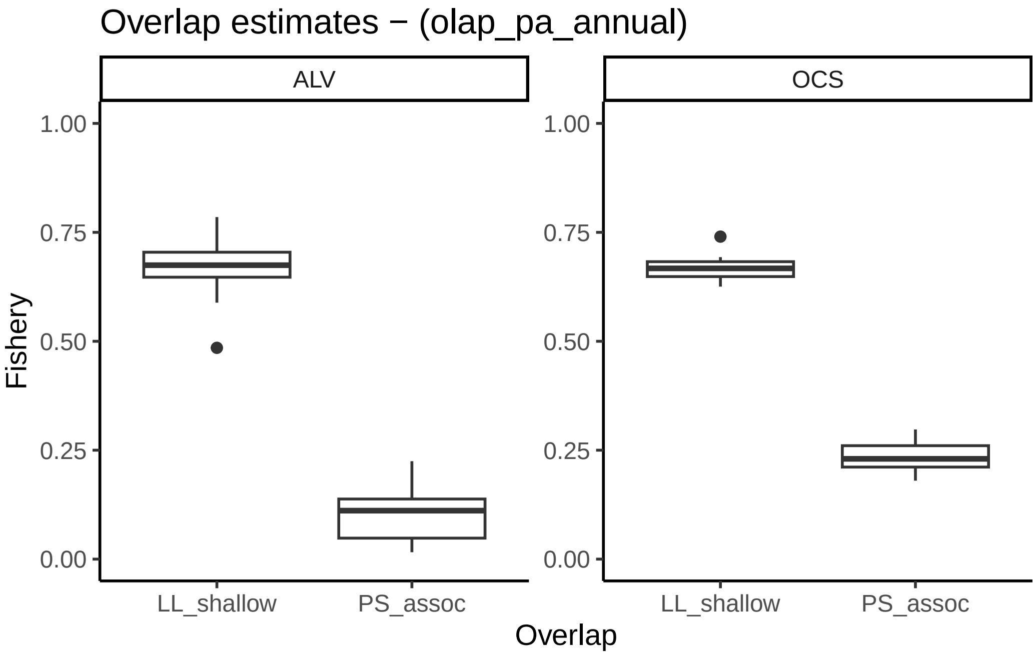
```
spp_depth <- lh_df |>
  dplyr::select(spp, min_depth_spp = min_depth, max_depth_spp = max_depth)

fshry_df <- left_join(fshry_df, spp_depth, by = c('spp'))

fshry_df <- fshry_df |>
  rowwise() |>
  mutate(encount = encount(min_depth_spp, max_depth_spp,
                  as.numeric(min_depth_flt), as.numeric(max_depth_flt))) |>
  dplyr::select(-c(min_depth_spp, max_depth_spp,
                   min_depth_flt, max_depth_flt, Lc))
```

Now, we have compiled all the required fishery parameters, we can now rearrange this data so that it can be combined with the life history information, and vary any parameters we desire. As in Hill et al. (in prep), we add variability to the overlap parameter. Now the fishery data is in an equivalent list structure as the life history data for easy input into EASI-Fish.

```
# Convert fishery df to list by species like life history data
fshry_list <- prep_fishery_data(fshry_df, param_settings = list(
        olap_pa_annual = list(dist = "truncnorm", sd = 0.05, min=0, max=1),
        olap_cumsum_month = list(dist = "truncnorm", sd = 0.05, min=0, max=1)))
head(fshry_list[['ALV']])
```

```
# A tibble: 6 x 12
    sim spp   fishery        Q effort season avail   AVM   PRM olap_pa_annual
  <int> <chr> <chr>      <dbl>  <dbl>  <dbl> <dbl> <dbl> <dbl>          <dbl>
1     1 ALV   LL_shallow     1      1      1     1  0.64  0.35          0.643
2     2 ALV   LL_shallow     1      1      1     1  0.64  0.35          0.485
3     3 ALV   LL_shallow     1      1      1     1  0.64  0.35          0.589
4     4 ALV   LL_shallow     1      1      1     1  0.64  0.35          0.689
5     5 ALV   LL_shallow     1      1      1     1  0.64  0.35          0.785
6     6 ALV   LL_shallow     1      1      1     1  0.64  0.35          0.660
# i 2 more variables: olap_cumsum_month <dbl>, encount <dbl>
```

Now we can plot some of the fishery data to see what it looks like.


Overlap estimates − (olap_pa_annual)
ALV
OCS
Fishery
1.00
0.75
0.50
0.25
0.00
1.00
0.75
0.50
0.25
0.00
LL_shallow
PS_assoc
LL_shallow
PS_assoc
Overlap

Estimating susceptibility

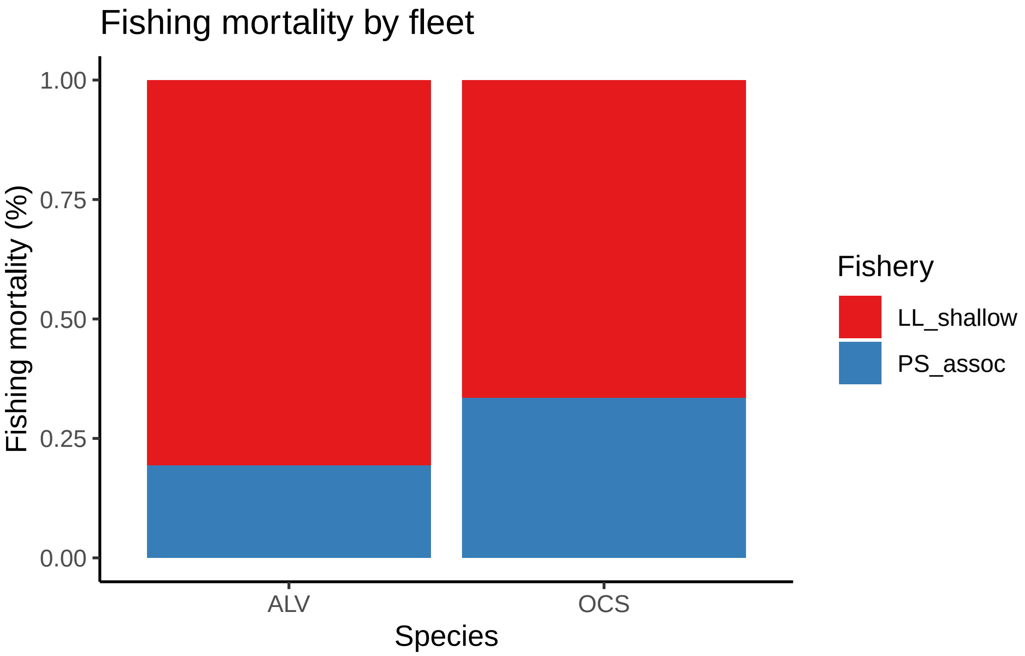
Now that we have prepped all the data we can now apply EASI-Fish. The first thing we do is estimate susceptibility. Here, we combine the life history and fisheries lists into a single dataframe, however these could be saved and read in separately within a loop if running a large number of species or scenarios.

The get_Ffinite() function applies the susceptibility portion of EASI-Fish, both the original and updated. It does this based on whether or not the olap_type is defined as 'annual' or 'month'. This function outputs a dataframe by species, sim, fishery and length with columns including selectivity, Ffinite, Fadj, and Fage.

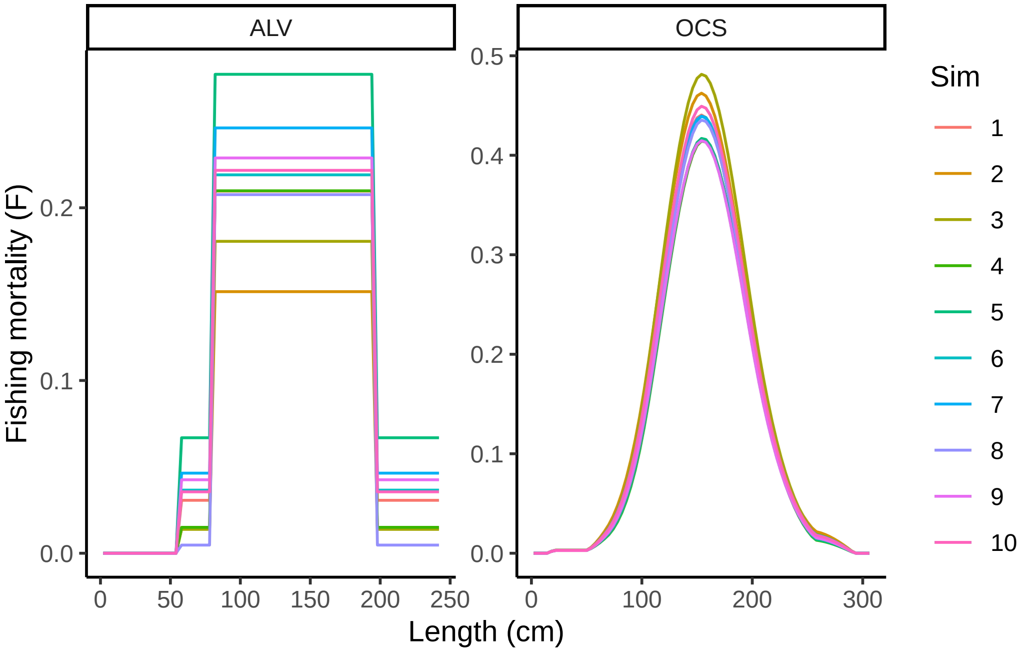
```
# Define olap type ('annual' or 'month')
olap_type <- 'annual'
olap_col <- 'olap_pa_annual'

# Combine data into a single object for ease
input_data <- map2(lh_processes, fshry_list,
  ~ left_join(.x, .y, by = c("spp", "sim", "fishery"))) |>
  bind_rows() |>
  arrange(spp, sim, fishery, mn_length)

# Estimate susceptibility - check column names
Fsusc_df <-
  input_data |>
  mutate(Ffinite = get_Ffinite(olap_type = olap_type, olap = .data[[olap_col]],
                               season = season, avail = avail, encount = encount,
                               sel = sel, avm = AVM, prm = PRM),
         Fadj = (Ffinite * Q) * effort,
         Fage = -log(1 - Fadj)) |>
  dplyr::select(spp, sim, fishery, mn_length, sel, Ffinite, Fadj, Fage)

head(Fsusc_df)
```

```
# A tibble: 6 x 8
  spp     sim fishery    mn_length   sel Ffinite  Fadj  Fage
  <chr> <int> <chr>          <dbl> <dbl>   <dbl> <dbl> <dbl>
1 ALV       1 LL_shallow         2     0       0     0     0
2 ALV       1 LL_shallow         6     0       0     0     0
3 ALV       1 LL_shallow        10     0       0     0     0
4 ALV       1 LL_shallow        14     0       0     0     0
5 ALV       1 LL_shallow        18     0       0     0     0
6 ALV       1 LL_shallow        22     0       0     0     0
```

Here, we collate susceptibility results by fishery and plot.



And by length.



Estimating productivity

Now that we have susceptibility values, we can input these into the yield-per-recruit analysis and spawn-per-recruit to determine productivity values for EASI-Fish. Note that the biomass values output are not independent of the susceptibility values and so only susceptibility values (i.e. fishing mortality) were presented in Hill et al. (in prep). But, we still apply this portion of the analysis for completeness. It also compares the susceptibility results against various reference points which is needed.

The get_Bprod() function takes life history input data as well as the results of the susceptibility portion of EASI-Fish and applies a yield-per-recruit and spawn-per-recruit analysis. It outputs a dataframe by simulation, species, x which represents fishing mortality, and results including ypr, spr, slope and SSBr.

```r
# Combine susceptibility results with lh processes to run ypr and spr
Bprod_ins <- Fsusc_len |>
  left_join(bind_rows(lh_processes),
            by = c("sim", "spp", "mn_length")) |>
  dplyr::select(sim, spp, mn_length, wt, mat, dT, M = M_normal, Fsusc) |>
  group_by(sim, spp) |>
  nest()

# Apply get_Bprod function from easi_funs script
Bprod_df <- get_Bprod3(Bprod_ins, x_vals = seq(0, 10, 0.02))

head(Bprod_df)
```

```
  sim spp    x       ypr      SSBr    slope       spr
1   1 ALV 0.00 0.0000000 11.991536 19.95112 1.0000000
2   1 ALV 0.02 0.3990224 11.205960 18.75092 0.9344891
3   1 ALV 0.04 0.7740409 10.471946 17.62282 0.8732781
4   1 ALV 0.06 1.1264973  9.786102 16.56239 0.8160841
5   1 ALV 0.08 1.4577451  9.145262 15.56548 0.7626431
6   1 ALV 0.10 1.7690547  8.546468 14.62820 0.7127083
```

Estimating summary statistics

Finally, with these results extracted we can estimate fishing mortality and biomass values relative to various reference points. This function does this for each species and simulation and extracts an array of relevant values.

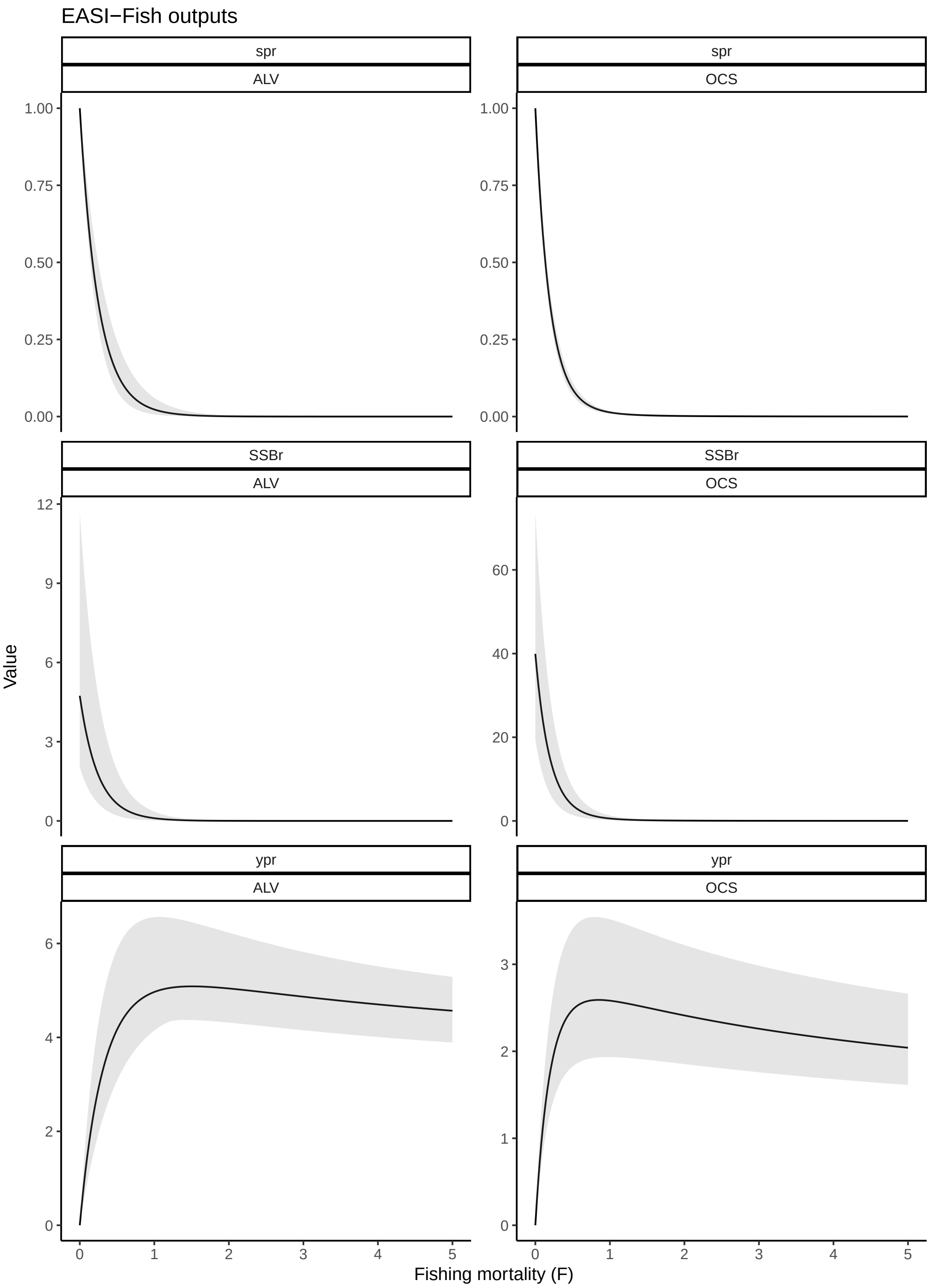
```r
# Extract summary statistics
easi_stats <- get_easi_stats2(Bprod_df, Fsusc_fshry)

head(easi_stats)
```

```
# A tibble: 6 x 4
# Groups:   sim, spp [1]
    sim spp   param value
  <int> <chr> <chr> <dbl>
1     1 ALV   Fest  0.121
2     1 ALV   FMSY  1.32
3     1 ALV   F01   0.72
4     1 ALV   F20   0.48
5     1 ALV   F40   0.28
6     1 ALV   F60   0.16
```

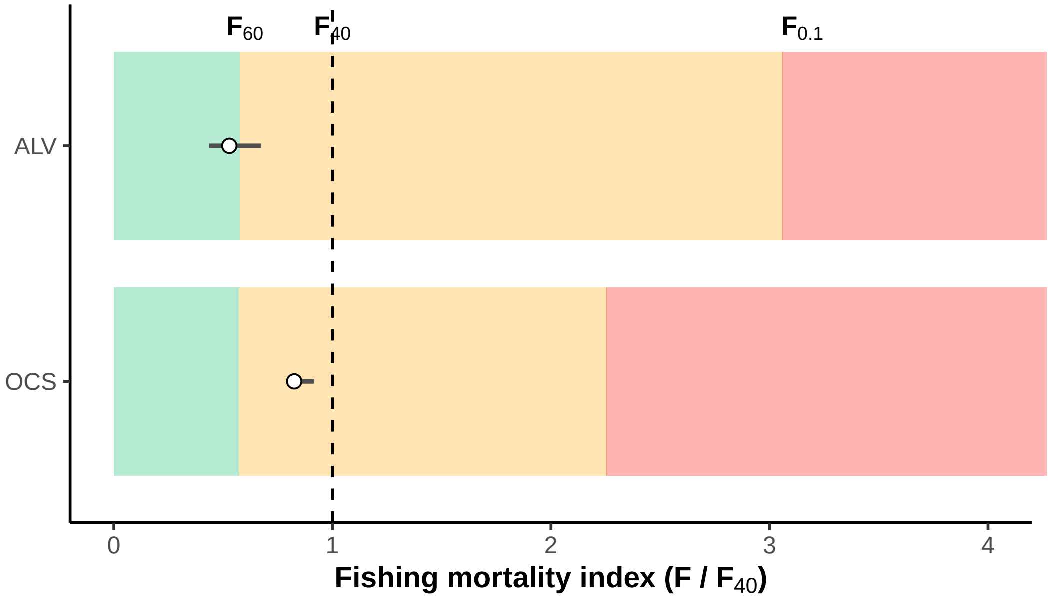
Explore results

Now that we have run the EASI-Fish analysis, we can explore outputs and present the data in a number of different ways. First, we plot yield-per-recruit, spawn-per-recruit and spawning stock biomass by recruit to inspect that outputs are realistic.

Note these results are not identical to those in Hill et al. (in prep) and so should not be compared as such.


EASI−Fish outputs
spr
ALV
spr
OCS
SSBr
ALV
SSBr
OCS
ypr
ALV
ypr
OCS
1.00
0.75
0.50
0.25
0.00
12
9
6
3
0
60
40
20
0
6
4
2
0
3
2
1
0
0
1
2
3
4
5
Value
Fishing mortality (F)

Next we can look at fishing mortality relative to various reference points.



And finally a table with a variety of results.

Parameter	ALV	OCS
F01	0.75 (0.555-1.109)	0.432 (0.385-0.476)
F20	0.42 (0.34-0.582)	0.332 (0.3-0.376)
F40	0.244 (0.2-0.331)	0.192 (0.18-0.216)
F60	0.14 (0.12-0.191)	0.11 (0.1-0.12)
F80	0.068 (0.06-0.096)	0.052 (0.04-0.06)
FF01	0.181 (0.079-0.306)	0.377 (0.335-0.465)
FFMSY	0.088 (0.034-0.162)	0.186 (0.154-0.246)
FMSY	1.618 (1.059-2.535)	0.882 (0.72-1.016)
Fest	0.125 (0.086-0.169)	0.162 (0.149-0.18)
SSB	2.864 (1.201-7.28)	18.549 (8.106-38.558)
SSB01	0.306 (0.084-0.987)	5.07 (1.99-10.344)
SSB20	0.913 (0.389-2.286)	7.576 (3.73-14.286)

Parameter	ALV	OCS
SSB40	1.813 (0.768-4.5)	15.102 (7.055-28.877)
SSB60	2.717 (1.152-6.732)	22.748 (10.974-42.248)
SSB80	3.607 (1.545-8.865)	30.294 (15.405-55.706)
SSBMSY	0.027 (0.001-0.127)	1.037 (0.222-2.658)
SSBSSB40	1.637 (1.249-2.025)	1.209 (1.028-1.402)
SSBSSBMSY	532.759 (57.422-1454.84)	24.216 (12.763-38.536)

References

- Griffiths, S.P., Kesner-Reyes, K., Garilao, C., Duffy, L.M. and Román, M.H., 2019. Ecological Assessment of the Sustainable Impacts of Fisheries (EASI-Fish): a flexible vulnerability assessment approach to quantify the cumulative impacts of fishing in data-limited settings. *Marine Ecology Progress Series*, *625*, pp.89-113.
- Hill, N.J., Allain, V., Griffiths, S., Nicol, S., Potts, J., Phillip Jr, N.B. and Rigby, C.L. (in prep). Assessing the vulnerability of elasmobranchs to western and central Pacific tuna fisheries using an improved ecological risk assessment method. *Fisheries Research.*