

Build Figure 3 and S1

Flexibility is a hidden axis of biomechanical diversity in fishes. J. Exp. Biol.

Eric D. Tytell, Yordano Jimenez, Kelsey Lucas, John Long

February 10, 2023

```
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr  0.3.5
## v tibble  3.1.8      v dplyr  1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(ape)
library(ggtree)

## ggtree v3.6.2 For help: https://yulab-smu.top/treedata-book/
##
## If you use the ggtree package suite in published research, please cite
## the appropriate paper(s):
##
## Guangchuang Yu, David Smith, Huachen Zhu, Yi Guan, Tommy Tsan-Yuk Lam.
## ggtree: an R package for visualization and annotation of phylogenetic
## trees with their covariates and other associated data. Methods in
## Ecology and Evolution. 2017, 8(1):28-36. doi:10.1111/2041-210X.12628
##
## Guangchuang Yu, Tommy Tsan-Yuk Lam, Huachen Zhu, Yi Guan. Two methods
## for mapping and visualizing associated data on phylogeny using ggtree.
## Molecular Biology and Evolution. 2018, 35(12):3041-3043.
## doi:10.1093/molbev/msy194
##
## Guangchuang Yu. Using ggtree to visualize data on tree-like structures.
## Current Protocols in Bioinformatics. 2020, 69:e96. doi:10.1002/cpbi.96
##
## Attaching package: 'ggtree'
##
## The following object is masked from 'package:ape':
##
##     rotate
##
## The following object is masked from 'package:tidyr':
##
##     expand
```

```
library(here)
```

```
## here() starts at /Users/etytel01/Documents/2023/Jimenez JEB stiffness review
```

```
library(patchwork)
```

```
lucasdata <- read_csv(here('stiffness data/Lucas 2014 flexion ratio.csv'))
```

```
## Rows: 662 Columns: 11
```

```
## -- Column specification -----
```

```
## Delimiter: ","
```

```
## chr (3): Organism, Species Name, Group
```

```
## dbl (8): Individual, Measurement, Inflection Point, Length, Ratio, Average R...
```

```
##
```

```
## i Use `spec()` to retrieve the full column specification for this data.
```

```
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
lucasdata <-
```

```
  lucasdata |>
```

```
  rename(organism = Organism,
```

```
         species = `Species Name`,
```

```
         group = Group,
```

```
         indiv = Individual,
```

```
         rep = Measurement,
```

```
         inflection.point = `Inflection Point`,
```

```
         length = Length) |>
```

```
  fill(organism, species, indiv) |>
```

```
  select(organism:length) |>
```

```
  mutate(flexion.ratio = inflection.point / length)
```

```
lucasdata <-
```

```
  lucasdata |>
```

```
  group_by(organism) |>
```

```
  fill(group)
```

```
lucasdatamn <-
```

```
  lucasdata |>
```

```
  group_by(group, organism) |>
```

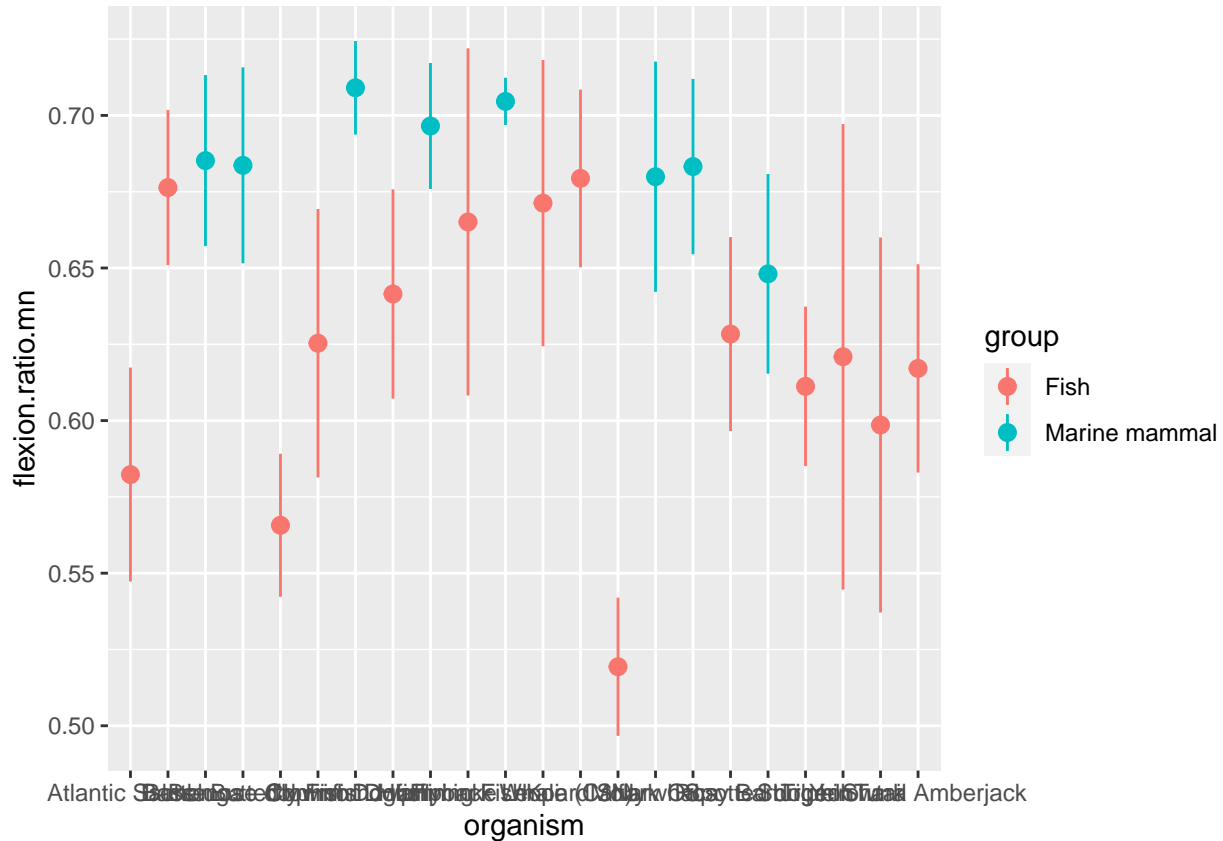
```
  summarize(across(c(length, flexion.ratio), list(mn = mean, sd = sd), .names = '{.col} {.fn}'), .grouped = TRUE)
```

```
lucasdatamn |>
```

```
  filter(group %in% c('Fish', 'Marine mammal')) |>
```

```
  ggplot(aes(x = organism, y = flexion.ratio.mn, color = group)) +
```

```
  geom_pointrange(aes(ymin = flexion.ratio.mn - flexion.ratio.sd, ymax = flexion.ratio.mn + flexion.ratio.sd))
```



```

aleyevdata <- read_csv(here('stiffness data/Aleyev stiffness.csv'))

## Rows: 37 Columns: 30
## -- Column specification -----
## Delimiter: ","
## chr (8): Species, Species.corrected, Species.match, Common name, Family, Or...
## dbl (22): V, L, E1, E2, E3, E4, E5, Q1, Q2, Q3, Q4, Q5, G1, G2, G3, G4, G5, ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
aleyevdata <-
  aleyevdata |>
  rename(Vtotal = V) |>
  pivot_longer(c(E1:E5, Q1:Q5, G1:G5, V1:V5), names_to = c('.value', 'point'), names_pattern = "(.)(.)")
  filter(str_detect(Group, 'fish'))

```

Aleyev reported

$$Q_n = \frac{q_n^{1/2}}{L}$$

where q_n is the cross-sectional area of segment n and L is the body length (to end of vertebral column). The two diameters of the cross-section are k_n perpendicular to the bending axis (the fish's width) and l_n parallel to the bending axis.

He also gave us

$$G_n = \frac{k_n}{l_n}$$

is the ratio of the two axes.

From this, we can figure out the two dimensions and the second moment of area. The area $q_n = \frac{\pi}{4} k_n l_n$. Then

$$G_n Q_n^2 = \left(\frac{k_n}{l_n} \right) \left(\frac{q_n^{1/2}}{L} \right)^2 = \frac{k_n}{l_n} \frac{\pi}{4} \frac{k_n l_n}{L^2} = \frac{\pi}{4 L^2} k_n^2$$

or

$$k_n = \frac{2L}{\pi^{1/2}} G_n^{1/2} Q_n$$

and

$$l_n = \frac{4}{\pi} \frac{L^2 Q_n^2}{k_n} = \frac{2}{\pi^{1/2}} \frac{L Q_n}{G_n^{1/2}}$$

which means that the second moment of area is

$$I_n = \frac{\pi}{4} k_n^3 l_n = \frac{2}{\pi} L^4 G_n Q_n^4$$

```
aleyevdata <-
  aleyevdata |>
  mutate(I.L4 = 2/pi * G * Q^4)
```

```
aleyevdata <-
  aleyevdata |>
  mutate(point = as.numeric(point),
         loc = 0.2 * point)
```

```
aleyevdata |>
  filter(str_detect(Group, 'fish')) |>
  #filter(Common name %in% c('Sprat', 'European eel')) |>
  ggplot(aes(x = loc, y = I.L4, color = Species)) +
  geom_line(show.legend = FALSE) +
  scale_y_log10() +
  facet_wrap(~ Species) #, scales='free')
```



Aleyev's E

Aleyev bent fish through an arc of 45 - 90 deg, then measured the chord length (the straight line distance) of the arc formed along the body for 5 equally spaced sections from head to the end of the vertebral column. The chord length is d_n and it should always be shorter than the arc length $0.2L$, where L is the standard length of the body.

Aleyev reported an index of flexibility E_n , which he defined as

$$E_n = \frac{0.2L - d_n}{\sum_{i=1}^5 (0.2L - d_i)}$$

The value E_n is essentially a curvature, but we cannot solve directly for the curvature κ_n . Given an arc with chord length d_n and arc length $0.2L$, we can solve for d_n as a function of κ_n . The total angle of the arc (2θ) is

$$2\theta = \Delta s \kappa_n$$

where Δs is half the arc length, or $0.1L$. Then

$$d_n = 2r \sin \theta = \frac{2}{\kappa_n} \sin(\Delta s \kappa_n)$$

We cannot solve this analytically for κ_n , but we can solve it numerically. The equation is undefined when $\kappa_n = 0$ (i.e., when the body is straight), but we know that in this case $d_n = 2\Delta s$.

Function to calculate E_n from κ_n

```

E_from_curve <- function(curve, ds2 = 0.1) {
  d <- if_else(curve == 0,
               2*ds2,
               2 / curve * sin(ds2 * curve))
  D <- sum(2*ds2 - d)

  (2*ds2 - d) / D
}

# minimize this function to find the curvature that matches a particular set of E values
minimize_for_kappa <- function(curve, E, ds2 = 0.1) {
  E_from_curve(curve) - E
}

```

Solve for curvature

```

# starting guess for curvature: linearly increasing along the body, with a magnitude slightly larger than
curve0 <- c(1, 2, 3, 4, 5)

aleyevcurve <-
  aleyevdata |>
  filter(str_detect(Group, 'fish')) |>
  group_by(Species) |>
  select(Species, loc, E) |>
  nest(data = c(loc, E)) |>
  mutate(nlfit = map(data, ~ nleqslv::nleqslv(curve0, minimize_for_kappa, jac=NULL, .x$E,
                                           control = list(trace=0))),
         curve = map(nlfit, ~ .$x),
         termcd = map(nlfit, ~ .$termcd)) |>
  select(-nlfit) |>
  unnest(c(data, curve, termcd))

```

Sometimes the default algorithm doesn't converge. These are the species for which we had problems.

```

aleyevcurve |>
  filter(termcd != 1) |>
  distinct(Species)

```

```

## # A tibble: 22 x 1
## # Groups:   Species [22]
##   Species
##   <chr>
## 1 Clupeonella delicatula delicatula
## 2 Alosa kessleri pontica
## 3 Belone belone euxini
## 4 Spicara smaris
## 5 Sciaena umbra
## 6 Squalus acanthias
## 7 Pomatomus saltatrix
## 8 Sarda sarda
## 9 Tachurus mediterraneus ponticus
## 10 Lebistes reticulatus
## # ... with 12 more rows

```

Run the solver again, starting with a different starting guess for the curvature. We also use a Newton's method solver and a small maximum step, so that curvature doesn't diverge.

```
curve0 <- c(0.01, 0.02, 0.5, 1, 2)

aleyevcurve2 <-
  aleyevcurve |>
  group_by(Species) |>
  filter(termcd != 1) |>
  select(Species, loc, E) |>
  nest(data = c(loc, E)) |>
  mutate(nlfit = map(data, ~ nleqslv::nleqslv(curve0, minimize_for_kappa, jac=NULL, .x$E,
    method = 'Newton', global = 'cline',
    control = list(trace=0, stepmax = 0.1, maxit = 200))),
    curve = map(nlfit, ~ .$x),
    termcd = map(nlfit, ~ .$termcd)) |>
  select(-nlfit) |>
  unnest(c(data, curve, termcd))
```

Look for non-convergence

```
aleyevcurve2 |>
  filter(termcd != 1) |>
  distinct(Species, .keep_all = TRUE)
```

```
## # A tibble: 1 x 5
## # Groups:   Species [1]
##   Species                loc      E   curve termcd
##   <chr>                <dbl> <dbl>   <dbl> <int>
## 1 Acipenser guldenstadtri colchicus  0.2    0 0.00307     5
```

That seems to have solved the problem.

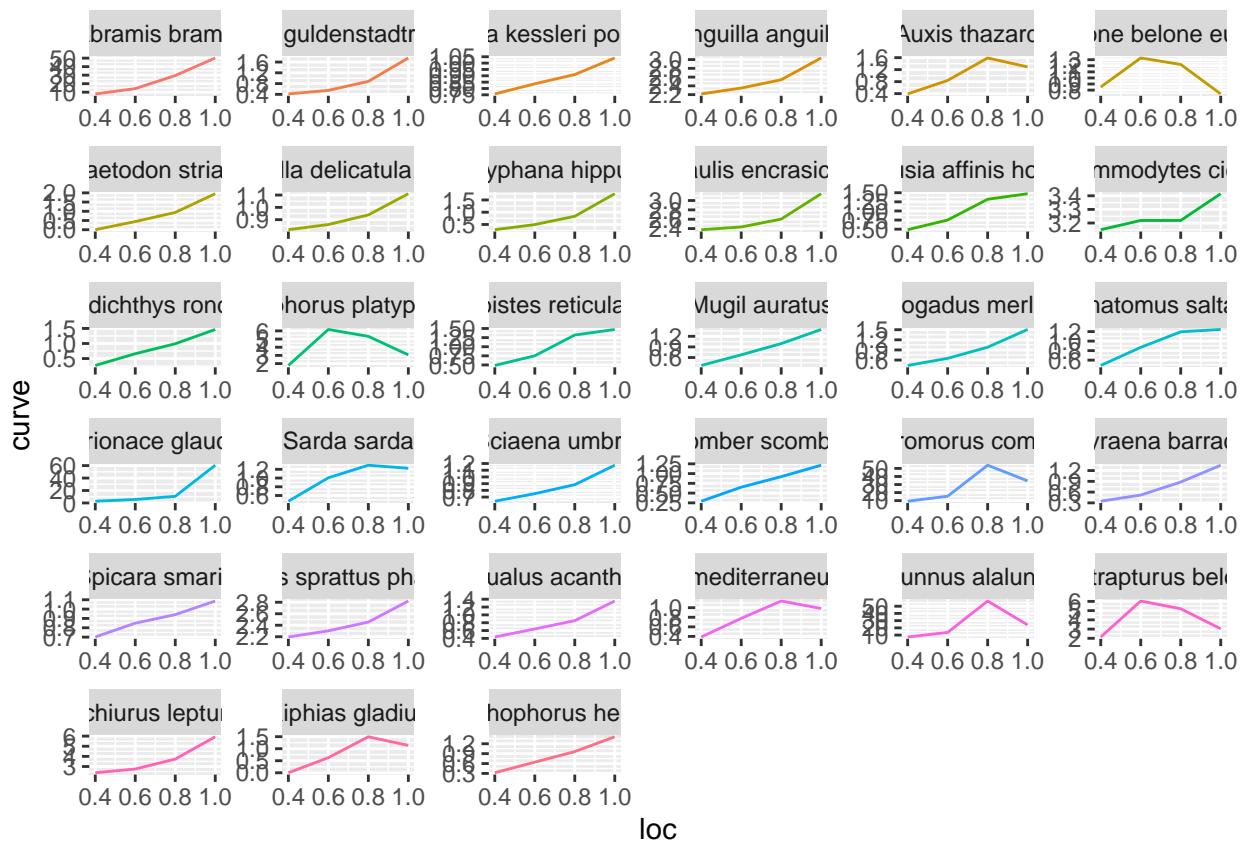
Update the estimated curvature values with the converged solutions

```
aleyevcurve <-
  aleyevcurve |>
  left_join(aleyevcurve2, by = c("Species", "loc", "E")) |>
  mutate(curve = coalesce(curve.y, curve.x),
    termcd = coalesce(termcd.y, termcd.x)) |>
  select(-contains('.x'), -contains('.y'))
```

And merge it in to the main data set.

```
aleyevdata <-
  aleyevdata |>
  left_join(aleyevcurve, by = c("Species", "loc", "E"))

aleyevdata |>
  filter(loc > 0.2) |>
  ggplot(aes(x = loc, y = curve, color = Species)) +
  geom_line(show.legend = FALSE) +
  # scale_y_log10() +
  facet_wrap(~ Species, scales='free')
```



Estimate EI

Now we have an estimate of curvature. The bending moment M is

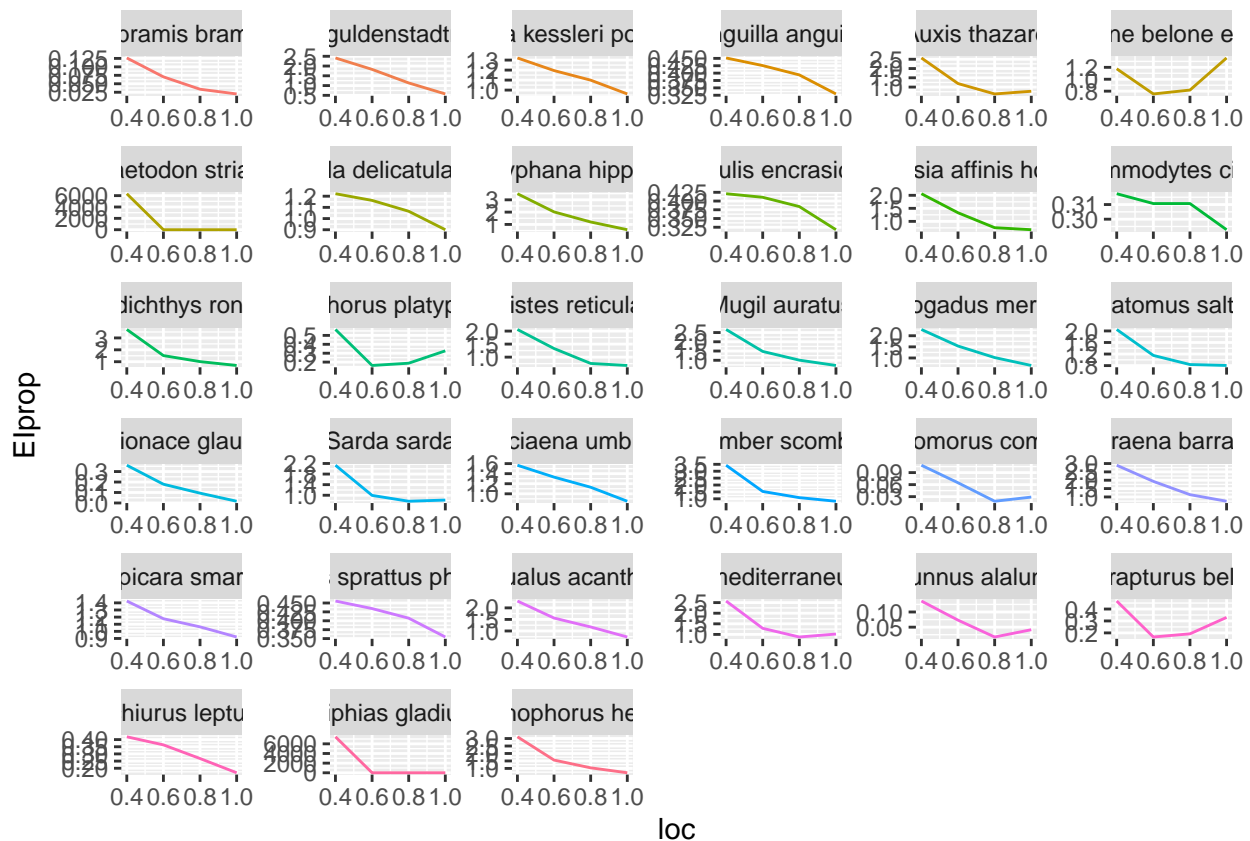
$$M = EI\kappa$$

If we assume that Aleyev used the same weight for each segment, then the bending moment is constant, and EI is proportional to κ^{-1} .

```
aleyevedata <-
  aleyevdata |>
  mutate(EIprop = 1/curve)
```

Hide the very first point, because it tends to be dramatically stiffer than the others, so it obscures the patterns.

```
aleyevedata |>
  filter(loc > 0.2) |>
  ggplot(aes(x = loc, y = EIprop, color = Species)) +
  geom_line(show.legend = FALSE) +
  # scale_y_log10() +
  facet_wrap(~ Species, scales='free')
```

We have a good estimate of I , so we can also estimate a relative Young's modulus.

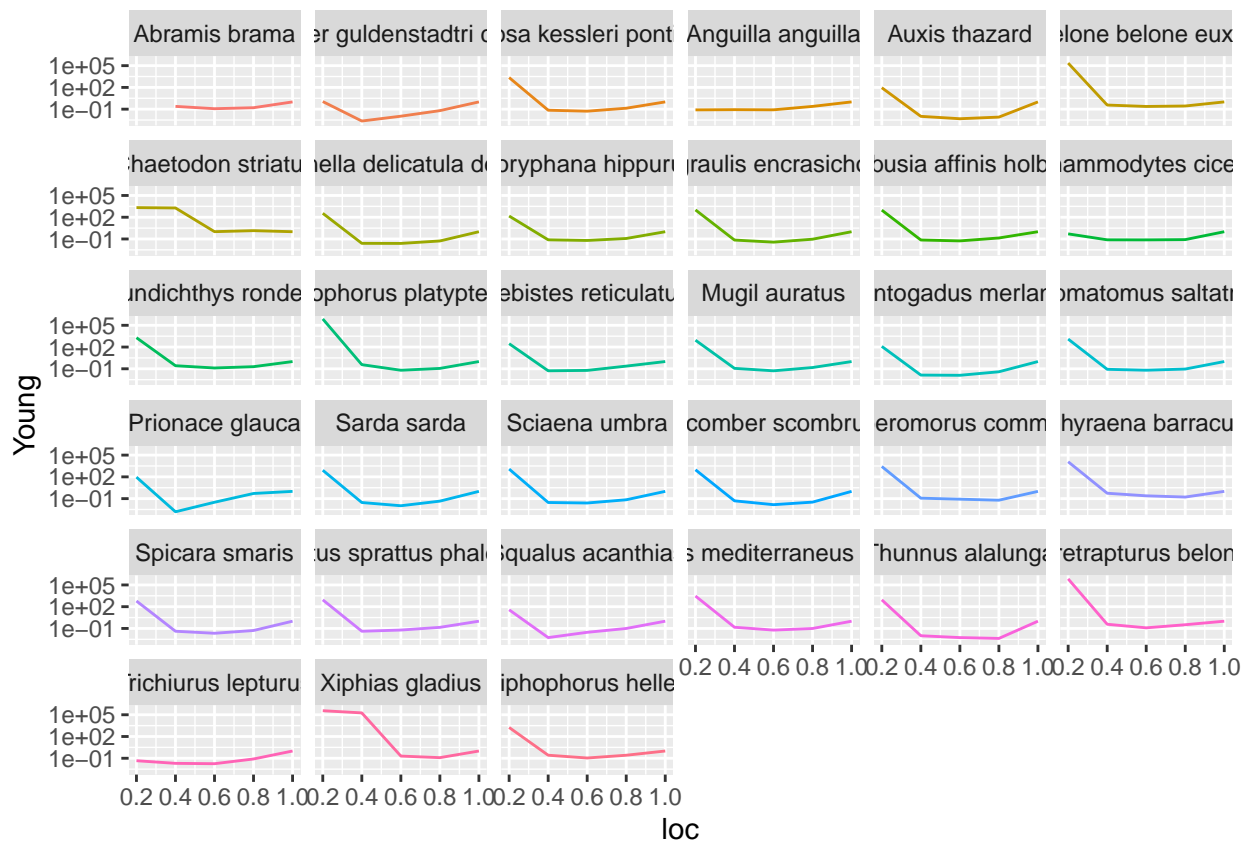
```
aleyevedata <-
  aleyevdata |>
  mutate(Young = Elprop / I.L4)
```

```
aleyevedata |>
  # filter(loc > 0.2) |>
  group_by(Species) |>
  mutate(Youngtail = if_else(point == 5, Young, NA_real_)) |>
  fill(Youngtail, .direction = 'up') |>
  mutate(Young = Young / Youngtail) |>
  ggplot(aes(x = loc, y = Young, color = Species)) +
  geom_line(show.legend = FALSE) +
  scale_y_log10() +
  facet_wrap(~ Species)#, scales='free')
```

```
## Warning in self$trans$transform(x): NaNs produced
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Removed 1 row containing missing values (`geom_line()`).
```



Phylogeny

This is the tree from the Fish Tree of Life: <https://fishtreeoflife.org/>

```
tree <- read.tree(here('actinopt_12k_treePL.tre.xz'))
```

Get the names of species from the tree.

```
allspecies <- tibble(tree$tip.label)
colnames(allspecies) <- c('Species')
head(allspecies)
```

```
## # A tibble: 6 x 1
##   Species
##   <chr>
## 1 Gambusia_marshi
## 2 Gambusia_panuco
## 3 Gambusia_regani
## 4 Gambusia_aurata
## 5 Gambusia_hurtadoi
## 6 Gambusia_gaigei
```

Set up the tip number (just the row)

```
allspecies$Tip <- seq_len(nrow(allspecies))
```

Check to make sure all species match with something in the phylogeny

```

aleyevdata <-
  aleyevdata |>
  filter(Group == 'Bony fish') |>
  mutate(Species.match = coalesce(Species.match, Species.corrected, Species),
         Species.match = str_replace_all(Species.match, ' ', '_')) |>
  left_join(allspecies, by = c("Species.match" = "Species"))

aleyevdata |>
  filter(is.na(Tip)) |>
  distinct(Species.match)

```

```

## # A tibble: 0 x 1
## # ... with 1 variable: Species.match <chr>

```

```

ourspecies <-
  aleyevdata |>
  distinct(Species, .keep_all=TRUE)

```

Prune the tree so that we have just our species

```

verttree <- keep.tip(tree, tip=as.vector(ourspecies$Tip))

```

And load in other taxonomic group names

```

tax <- read_csv(here('PFC_taxonomy.csv.xz')) |>
  separate(genus.species, sep = ' ', into=c('genus2', 'species'),
         extra = 'drop', remove = FALSE) |>
  mutate(across(everything(), ~replace_na(.x, '-')),
         alltaxon = str_c(superclass, class, subclass, infraclass,
                           megacohort, supercohort, cohort, subcohort, infracohort,
                           section, subsection, division, subdivision, series, superorder,
                           sep = '-'),
         alltaxon = str_replace_all(alltaxon, '-+', '-')) |>
  select(alltaxon, order, family, genus, species) |>
  rename(Order = order,
         Family = family,
         Genus = genus,
         Species = species) |>
  mutate(Species.match = str_c(Genus, Species, sep = '_')) |>
  distinct(Genus, Species, .keep_all = TRUE)

```

```

## Rows: 31516 Columns: 21
## -- Column specification -----
## Delimiter: ","
## chr (21): superclass, class, subclass, infraclass, megacohort, supercohort, ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
head(tax)

```

```

## # A tibble: 6 x 6
##   alltaxon                                Order      Family Genus Species Speci~1
##   <chr>                                <chr>      <chr> <chr> <chr>   <chr>
## 1 Actinopterygii-Actinopteri-Chondrostei- Acipense~ Acipe~ Acip~ baerii Acipen~
## 2 Actinopterygii-Actinopteri-Chondrostei- Acipense~ Acipe~ Acip~ brevir~ Acipen~
## 3 Actinopterygii-Actinopteri-Chondrostei- Acipense~ Acipe~ Acip~ dabrya~ Acipen~

```

```
## 4 Actinopterygii-Actinopteri-Chondrostei- Acipense~ Acipe~ Acip~ fulves~ Acipen~
## 5 Actinopterygii-Actinopteri-Chondrostei- Acipense~ Acipe~ Acip~ guelde~ Acipen~
## 6 Actinopterygii-Actinopteri-Chondrostei- Acipense~ Acipe~ Acip~ mediro~ Acipen~
## # ... with abbreviated variable name 1: Species.match
```

```
tax |>
  transmute(perc = str_extract(alltaxon, '-Perc\\w+-')) |>
  filter(!is.na(perc)) |>
  distinct(perc)
```

```
## # A tibble: 2 x 1
##   perc
##   <chr>
## 1 -Percomorphaceae-
## 2 -Percopsaria-
```

```
aleyevdata <-
  aleyevdata |>
  select(-Order, -Family) |>
  left_join(tax |>
    select(-Genus, -Species),
    by = "Species.match")
```

Normalize *EI* by the value at the tail

```
aleyevdata <-
  aleyevdata |>
  mutate(EItail = if_else(point == 5, EIprop, NA_real_),
    Youngtail = if_else(point == 5, Young, NA_real_)) |>
  fill(Youngtail, EItail, .direction = 'up') |>
  mutate(Young2 = Young / Youngtail,
    EI2 = EIprop / EItail)
```

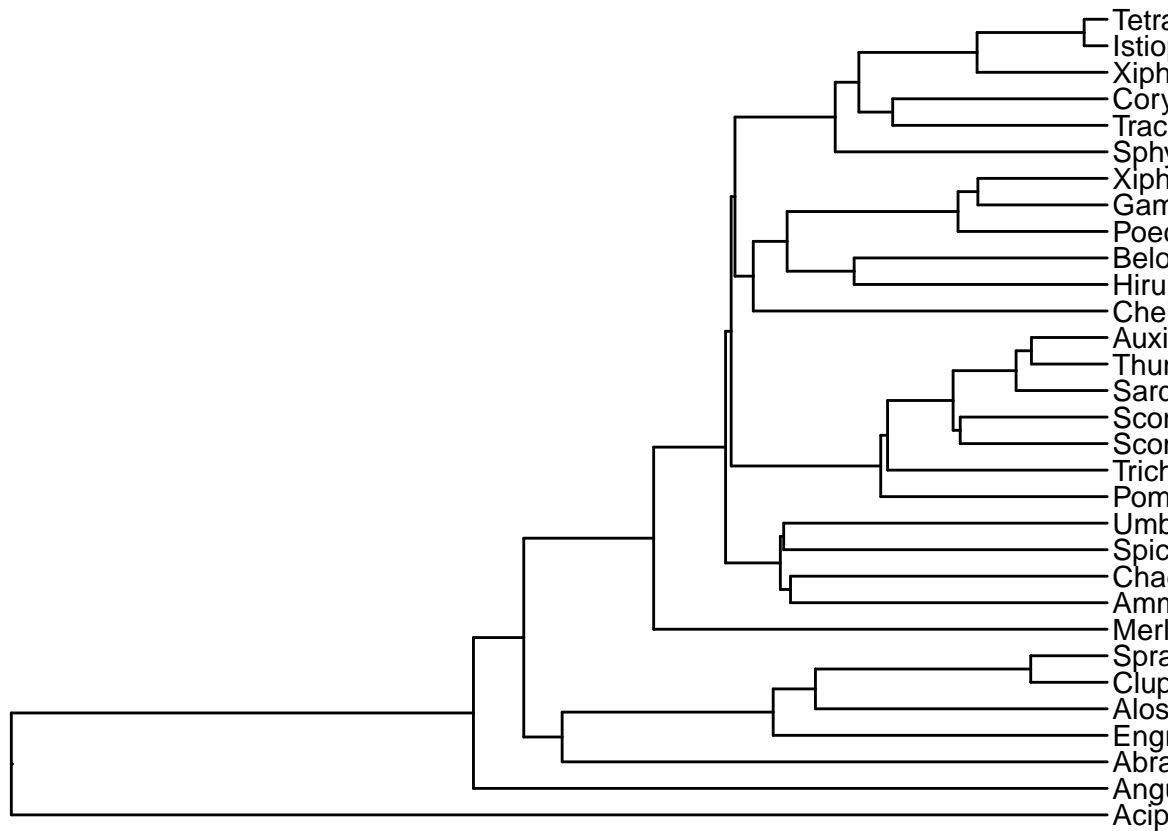
Get the order of species in the tree

```
d <- fortify(verttree)
dd <- subset(d, isTip)
speciesorder <- dd$label[order(dd$y, decreasing = TRUE)]
```

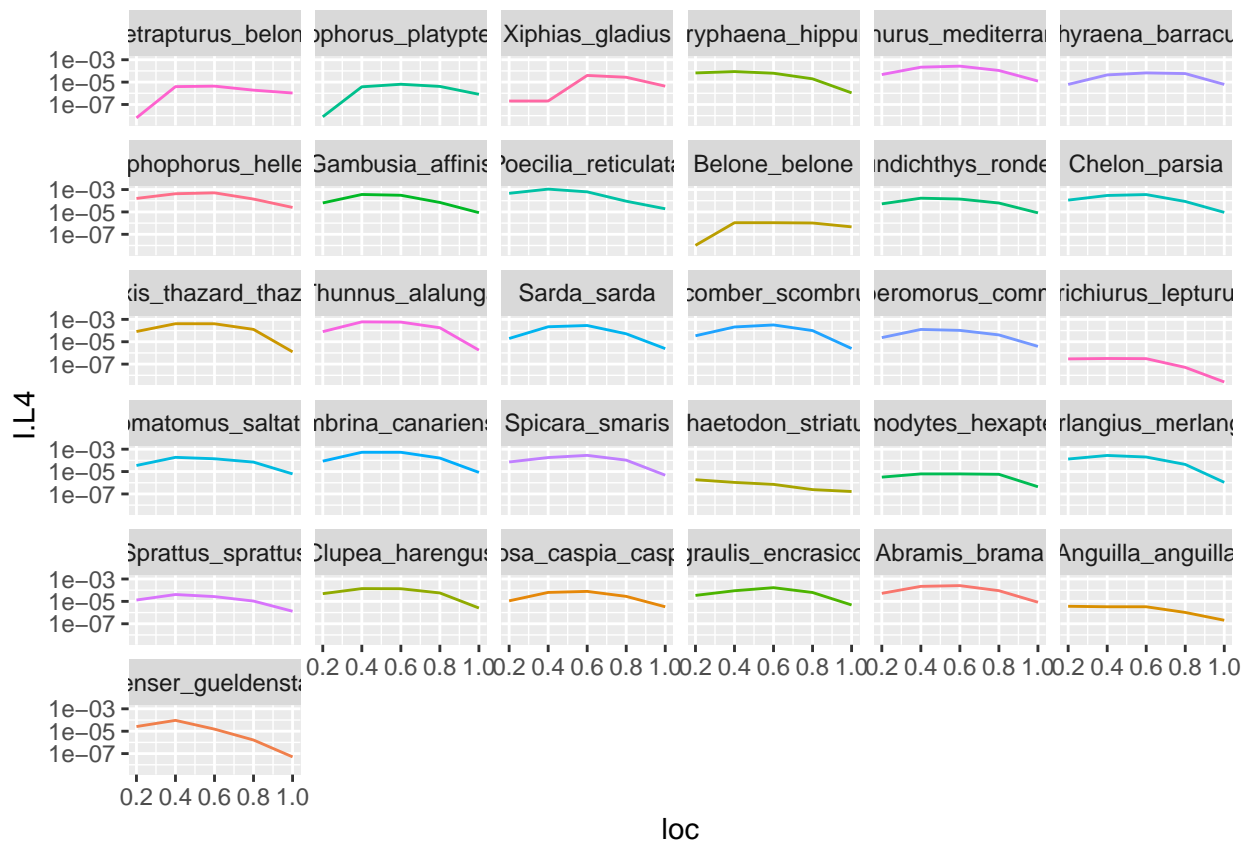
```
speciesorder
```

```
## [1] "Tetrapturus_belone"      "Istiophorus_platypterus"
## [3] "Xiphias_gladus"         "Coryphaena_hippurus"
## [5] "Trachurus_mediterraneus" "Sphyraena_barracuda"
## [7] "Xiphophorus_hellerii"   "Gambusia_affinis"
## [9] "Poecilia_reticulata"    "Belone_belone"
## [11] "Hirundichthys_rondeletii" "Chelon_parsia"
## [13] "Auxis_thazard_thazard"  "Thunnus_alalunga"
## [15] "Sarda_sarda"            "Scomber_scombrus"
## [17] "Scomberomorus_commercion" "Trichiurus_lepturus"
## [19] "Pomatomus_saltatrix"    "Umbrina_canariensis"
## [21] "Spicara_smaris"         "Chaetodon_striatus"
## [23] "Ammodytes_hexapterus"   "Merlangius_merlangus"
## [25] "Sprattus_sprattus"      "Clupea_harengus"
## [27] "Alosa_caspia_caspia"    "Engraulis_encrasicolus"
## [29] "Abramis_brama"          "Anguilla_anguilla"
## [31] "Acipenser_gueldenstaedtii"
```

```
ggtree(verttree) +  
  geom_tiplab()
```



```
aleyevdata |>  
  mutate(Species.match = factor(Species.match, levels = speciesorder)) |>  
  ggplot(aes(x = loc, y = I.L4, color = Species)) +  
  geom_line(show.legend = FALSE) +  
  scale_y_log10() +  
  facet_wrap(~ Species.match) #, scales='free')
```



```

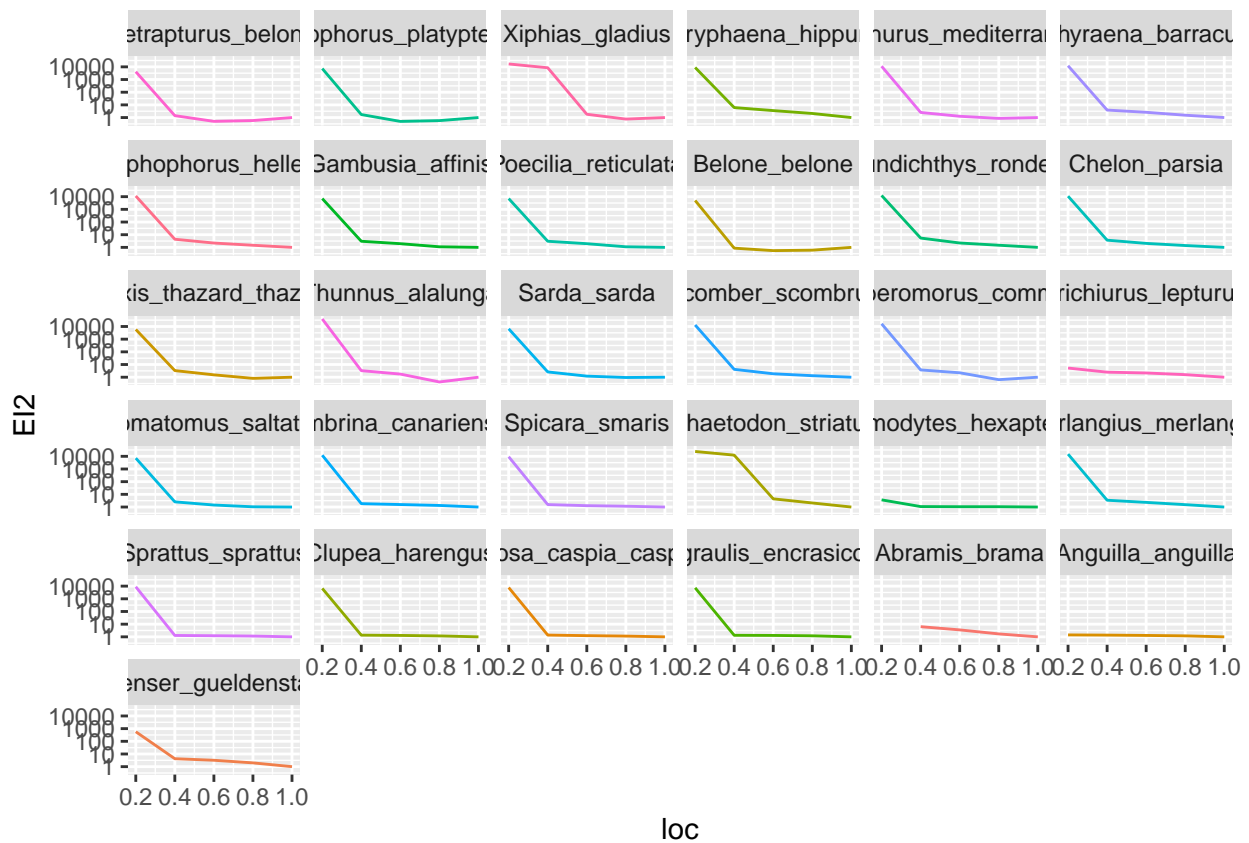
aleyevdata |>
  mutate(Species.match = factor(Species.match, levels = speciesorder)) |>
  ggplot(aes(x = loc, y = EI2, color = Species)) +
  geom_line(show.legend = FALSE) +
  scale_y_log10() +
  facet_wrap(~ Species.match) #, scales='free')

```

```

## Warning in self$trans$transform(x): NaNs produced
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Removed 1 row containing missing values (`geom_line()`).

```



Choose which species to show, trying to highlight different patterns

```
showspecies <- tibble(Species =
  c('Tetrapturus_belone',
    'Xiphias_gladius',
    'Trachurus_mediterraneus',
    'Sphyrhaena_barracuda',
    'Gambusia_affinis',
    'Poecilia_reticulata',
    'Belone_belone',
    # 'Auxis_thazard_thazard',
    # 'Sarda_sarda',
    'Scomber_scombrus',
    'Trichiurus_lepturus',
    'Pomatomus_saltatrix',
    'Spicara_smaris',
    'Chaetodon_striatus',
    'Ammodytes_hexapterus',
    'Merlangius_merlangus',
    'Sprattus_sprattus',
    # 'Alosa_caspia_caspia',
    'Abramis_brama',
    'Anguilla_anguilla',
    'Acipenser_gueldenstaedtii'))

showspecies$Species.num = seq(1, nrow(showspecies))
```

```

aleyevdata <-
  aleyevdata |>
  left_join(showspecies, by = c("Species.match" = "Species"))

```

```

shortspecies <- tibble(verttree$tip.label)
colnames(shortspecies) <- c('Species')
shortspecies$Tip = seq(1, nrow(shortspecies))

```

```

shortspecies |>
  filter(Species %in% pull(showspecies, Species))

```

```

## # A tibble: 18 x 2
##   Species      Tip
##   <chr>      <int>
## 1 Gambusia_affinis      1
## 2 Poecilia_reticulata    3
## 3 Belone_belone         5
## 4 Sphyraena_barracuda    7
## 5 Trachurus_mediterraneus 8
## 6 Tetrapturus_belone    11
## 7 Xiphias_gladus        12
## 8 Scomber_scombrus      14
## 9 Trichiurus_lepturus   18
## 10 Pomatomus_saltatrix   19
## 11 Ammodytes_hexapterus  20
## 12 Chaetodon_striatus    21
## 13 Spicara_smaris        22
## 14 Merlangius_merlangus  24
## 15 Abramis_brama        25
## 16 Sprattus_sprattus     28
## 17 Anguilla_anguilla     30
## 18 Acipenser_gueldenstaedtii 31

```

```

verttreeshort <- keep.tip(verttree, shortspecies |> filter(Species %in% showspecies$Species) |> pull(Tip))

```

```

orders <-
  as_tibble(verttreeshort) |>
  left_join(aleyevdata |>
    distinct(Species, .keep_all = TRUE),
    by = c("label" = "Species.match")) |>
  group_by(Order) |>
  dplyr::summarize(id = min(parent),
    n = n()) |>
  filter(n >= 2 & !str_detect(Order, 'Incertae') & !is.na(Order)) |>
  rename(taxon = Order)
orders

```

```

## # A tibble: 3 x 3
##   taxon      id    n
##   <chr>    <int> <int>
## 1 Cyprinodontiformes    27    2
## 2 Istiophoriformes     30    2
## 3 Scombriformes        31    3

```

```

nodestolabel <- c('Actinopterygii',
  # 'Neopterygii',

```



```

      'Teleostei',
      'Otomorpha',
      # 'Euteleostomorpha',
      'Neoteleostei',
      # 'Acanthomorphata',
      'Percomorphaceae',
      'Eupercaria')

allnodes <-
  left_join(as_tibble(verttreeshort),
            aleyevdata |>
              distinct(Species, .keep_all = TRUE),
              by = c("label" = "Species.match")) |>
  mutate(Species = label,
         label = str_replace(label, '_', ' '),
         alltaxon = replace_na(alltaxon, '-')) |>
  select(parent, node, alltaxon)

labelnodes <- tibble()
for (n in nodestolabel) {
  print(n[[1]])
  labelnodes <-
    allnodes |>
    dplyr::filter(str_detect(alltaxon, n[[1]])) |>
    dplyr::summarize(taxon = n[[1]],
                    # alltaxon = alltaxon[1],
                    id = min(parent),
                    n = n()) |>
    bind_rows(labelnodes)
}

## [1] "Actinopterygii"
## [1] "Teleostei"
## [1] "Otomorpha"
## [1] "Neoteleostei"
## [1] "Percomorphaceae"
## [1] "Eupercaria"

labelnodes <-
  bind_rows(orders, labelnodes)

phyldata <-
  as_tibble(verttreeshort) |>
  # left_join(labelnodes, by = c('node' = 'pmin')) |>
  # select(-nmin) |>
  left_join(aleyevdata |>
            arrange(Species, loc) |>
            group_by(Species.match, Species.num, Species) |>
            mutate(Young = if_else(point == 1, NA_real_, Young),
                   EI3 = if_else(point == 1, NA_real_, EI2)) |>
            summarize(Etail = last(E),
                      EIrng = max(EI3, na.rm = TRUE) - min(EI3, na.rm = TRUE)),
            by = c("label" = "Species.match")) |>
  rename(label2 = label,
         label = Species.num) |>

```

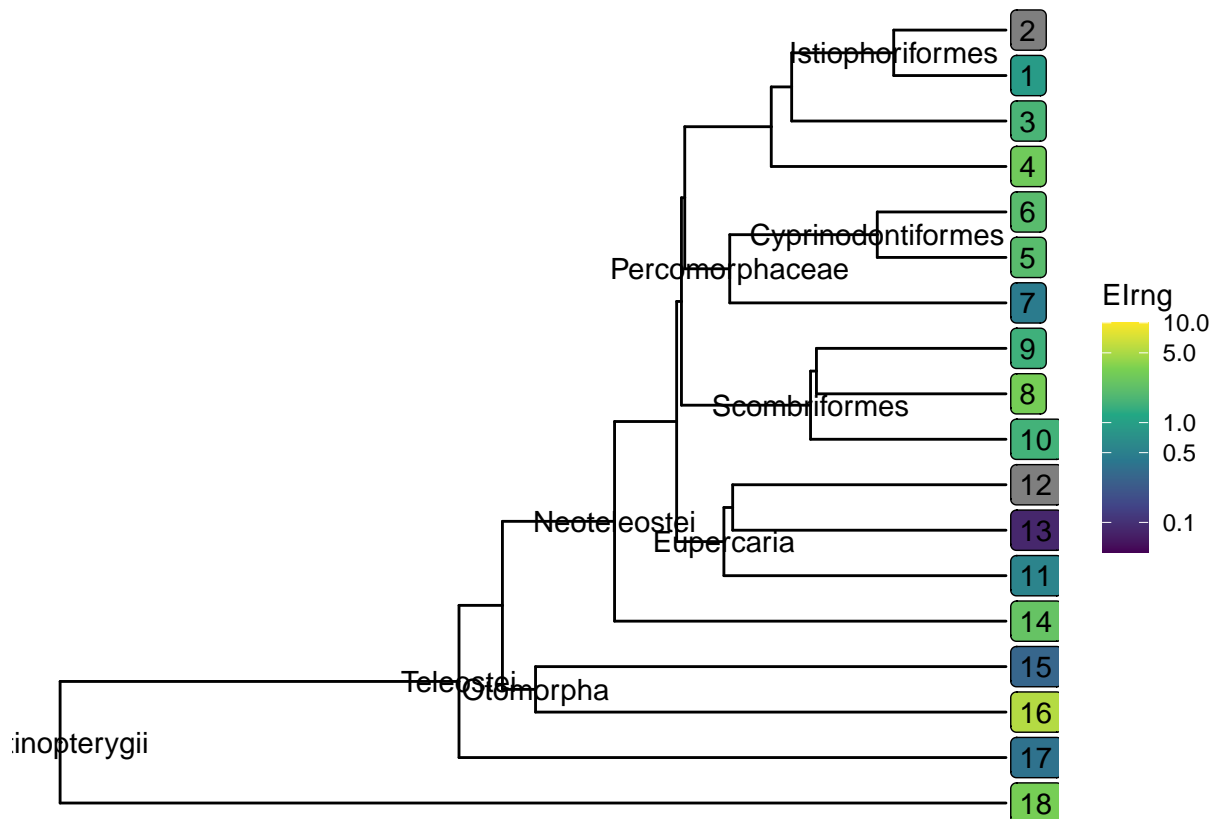
```
mutate(label3 = if_else(is.na(label), label2, NA_character_))
```

`summarise()` has grouped output by 'Species.match', 'Species.num'. You can
override using the `.groups` argument.

```
phylo_plot <-  
  phylodata |>  
  left_join(labelnodes, by = c('node' = 'id')) |>  
  tidytree::as.treedata() |>  
  ggtree() + #branch.length = 'none' +  
  geom_text(aes(label = taxon)) +  
  geom_tiplab(geom = 'label', aes(fill = EInrg)) +  
  scale_fill_continuous(type = 'viridis', trans = 'log', limits = c(0.05, 10),  
                        breaks = c(0.1, 0.5, 1, 5, 10))
```

phylo_plot

Warning: Removed 26 rows containing missing values (`geom_text()`).



```
stiff_plot <-  
  aleyevdata |>  
  filter(Species.match %in% showspecies$Species) |>  
  mutate(Species.match = factor(Species.match, levels = speciesorder),  
         species.num = as.integer(Species.match)) |>  
  mutate(Young2 = if_else(Young2 > 100, 100, Young2)) |>  
  ggplot(aes(x = (loc-0.1)*100, y = Young2, group = Species)) +  
  geom_line() +  
  geom_point(aes(color = Young2)) +
```

```

geom_label(data = ~ distinct(., Species.match, .keep_all = TRUE), aes(label=Species.num),
           x = 10, y = -2.5, hjust = 0, vjust = 0, size = 3) +
geom_text(data = ~ distinct(., Species.match, .keep_all = TRUE), aes(label=Species),
          x = 10, y = 2.1, hjust = 0, vjust = 1, size = 2) +
# geom_text(data = ~ distinct(., Species.match, .keep_all = TRUE), aes(label=`Common name`),
#           x = 10, y = 4, hjust = 0, vjust = 1, size = 3) +
scale_y_log10(breaks = c(0.01, 1, 1e2), labels = c('0.01', '1', '>100')) +
scale_x_continuous(breaks = c(25, 50, 75)) +
scale_color_viridis_c(trans = 'log', limits = c(0.002, 10), direction = -1,
                     breaks = c(0.01, 0.1, 1, 10), na.value = 'black', option = 'plasma') +
facet_wrap(~ Species.match, ncol = 6) +
labs(y = 'Stiffness relative to cadual segment', x = 'Body location (%)') +
theme_light() +
theme(strip.background = element_blank(),
      panel.grid.minor = element_blank())

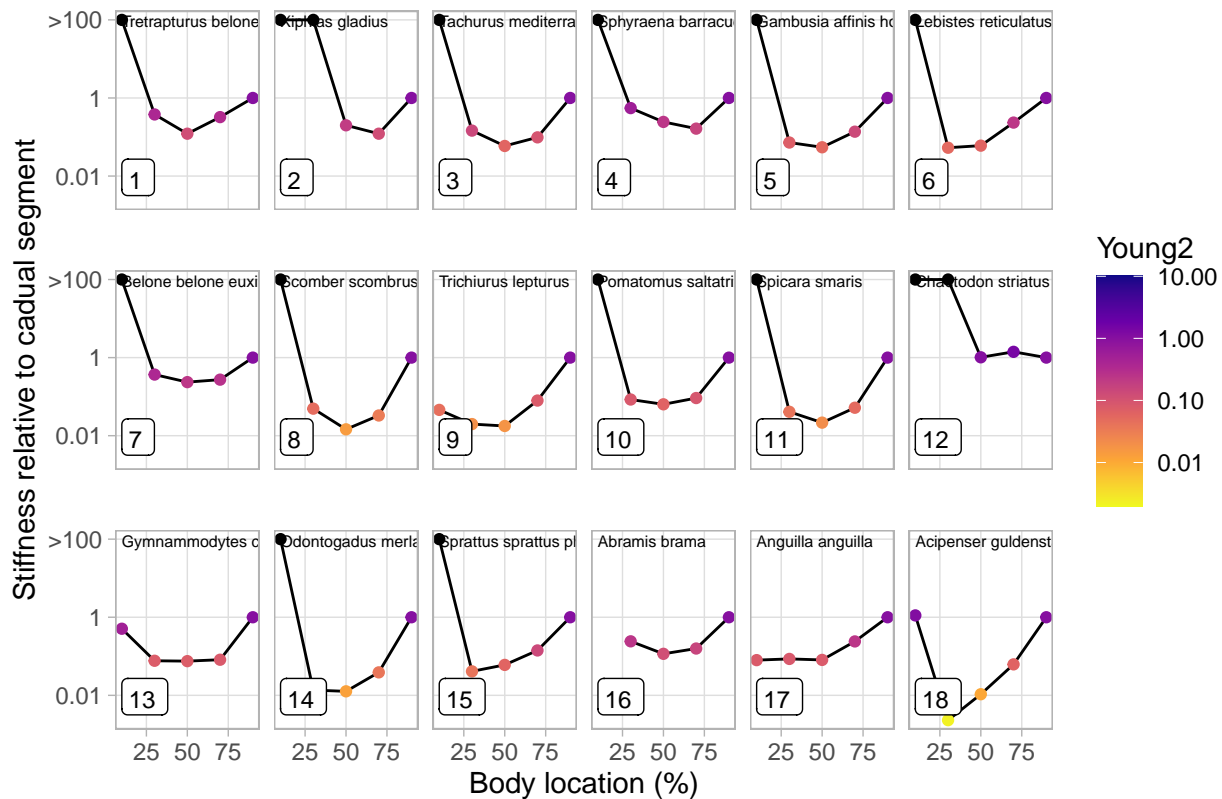
stiff_plot

```

```

## Warning in self$trans$transform(x): NaNs produced
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning in self$trans$transform(x): NaNs produced
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning in self$trans$transform(x): NaNs produced
## Warning: Transformation introduced infinite values in discrete y-axis
## Warning: Removed 1 row containing missing values (`geom_line()`).
## Warning: Removed 1 rows containing missing values (`geom_point()`).

```



```

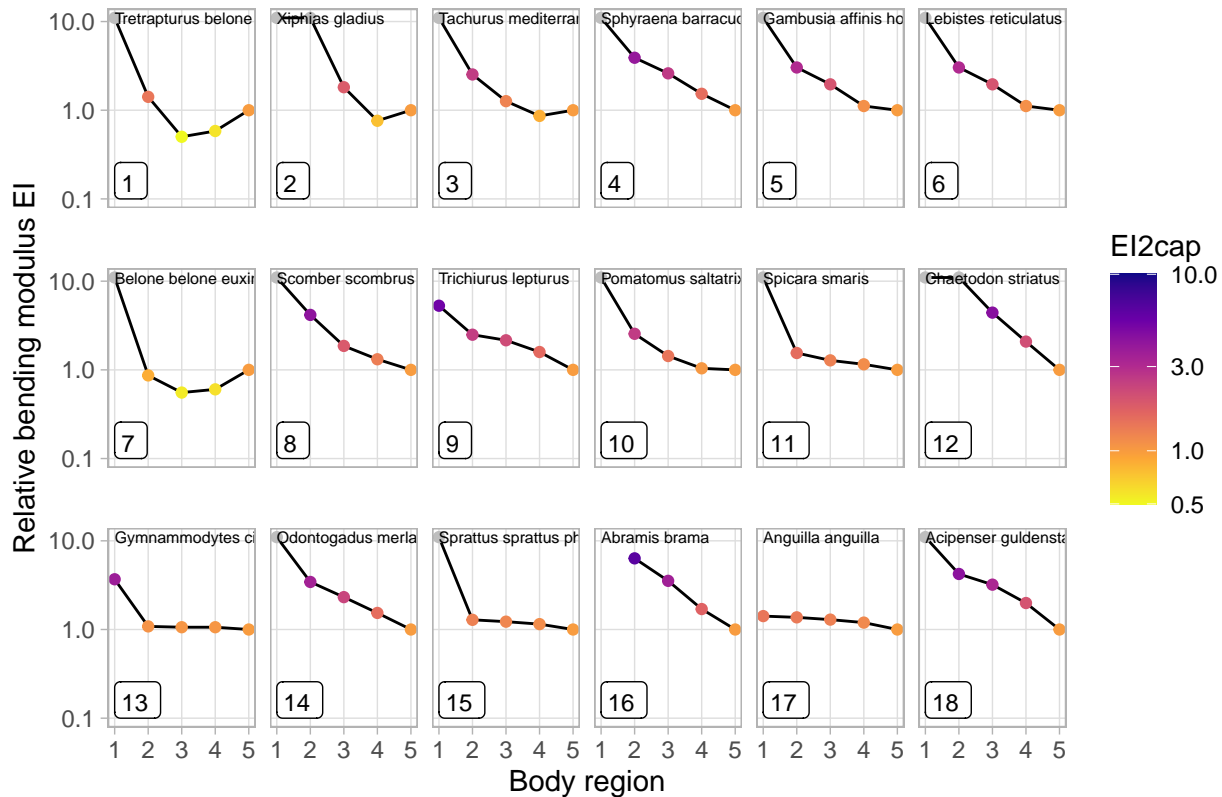
EI_plot <-
  aleyevdata |>
  filter(Species.match %in% showspecies$Species) |>
  mutate(Species.match = factor(Species.match, levels = speciesorder),
         species.num = as.integer(Species.match)) |>
  mutate(EI2cap = if_else(EI2 > 10, 11, EI2)) |>
  ggplot(aes(x = point, y = EI2cap, group = Species)) +
  geom_line() +
  geom_point(aes(color = EI2cap)) +
  geom_label(data = ~ distinct(., Species.match, .keep_all = TRUE), aes(label=Species.num),
            x = 1, y = -1, hjust = 0, vjust = 0, size = 3) +
  geom_text(data = ~ distinct(., Species.match, .keep_all = TRUE), aes(label=Species),
            x = 1, y = 1.1, hjust = 0, vjust = 1, size = 2) +
  # geom_text(data = ~ distinct(., Species.match, .keep_all = TRUE), aes(label='Common name'),
  #           x = 10, y = 4, hjust = 0, vjust = 1, size = 3) +
  scale_y_log10(breaks = c(0.1, 1, 10), limits = c(0.1, 11)) + # (breaks = c(0.01, 1, 1e2), labels = c('
  # scale_x_continuous(breaks = c(25, 50, 75)) +
  scale_color_viridis_c(trans = 'log', limits = c(0.5, 10), direction = -1,
                       breaks = c(0.5, 1, 3, 10), na.value = 'gray', option = 'plasma') +
  facet_wrap(~ Species.match, ncol = 6) +
  labs(y = 'Relative bending modulus EI', x = 'Body region') +
  theme_light() +
  theme(strip.background = element_blank(),
        panel.grid.minor = element_blank())

```

EI_plot

Warning in self\$trans\$transform(x): NaNs produced

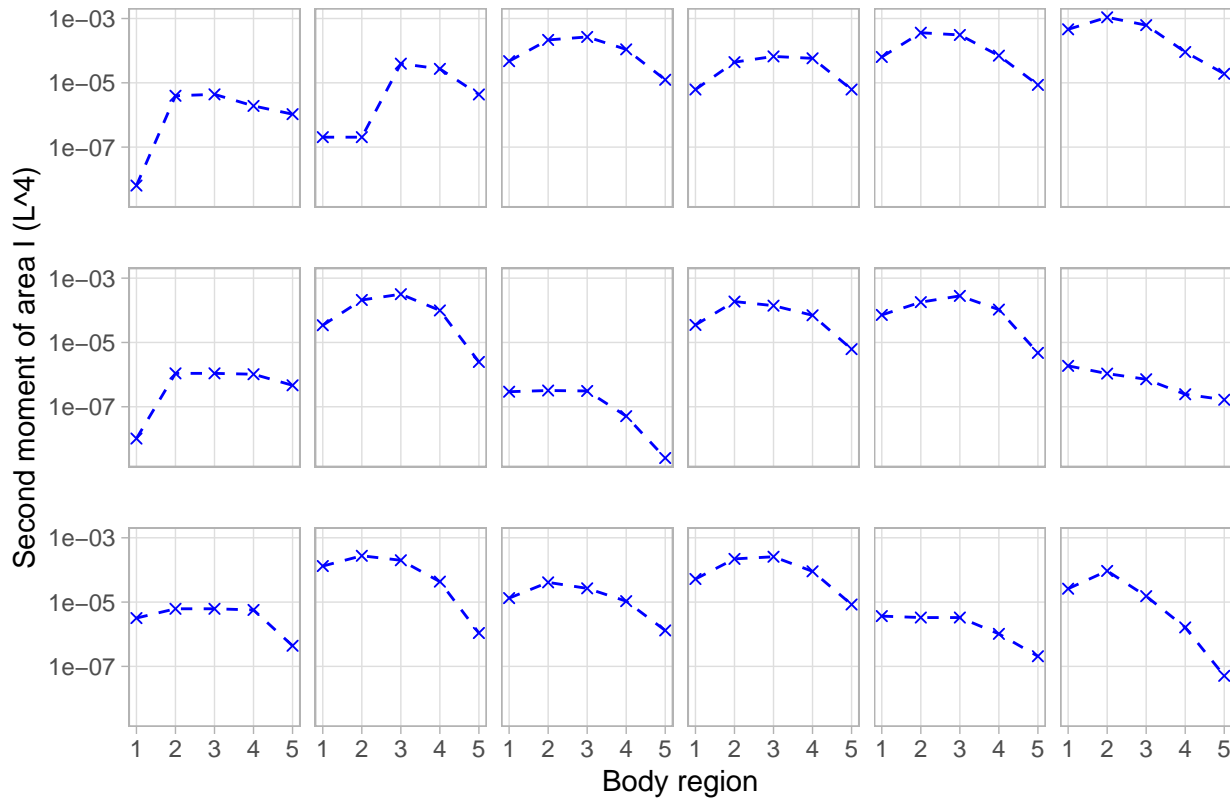
```
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning in self$trans$transform(x): NaNs produced
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning in self$trans$transform(x): NaNs produced
## Warning: Transformation introduced infinite values in discrete y-axis
## Warning: Removed 1 row containing missing values (`geom_line()`).
## Warning: Removed 1 rows containing missing values (`geom_point()`).
```



```
I_plot <-
  aleyevdata |>
  filter(Species.match %in% showspecies$Species) |>
  mutate(Species.match = factor(Species.match, levels = speciesorder)) |>
  # mutate(I.L4.cap = case_when(I.L4 > 1e-3 ~ 1e-3,
  #                             I.L4 < 1e-7 ~ 1e-7,
  #                             TRUE ~ I.L4),
  #        I.outofrange = if_else(between(I.L4, 1e-7, 1e-3), 'in', 'out')) |>
  ggplot(aes(x = point, y = I.L4, group = Species)) +
  geom_line(color = 'blue', linetype = 2) +
  geom_point(shape = 4, color = 'blue') +
  scale_y_log10(breaks = c(1e-7, 1e-5, 1e-3)) + #, limits = c(1e-7, 1e-3), labels = c('10^-7', '10^-5',
  # geom_label(data = ~ distinct(., Species.match, .keep_all = TRUE), aes(label=Species.num),
  #           x = 10, y = -8, hjust = 0, vjust = 0, size = 3) +
  # geom_text(data = ~ distinct(., Species.match, .keep_all = TRUE), aes(label=Species),
  #           x = 0.1, y = -3, hjust = 0, vjust = 1, size = 3) +
  # geom_text(data = ~ distinct(., Species.match, .keep_all = TRUE), aes(label=`Common name`),
```

```
#           x = 0.1, y = -4, hjust = 0, vjust = 1, size = 3) +
# scale_x_continuous(breaks = c(25, 50, 75)) +
facet_wrap(~ Species.match, ncol = 6) +
scale_shape_manual(values = c(4, 1)) +
labs(y = 'Second moment of area I (L^4)', x = 'Body region') +
theme_light() +
theme(strip.background = element_blank(),
      panel.grid.minor = element_blank())
```

I_plot



```
ggsave('aleyev_phylogeny.pdf', phylo_plot, width = 2, height = 4, units = 'in')
```

```
## Warning: Removed 26 rows containing missing values (`geom_text()`).
```

```
ggsave('aleyev_stiffness.pdf', EI_plot, width = 4.5, height = 4, units = 'in')
```

```
## Warning in self$trans$transform(x): NaNs produced
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning in self$trans$transform(x): NaNs produced
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning in self$trans$transform(x): NaNs produced
```

```
## Warning: Transformation introduced infinite values in discrete y-axis
```

```
## Warning: Removed 1 row containing missing values (`geom_line()`).
```

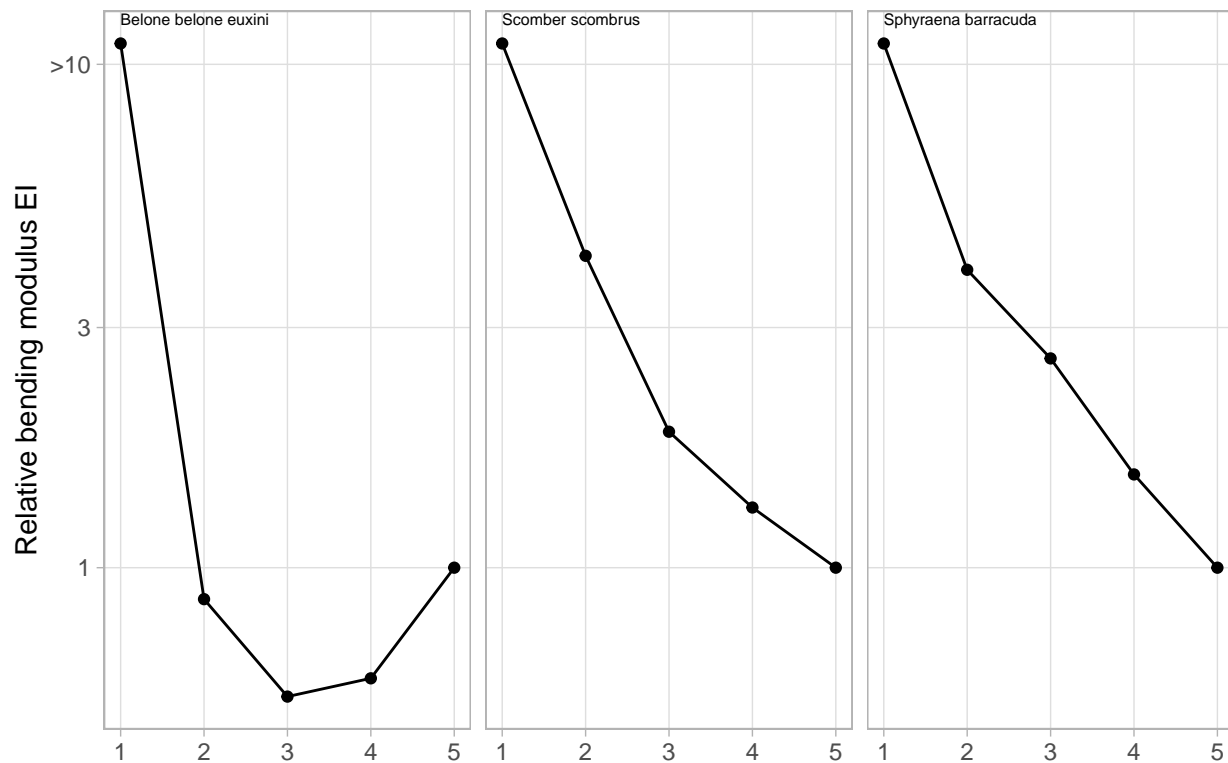
```
## Warning: Removed 1 rows containing missing values (`geom_point()`).
```

```
ggsave('aleyev_I.pdf', I_plot, width = 4.5, height = 4, units = 'in')
```

Simplified plot

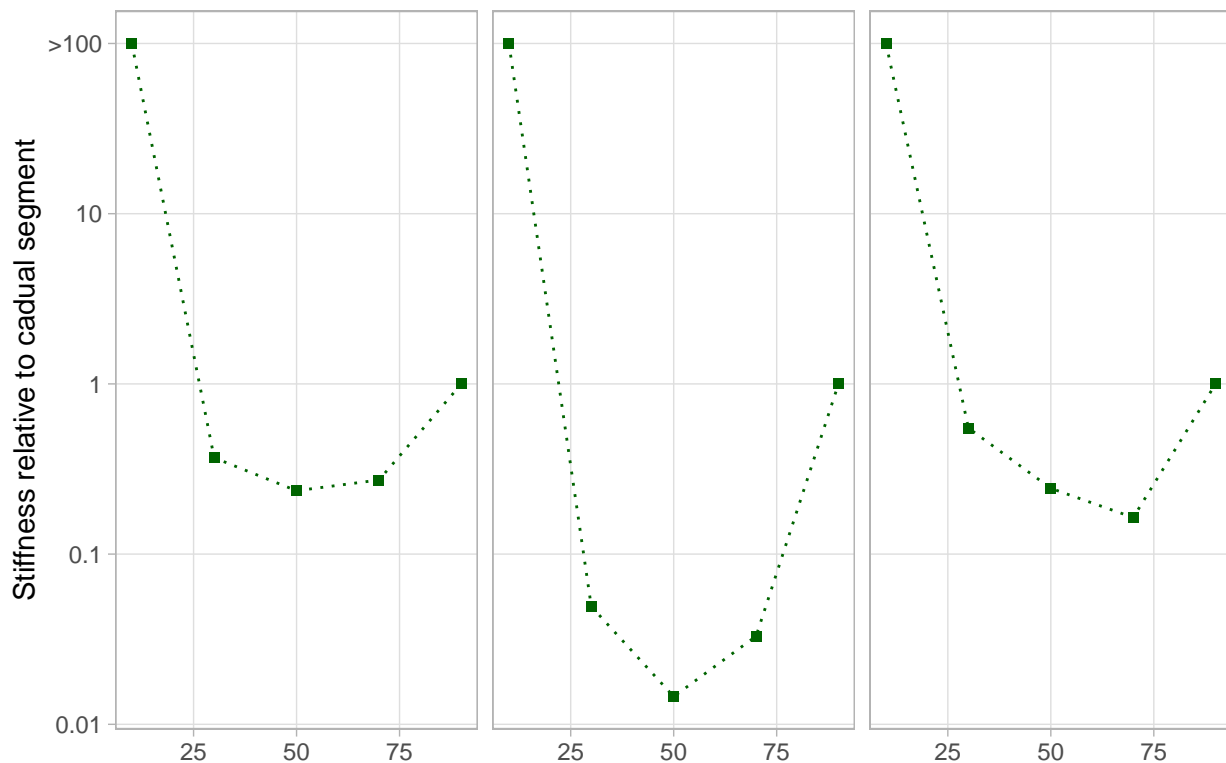
```
simplespecies <- c("Belone_belone",  
                  "Scomber_scombrus",  
                  "Sphyraena_barracuda")
```

```
EI_plot_simple <-  
  aleyevdata |>  
  filter(Species.match %in% simplespecies) |>  
  mutate(EI2cap = if_else(EI2 > 10, 11, EI2)) |>  
  ggplot(aes(x = point, y = EI2cap, group = Species)) +  
  geom_line() +  
  geom_point() +  
  geom_text(data = ~ distinct(., Species.match, .keep_all = TRUE), aes(label=Species),  
            x = 1, y = 1.1, hjust = 0, vjust = 1, size = 2) +  
  scale_y_log10(breaks = c(1, 3, 10), labels = c('1', '3', '>10')) +  
  
  scale_color_viridis_c(trans = 'log', limits = c(0.5, 10), direction = -1,  
                        breaks = c(0.5, 1, 3, 10), na.value = 'gray', option = 'plasma') +  
  facet_wrap(~ Species.match, ncol = 6) +  
  labs(y = 'Relative bending modulus EI', x = '') +  
  theme_light() +  
  theme(strip.background = element_blank(),  
        panel.grid.minor = element_blank())  
  
EI_plot_simple
```



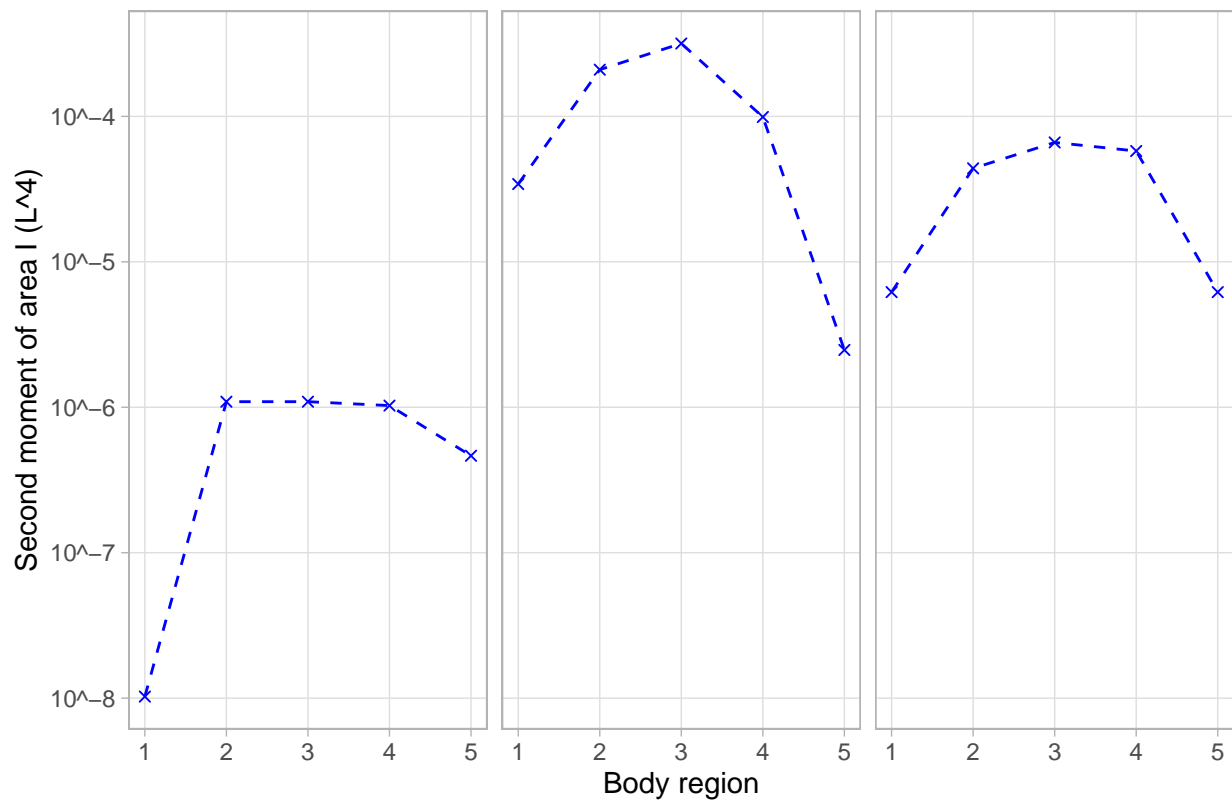
```
stiff_plot_simple <-
  aleyevdata |>
  filter(Species.match %in% simplespecies) |>
  mutate(Young2 = if_else(Young2 > 100, 100, Young2)) |>
  ggplot(aes(x = (loc-0.1)*100, y = Young2, group = Species)) +
  geom_line(color = "darkgreen", linetype = 3) +
  geom_point(shape = 15, color = "darkgreen") +
  scale_y_log10(breaks = c(0.01, 0.1, 1, 10, 1e2), labels = c('0.01', '0.1', '1', '10', '>100')) +
  scale_x_continuous(breaks = c(25, 50, 75)) +
  facet_wrap(~ Species.match, ncol = 6) +
  labs(y = 'Stiffness relative to cadual segment', x = '') +
  theme_light() +
  theme(strip.background = element_blank(),
        panel.grid.minor = element_blank())

stiff_plot_simple
```

```
I_plot_simple <-
  aleyevdata |>
  filter(Species.match %in% simplespecies) |>
  #mutate(Species.match = factor(Species.match, levels = speciesorder)) |>
  # mutate(I.L4.cap = case_when(I.L4 > 1e-3 ~ 1e-3,
  #                             I.L4 < 1e-7 ~ 1e-7,
  #                             TRUE ~ I.L4),
  #        I.outofrange = if_else(between(I.L4, 1e-7, 1e-3), 'in', 'out')) |>
  ggplot(aes(x = point, y = I.L4, group = Species)) +
  geom_line(color = 'blue', linetype = 2) +
  geom_point(shape = 4, color = 'blue') +
  scale_y_log10(breaks = c(1e-8, 1e-7, 1e-6, 1e-5, 1e-4),
               labels = c('10^-8', '10^-7', '10^-6', '10^-5', '10^-4')) +
  # geom_label(data = ~ distinct(., Species.match, .keep_all = TRUE), aes(label=Species.num),
  #           x = 10, y = -8, hjust = 0, vjust = 0, size = 3) +
  # geom_text(data = ~ distinct(., Species.match, .keep_all = TRUE), aes(label=Species),
  #          x = 0.1, y = -3, hjust = 0, vjust = 1, size = 3) +
  # geom_text(data = ~ distinct(., Species.match, .keep_all = TRUE), aes(label='Common name'),
  #          x = 0.1, y = -4, hjust = 0, vjust = 1, size = 3) +
  # scale_x_continuous(breaks = c(25, 50, 75)) +
  facet_wrap(~ Species.match, ncol = length(simplespecies)) +
  scale_shape_manual(values = c(4, 1)) +
  labs(y = 'Second moment of area I (L^4)', x = 'Body region') +
  theme_light() +
  theme(strip.background = element_blank(),
        panel.grid.minor = element_blank())

I_plot_simple
```

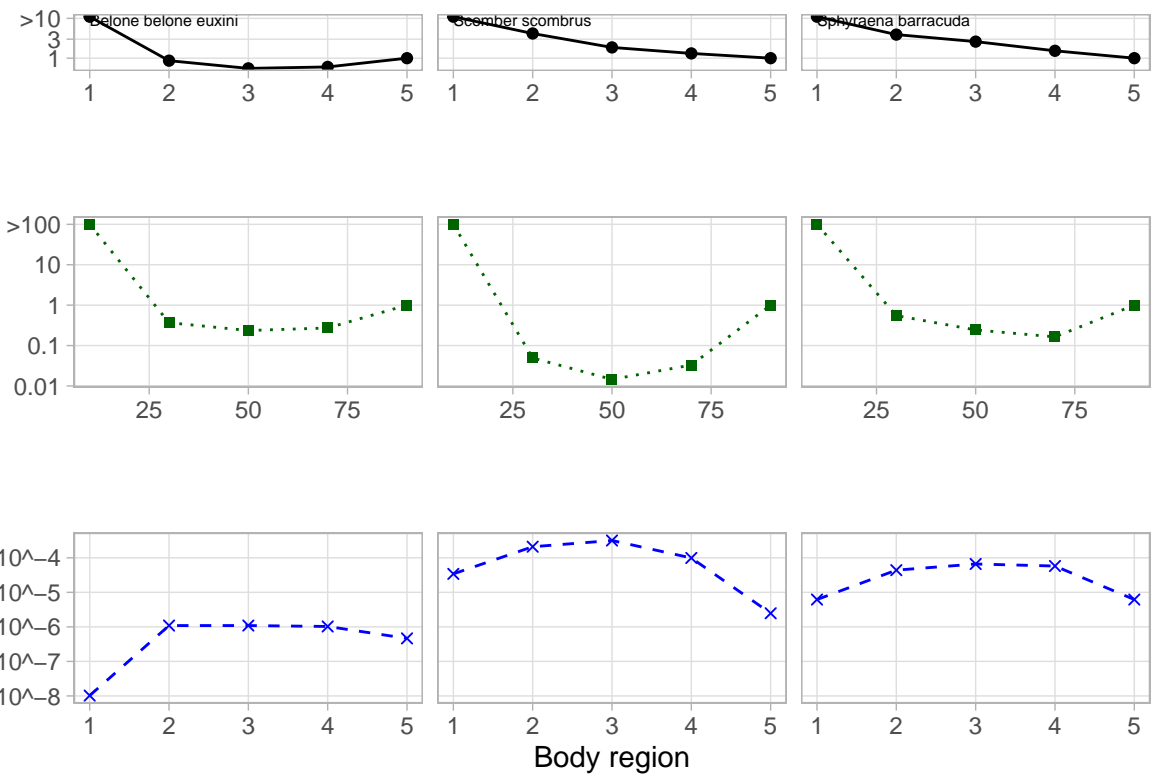


```

EI_plot_simple + stiff_plot_simple + I_plot_simple +
  plot_layout(nrow = 3, heights = c(1,3,3))

```

second moment of area I (less) relative to cadual segment depending model



```
ggsave('alejev_mech_simple.pdf', width = 5.5, height = 6, units = 'in')
```