

# Demo: Parameterized METE Models in R

Meng Xu & Ignasi Arranz

2025-01-08

## Introduction

The purpose of this document is to demonstrate how one can use the parameterized Maximum Entropy Theory of Ecology (pMETE) models to infer the metabolic scaling exponent  $b$ , as in the metabolic scaling relationship:  $R = aM^b$ , where  $R$  is the absolute metabolic rate of an individual and  $M$  is the body mass of the individual. To facilitate understanding of the modeling process, I wrote this document in RMarkdown on RStudio, with R code and output following each modeling step.

The general principle of the pMETE is to integrate the metabolic scaling relationship and the original METE developed by Harte (2011), and to infer  $b$  through fitting the predicted individual size distribution (ISD) of METE to the observed ISD. The METE is a constraint-based theory that predicts various macroecological patterns (e.g., species abundance distribution, species-area relationship, and ISD) within an ecological community and its constituent species populations. METE uses four macro-state variables:  $N$  (number of individuals, or total abundance),  $S$  (number of species),  $E$  (total rescaled metabolic rate), and  $A$  (total area) to define constraints needed for macroecological predictions. Here “rescaled” means the smallest individual metabolic rate or body mass is converted to one in the studied community or population. To predict ISD, METE first seeks the maximum entropy solution to the joint distribution of species abundance and individual metabolic rate within a community, subject to the constraints of mean species abundance ( $N/S$ ) and mean total metabolic rate per species ( $E/S$ ). This constrained maximization is performed using the Lagrange multiplier method. Then, by summing the product of species abundance and joint distribution over species abundance, one obtains the predicted individual metabolic rate distribution (Harte and Newman 2014). Finally, using the change of variable formula (Stegen and White 2008) and the metabolic scaling relationship, one can derive the predicted ISD. To compare the METE prediction with empirical data,  $b$  is assumed as a universal constant. Previous work has used this approach to test the predictions of METE (Newman et al. 2014, Xiao et al. 2015, Brush et al. 2022). In contrast, the pMETE models treat  $b$  as a free parameter instead of a universal constant, and allow estimation of  $b$  from the observed ISD. In addition, depending on the level that the ISD is analyzed (community or population), the pMETE models can generate an estimate of  $b$  specific to a given community or population. We denote the community-specific  $b$  as  $b_c$  and the population-specific  $b$  as  $b_p$ .

Several assumptions of the pMETE models must be realized before carrying out the models and interpreting the results:

- two ratios ( $N/S$  and  $E/S$ ) provide sufficient information on the ecological constraints of a community
- ISD is intrinsically related to the metabolic rate distribution via the metabolic scaling relationship
- each community has a unique  $b_c$  and each species population has a unique  $b_p$
- $b_c$  and  $b_p$  are estimated when predicted ISD matches most closely with the observed ISD

## Loading packages and data

Individual fish body mass data collected from the French streams are used here to demonstrate the use of pMETE models to obtain estimates of  $b_c$  and  $b_p$ . Note that the pMETE models are not restricted by

the particular structure of the French fish data examined here, and can be applied to any community or population with a vector of individual body mass.

All fishes within a single sampling event (“ope\_id”) are defined as a fish community. Individual fish and fish community are filtered based on their body length ( $\geq 60$  mm, since fish below this cut-off is underrepresented) and sample size ( $\geq 50$  individuals, which was tested to be the smallest sample size necessary for numerical convergence of the pMETE model, see (Xu and Arranz 2023)).

```
library(zipfR) # for calculating incomplete gamma function for quasi-Weibull likelihood
library(tidyverse) # for data analysis and plotting
library(nleqslv) # for solving nonlinear equation

data("size_data_com") # individual size data from a single fish community

test_data <- size_data_com %>%
  filter(mass != 0, length >= 60, outlier == 0) %>%
  group_by(ope_id) %>%
  filter(n() >= 50) %>%
  ungroup() %>%
  select(ope_id:length)
```

The table below shows the first six rows of test\_data. Each row represents an individual fish within the community. ope\_id identifies a fish community, species\_name shows the scientific name of the fish in the community, mass and length are the body mass (in g) and total length (in mm) of an individual fish, respectively.

```
head(test_data)
```

```
## # A tibble: 6 x 4
##   ope_id species_name  mass length
##   <int> <chr>         <dbl> <int>
## 1  10601 Cottus gobio  3.25     60
## 2  10601 Cottus gobio  8.48     85
## 3  10601 Cottus gobio  6.01     75
## 4  10601 Cottus gobio  3.56     62
## 5  10601 Cottus gobio  4.05     65
## 6  10601 Cottus gobio  3.25     60
```

There are 270 fishes from 5 species in the selected community. It is believed that a community with at least 4 species is necessary to guarantee the successful implementation of the METE (personal communication with John Harte). But this belief has not been rigorously tested.

```
nrow(test_data)
```

```
## [1] 270
```

```
length(unique(test_data$species_name))
```

```
## [1] 5
```

## Modeling

The following steps show how to use pMETE models to estimate  $b_c$  and  $b_p$  for a multi-species (fish) community and a particular species (brown trout) population within the community. If individual size data are available for a species population only, such as in communities with singleton species, or for a monodominant community where a single species is most predominant in number, then an approximate pMETE model at the population level (from step 3) should be used directly. This case will be demonstrated following this section.

### step 1

In this step we calculate  $S$  and  $N$  from the selected community and estimate  $\beta$  (or  $\lambda_1 + \lambda_2$ ).  $\lambda_1$  and  $\lambda_2$  are the Lagrange multipliers in the Lagrange multiplier method used to solve the joint distribution. In the original METE,  $\lambda_1$  and  $\lambda_2$  are solved numerically from a 2 by 2 system of nonlinear equations representing the two constraints. Also,  $b_c$  is assumed as 0.75 and  $E$  is calculated by summing the rescaled body mass raised to 0.75 (based on the metabolic scaling relationship). In the pMETE, since  $b_c$  is treated as a free parameter,  $E$  becomes a function of  $b_c$ . In addition, the nonlinear system contains two equations (from the two constraints) but three unknowns ( $\lambda_1$ ,  $\lambda_2$ , and  $b_c$ ), and is therefore underdetermined. One can rearrange the system algebraically to derive an approximate single equation for  $\beta$  and express  $\lambda_2$  as a function of  $b_c$ , but  $b_c$  remains unknown. The realization of this step relies on two approximate conditions (see Box 7.4 in Harte (2011) and step 2).

We append `test_data` with the values of  $S$  (column “S\_0”),  $N$  (column “N\_0”), and  $\beta$ . We also added columns “E\_0\_mete” and “L2\_com\_mete”, which show, respectively, values of  $E$  and  $\lambda_2$  of the original METE model. They are needed for the model comparison later in step 4.

```
eqn_beta <- function(x, N_0, S_0) {  
  # equation for beta  
  sum(exp(-x * seq(1, N_0)))/sum((exp(-x * seq(1, N_0))/seq(1,  
    N_0)) - N_0/S_0  
}  
com_summary <- test_data %>%  
  group_by(ope_id) %>%  
  summarise(N_0 = length(mass), S_0 = length(unique(species_name)),  
    E_0_mete = sum((mass/min(mass))^0.75)) %>%  
  ungroup() %>%  
  # this line finds lambda_2 for the original METE  
mutate(L2_com_mete = S_0/(E_0_mete - N_0))  
  
beta <- com_summary %>%  
  group_by(ope_id) %>%  
  # solve for beta  
do(beta = nleqslv(x = 1/.$N_0, N_0 = .$N_0, S_0 = .$S_0, fn = eqn_beta)$x) %>%  
  unnest(cols = c(beta))  
  
com_analysis <- test_data %>%  
  left_join(com_summary, by = "ope_id") %>%  
  # prepare data for modeling  
left_join(beta, by = "ope_id")  
  
head(com_analysis)
```

```
## # A tibble: 6 x 9
```

##	ope_id	species_name	mass	length	N_0	S_0	E_0_mete	L2_com_mete	beta
##	<int>	<chr>	<dbl>	<int>	<int>	<int>	<dbl>	<dbl>	<dbl>
## 1	10601	Cottus gobio	3.25	60	270	5	2112.	0.00271	-0.00226
## 2	10601	Cottus gobio	8.48	85	270	5	2112.	0.00271	-0.00226
## 3	10601	Cottus gobio	6.01	75	270	5	2112.	0.00271	-0.00226
## 4	10601	Cottus gobio	3.56	62	270	5	2112.	0.00271	-0.00226
## 5	10601	Cottus gobio	4.05	65	270	5	2112.	0.00271	-0.00226
## 6	10601	Cottus gobio	3.25	60	270	5	2112.	0.00271	-0.00226

## step 2

In this step we estimate  $b_c$  and  $\lambda$ 's by fitting the predicted ISD to the observed ISD for the fish community.  $\lambda_2$  and  $b_c$  are community-level parameters and needed for the population-level modeling (see step 3). From the constraint equations, we express  $\lambda_2 = S/(E - N) = S/(\sum m^{b_c} - N)$  ( $m$  is the rescaled individual body mass) and  $\lambda_1 = \beta - \lambda_2$ , so that  $\lambda_1$  and  $\lambda_2$  become functions of  $b_c$ , which is the only free parameter in the model.

The fitting is done using the maximum likelihood method. To do this we first derive the probability density function of the predicted ISD (detailed derivation in the Supporting Information of Xu (2020)), with  $b_c$  as a free parameter in the function. Then we use the observed individual size data (mass column) to calculate the log-likelihood function of the predicted ISD, and search for the  $b_c$  that maximizes the log-likelihood function using the Brent method, with boundaries of  $b_c$  from 0 to 10. Then we remove  $b_c$  stuck near the upper bound of 10 because they are empirically unlikely. Once  $b_c$  is estimated,  $\lambda_1$  and  $\lambda_2$  can be solved because they are functions of  $b_c$ . In this particular community,  $b_c$  is estimated as 0.654 (column "b\_com\_pmete" in fish\_com\_b).

```
negLL_com_pmete = function(par, x, xmin, beta, N_0, S_0) {
  # community-level metabolic scaling exponent
  bc = par[1]
  # lambda_2 by eqn 7.26 in Harte (2011)
  L2 = S_0/(sum((x/xmin)^bc) - N_0)
  # lambda_1
  L1 = beta - L2
  negLL_exact = -N_0 * log(bc * L2) + (1 - bc) * sum(log(x/xmin)) +
    sum(L1 + L2 * (x/xmin)^bc) + sum(log((1 - exp(-L1 - L2 *
    (x/xmin)^bc))^2)) - sum(log(1 - (N_0 + 1) * exp(-N_0 *
    (L1 + L2 * (x/xmin)^bc)) + N_0 * exp(-(N_0 + 1) * (L1 +
    L2 * (x/xmin)^bc)))) + N_0 * log((exp(-beta) - exp(-beta *
    (N_0 + 1)))/(1 - exp(-beta)) - (exp(-(L1 + L2 * sum((x/xmin)^bc))) -
    exp(-(L1 + L2 * sum((x/xmin)^bc))) * (N_0 + 1)))/(1 -
    exp(-(L1 + L2 * sum((x/xmin)^bc)))))
  return(negLL_exact)
}

fish_com_b = com_analysis %>%
  group_by(ope_id) %>%
  do(b_com_pmete = optim(par = 0.75, negLL_com_pmete, x = .$mass,
    xmin = min(.$mass), beta = mean(.$beta), N_0 = mean(.$N_0),
    S_0 = mean(.$S_0), method = "Brent", lower = 0, upper = 10)$par[1],
    L2_com_pmete = mean(.$S_0)/(sum((.$mass/min(.$mass))^optim(par = 0.75,
    negLL_com_pmete, x = .$mass, xmin = min(.$mass),
    beta = mean(.$beta), N_0 = mean(.$N_0), S_0 = mean(.$S_0),
    method = "Brent", lower = 0, upper = 10)$par[1]) -
    mean(.$N_0)), L1_com_pmete = mean(.$beta) - mean(.$S_0)/(sum((.$mass/min(.$mass))^optim(par
```

```

negLL_com_pmete, x = .$mass, xmin = min(.$mass),
beta = mean(.$beta), N_0 = mean(.$N_0), S_0 = mean(.$S_0),
method = "Brent", lower = 0, upper = 10)$par[1]) -
mean(.$N_0)), negll_com_pmete = optim(par = 0.75,
negLL_com_pmete, x = .$mass, xmin = min(.$mass),
beta = mean(.$beta), N_0 = mean(.$N_0), S_0 = mean(.$S_0),
method = "Brent", lower = 0, upper = 10)$value, negll_com_mete = negLL_com_pmete(par = 0.75,
x = .$mass, xmin = min(.$mass), beta = mean(.$beta),
N_0 = mean(.$N_0), S_0 = mean(.$S_0))) %>%
unnest(cols = c(b_com_pmete, L2_com_pmete, L1_com_pmete,
negll_com_pmete, negll_com_mete)) %>%
filter(b_com_pmete < 9)

fish_com_b

## # A tibble: 1 x 6
##   ope_id b_com_pmete L2_com_pmete L1_com_pmete negll_com_pmete negll_com_mete
##   <int>     <dbl>     <dbl>     <dbl>         <dbl>         <dbl>
## 1  10601         0.654         0.00395        -0.00621         1006.         1007.

```

Finally, we only keep the community if the two approximate conditions needed for solving  $\beta$  and expressing  $\lambda$ s as functions of  $b_c$  are satisfied (for our selected community both conditions are satisfied).

```

com_analysis2 <- right_join(com_analysis, fish_com_b, by = "ope_id")

com_check_pmete <- com_analysis2 %>%
  group_by(ope_id) %>%
  summarise(cond1a_pmete = sum(exp(-mean(beta) * seq(1, n()))),
    cond1b_pmete = sum((mass/min(mass))^mean(b_com_pmete)) *
      sum(exp(-(mean(L1_com_pmete) + sum((mass/min(mass))^mean(b_com_pmete)) *
        mean(L2_com_pmete)) * seq(1, n()))), cond2a_pmete = sum(exp(-mean(beta) *
      seq(1, n()))/seq(1, n())), cond2b_pmete = sum(exp(-(mean(L1_com_pmete) +
      sum((mass/min(mass))^mean(b_com_pmete)) * mean(L2_com_pmete)) *
      seq(1, n()))/seq(1, n())) %>%
  mutate(cond1_pmete = cond1a_pmete/cond1b_pmete > 10, cond2_pmete = cond2a_pmete/cond2b_pmete >
    10) %>%
  dplyr::select(ope_id, cond1_pmete, cond2_pmete)

head(com_check_pmete)

## # A tibble: 1 x 3
##   ope_id cond1_pmete cond2_pmete
##   <int> <lg1>      <lg1>
## 1  10601 TRUE        TRUE

```

### step 3

This step estimates  $b_p$ , the metabolic scaling exponent for the brown trout population from the selected community. This step is similar to step 2, but is performed at the population level. We first filter the population so that it has sufficient sample size ( $\geq 50$ ). The particular population in the selected community has 219 individual brown trout. Then we derive the probability density function of the predicted ISD (detailed derivation in the Supporting Information of Xu (2020)), and fit it to the brown trout individual size data

using the maximum likelihood method. Notice that in this population-level predicted ISD,  $b_c$  and  $\lambda_2$  are known values from the community-level modeling (step 2).  $b_p$  is the only free parameter to be estimated. In this particular population,  $b_p$  is estimated as 0.325 (column “b\_pop\_pmete” in pop\_stat).

```
com_data_new <- full_join(com_analysis2, com_check_pmete, by = "ope_id") %>%
  group_by(ope_id) %>%
  filter(cond1_pmete == TRUE, cond2_pmete == TRUE) %>%
  ungroup() %>%
  mutate(mass_bt = ifelse(species_name == "Salmo trutta", mass,
    NA))

pop_data_size_filter <- com_data_new %>%
  group_by(ope_id) %>%
  summarise(n_bt = sum(species_name == "Salmo trutta")) %>%
  filter(n_bt > 50) %>%
  ungroup()

com_data_new2 <- com_data_new %>%
  filter(ope_id %in% unique(pop_data_size_filter$ope_id))

com_data_new2 %>%
  filter(species_name == "Salmo trutta")
```

```
## # A tibble: 219 x 17
##   ope_id species_name mass length N_0 S_0 E_0_mete L2_com_mete beta
##   <int> <chr>      <dbl> <int> <int> <int> <dbl> <dbl> <dbl>
## 1 10601 Salmo trutta 2.58 62 270 5 2112. 0.00271 -0.00226
## 2 10601 Salmo trutta 4.04 72 270 5 2112. 0.00271 -0.00226
## 3 10601 Salmo trutta 24.8 132 270 5 2112. 0.00271 -0.00226
## 4 10601 Salmo trutta 57.6 175 270 5 2112. 0.00271 -0.00226
## 5 10601 Salmo trutta 32.8 145 270 5 2112. 0.00271 -0.00226
## 6 10601 Salmo trutta 28.3 138 270 5 2112. 0.00271 -0.00226
## 7 10601 Salmo trutta 4.21 73 270 5 2112. 0.00271 -0.00226
## 8 10601 Salmo trutta 28.3 138 270 5 2112. 0.00271 -0.00226
## 9 10601 Salmo trutta 4.57 75 270 5 2112. 0.00271 -0.00226
## 10 10601 Salmo trutta 73.6 190 270 5 2112. 0.00271 -0.00226
## # i 209 more rows
## # i 8 more variables: b_com_pmete <dbl>, L2_com_pmete <dbl>,
## # L1_com_pmete <dbl>, negll_com_pmete <dbl>, negll_com_mete <dbl>,
## # cond1_pmete <lgl>, cond2_pmete <lgl>, mass_bt <dbl>
```

```
negLL_pop_pmete = function(par, x, xmin, n, L2, bc, m_com) {
  # population-level metabolic scaling exponent
  bp = par[1]
  # exact likelihood function from eqn 7.25 in Harte
  # (2011)
  negLL_exact = -n * log(bp * L2 * n) + (1 - bp) * sum(log(x/xmin)) +
    L2 * n * sum((x/xmin)^bp) + n * log(exp(-n * L2) - exp(-n *
    L2 * sum((m_com/min(m_com))^bc)))
  return(negLL_exact)
}

fish_pop_b = com_data_new2 %>%
```

```

group_by(ope_id) %>%
do(b_pop_pmete = optim(par = 0.75, negLL_pop_pmete, x = as.numeric(na.omit(. $mass_bt)),
  xmin = min(as.numeric(na.omit(. $mass_bt))), n = length(as.numeric(na.omit(. $mass_bt))),
  L2 = mean(. $L2_com_pmete), bc = mean(. $b_com_pmete),
  m_com = . $mass, method = "Brent", lower = 0, upper = 10) $par[1],
negll_pop_pmete = optim(par = 0.75, negLL_pop_pmete,
  x = as.numeric(na.omit(. $mass_bt)), xmin = min(as.numeric(na.omit(. $mass_bt))),
  n = length(as.numeric(na.omit(. $mass_bt))), L2 = mean(. $L2_com_pmete),
  bc = mean(. $b_com_pmete), m_com = . $mass, method = "Brent",
  lower = 0, upper = 10) $value, negll_pop_mete = negLL_pop_pmete(par = 0.75,
  x = as.numeric(na.omit(. $mass_bt)), xmin = min(as.numeric(na.omit(. $mass_bt))),
  n = length(as.numeric(na.omit(. $mass_bt))), L2 = mean(. $L2_com_mete),
  bc = 0.75, m_com = . $mass)) %>%
unnest(cols = c(b_pop_pmete, negll_pop_pmete, negll_pop_mete)) %>%
filter(b_pop_pmete < 9)

pop_analysis <- right_join(com_data_new2, fish_pop_b, by = "ope_id") %>%
  filter(species_name == "Salmo trutta")

pop_stat <- pop_analysis %>%
  group_by(ope_id) %>%
  do(n_0 = length(. $mass), L2_com_pmete = mean(. $L2_com_pmete),
    b_pop_pmete = mean(. $b_pop_pmete), aicc_pop_pmete = 2 *
      1 + 2 * mean(. $negll_pop_pmete) + (2 * 1^2 + 2 *
      1)/(length(. $mass) - 1 - 1), L2_com_mete = mean(. $L2_com_mete),
    aicc_pop_mete = 2 * mean(. $negll_pop_mete)) %>%
  unnest(cols = c(n_0, L2_com_pmete, b_pop_pmete, aicc_pop_pmete,
    L2_com_mete, aicc_pop_mete))

pop_stat

## # A tibble: 1 x 7
##   ope_id  n_0 L2_com_pmete b_pop_pmete aicc_pop_pmete L2_com_mete aicc_pop_mete
##   <int> <int>      <dbl>      <dbl>          <dbl>      <dbl>      <dbl>
## 1  10601   219      0.00395      0.325          1576.      0.00271      2392.

```

#### step 4

The last step involves comparing the predicted ISD between the pMETE and other models (i.e., power-law distribution, exponential distribution, Weibull distribution, and quasi-Weibull distribution) at the population level. This step is needed in order to evaluate whether the pMETE gives reasonable fit of the observed ISD. All models are fitted using the maximum likelihood method and compared using the model AICc. For each brown trout population, if AICc of the pMETE is greater than the lowest AICc by 2 or less, then the corresponding population and estimate of  $b_p$  are kept. For the brown trout population in this particular example, pMETE and Weibull models are selected because they have the lowest AICc. Hence,  $b_p$  from the pMETE is kept. The rationale of this step is: pMETE that has the most empirical support is believed to yield accurate estimate of  $b_p$  (assumption #4). Note that this step can be similarly performed to evaluate the pMETE fit at the community level (step 2). In addition, further model evaluations can be done using other statistical metrics (e.g., Kolmogorov-Smirnov test and R-squared). They are omitted here but can be included later if necessary.

```

# left-truncated exponential
negLL_exp = function(par, x, xmin, n) {
  # metabolic scaling exponent
  lambda = par[1]
  negLL = -n * log(lambda) - n * lambda + lambda * sum(x/xmin)
  return(negLL)
}

# left-truncated power-law
negLL_pow = function(par, x, xmin, n) {
  # power-law distribution parameter
  a = par[1]
  negLL = -n * log(a - 1) + a * sum(log(x/xmin))
  return(negLL)
}

# left-truncated Weibull
negLL_wei = function(par, x, n, xmin) {
  # scale parameter
  a = par[1]
  # shape parameter
  b = par[2]
  negll = -n * log(b) + n * b * log(a) + (1 - b) * sum(log(x/xmin)) +
    sum(((x/xmin)/a)^b) - n * (1/a)^b
  return(negll)
}

# left-truncated quasi-Weibull
negLL_qwei = function(par, x, xmin, n) {
  # quasi-Weibull distribution parameter alpha
  qal = par[1]
  # quasi-Weibull distribution parameter beta
  qbe = par[2]
  # quasi-Weibull distribution parameter gamma
  qga = par[3]
  negLL = -n * log(qal + qga) + qal * n * log(qbe) + n * log(igamma(qal/(qal +
    qga), (1/qbe)^(qal + qga), lower = F)) - (qal - 1) *
    sum(log(x/xmin)) + sum(((x/xmin)/qbe)^(qal + qga))
  return(negLL)
}

pop_stat_othermod <- pop_analysis %>%
  group_by(ope_id) %>%
  do(p1_pop_exp = optim(par = 1, fn = negLL_exp, x = .$mass,
    n = length(.$mass), xmin = min(.$mass), method = "Brent",
    lower = 0, upper = 10)$par, aicc_pop_exp = 2 * 1 + 2 *
    optim(par = 1, fn = negLL_exp, x = .$mass, n = length(.$mass),
      xmin = min(.$mass), method = "Brent", lower = 0,
      upper = 10)$value + (2 * 1^2 + 2 * 1)/(length(.$mass) -
    1 - 1), p1_pop_pow = optim(par = 0.75, fn = negLL_pow,
    x = .$mass, n = length(.$mass), xmin = min(.$mass), method = "Brent",
    lower = 0, upper = 10)$par, aicc_pop_pow = 2 * 1 + 2 *
    optim(par = 0.75, fn = negLL_pow, x = .$mass, n = length(.$mass),

```



```

      xmin = min(.$mass), method = "Brent", lower = 0,
      upper = 10)$value + (2 * 1^2 + 2 * 1)/(length(.$mass) -
1 - 1), p1_pop_wei = optim(par = c(10, 2), fn = negLL_wei,
x = .$mass, n = length(.$mass), xmin = min(.$mass), control = list(maxit = 10000))$par[1],
p2_pop_wei = optim(par = c(10, 2), fn = negLL_wei, x = .$mass,
n = length(.$mass), xmin = min(.$mass), control = list(maxit = 10000))$par[2],
aicc_pop_wei = 2 * 2 + 2 * optim(par = c(10, 2), fn = negLL_wei,
x = .$mass, n = length(.$mass), xmin = min(.$mass),
control = list(maxit = 10000))$value + (2 * 2^2 +
2 * 2)/(length(.$mass) - 2 - 1), p1_pop_qwei = optim(par = c(2,
2, 2), fn = negLL_qwei, x = .$mass, n = length(.$mass),
xmin = min(.$mass), control = list(maxit = 10000))$par[1],
p2_pop_qwei = optim(par = c(2, 2, 2), fn = negLL_qwei,
x = .$mass, n = length(.$mass), xmin = min(.$mass),
control = list(maxit = 10000))$par[2], p3_pop_qwei = optim(par = c(2,
2, 2), fn = negLL_qwei, x = .$mass, n = length(.$mass),
xmin = min(.$mass), control = list(maxit = 10000))$par[3],
aicc_pop_qwei = 2 * 3 + 2 * optim(par = c(2, 2, 2), fn = negLL_qwei,
x = .$mass, n = length(.$mass), xmin = min(.$mass),
control = list(maxit = 10000))$value + (2 * 3^2 +
2 * 3)/(length(.$mass) - 3 - 1)) %>%
unnest()

# combine all model statistics
pop_stat_allmod <- full_join(pop_stat, pop_stat_othermod, by = "ope_id")

pop_stat_allmod_aicc <- pop_stat_allmod %>%
  dplyr::select(ope_id, contains("aicc")) %>%
  gather(key = "mod", value = "aicc", 2:7) %>%
  mutate(model = substring(mod, 10)) %>%
  # extract aicc columns and models
dplyr::select(-mod)

pop_stat_selectmod_aicc <- pop_stat_allmod_aicc %>%
  group_by(ope_id) %>%
  mutate(aicc_diff = aicc - min(aicc)) %>%
  filter(aicc_diff <= 2) %>%
  # choose the model with a delta aicc <=2
ungroup()

# selected model for this brown trout population
pop_stat_selectmod_aicc

## # A tibble: 2 x 4
##   ope_id aicc model aicc_diff
##   <int> <dbl> <chr>    <dbl>
## 1  10601 1576. pmete      0
## 2  10601 1578. wei      1.90

```

After this step, the remaining  $b_p$  can be combined with external factors about the sampling event (e.g., temperature, human footprint) to analyze the effect of these factors on  $b_p$ .

## Cases when individual body mass are available for species population only

In this section, we suppose that individual body mass are available for the brown trout species only. For such data, we apply an approximate pMETE model for the population-level ISD. The approximate model has two free parameters:  $b_p$  and  $\lambda_2$ . The approximate model does not require the community-level  $b_c$  or  $\lambda_2$  as inputs, and can be directly used to estimate  $b_p$  for the focal species population. For the brown trout data, the approximate model estimates  $b_p$  as 0.305, which differs slightly from the previous estimate (0.325) due to the difference in the predicted ISD (one exact and one approximate).

```
negLL_pop_pmete_approx = function(par, x, xmin, n) {
  b = par[1] # metabolic scaling exponent
  L2 = par[2] # Lagrange multiplier lambda2 as a parameter
  negLL_approx = -n * log(b * L2 * n) + (1 - b) * sum(log(x/xmin)) +
    L2 * n * sum((x/xmin)^b - 1) # approximate likelihood function from eqn 7.25 in Harte (2011)
  return(negLL_approx)
}

fish_pop_only_b = com_data_new2 %>%
  filter(species_name == "Salmo trutta") %>%
  group_by(ope_id) %>%
  do(b_pop_only_pmete = optim(par = c(0.75, 0.01), negLL_pop_pmete_approx,
    x = .$mass, xmin = min(.$mass), n = length(.$mass))$par[1],
    negll_pop_only_pmete = optim(par = c(0.75, 0.01), negLL_pop_pmete_approx,
    x = .$mass, xmin = min(.$mass), n = length(.$mass))$value) %>%
  unnest(cols = c(b_pop_only_pmete, negll_pop_only_pmete)) %>%
  filter(b_pop_only_pmete < 9)

pop_only_analysis <- right_join(com_data_new2, fish_pop_only_b,
  by = "ope_id") %>%
  filter(species_name == "Salmo trutta")

pop_only_stat <- pop_only_analysis %>%
  group_by(ope_id) %>%
  do(n_0 = length(.$mass), b_pop_only_pmete = mean(.$b_pop_only_pmete),
    aicc_pop_only_pmete = 2 * 2 + 2 * mean(.$negll_pop_only_pmete) +
    (2 * 2^2 + 2 * 2)/(length(.$mass) - 2 - 1)) %>%
  unnest(cols = c(n_0, b_pop_only_pmete, aicc_pop_only_pmete))

pop_only_stat

## # A tibble: 1 x 4
##   ope_id   n_0 b_pop_only_pmete aicc_pop_only_pmete
##   <int> <int>         <dbl>         <dbl>
## 1  10601   219           0.305           1578.
```

Brush, M., Matthews, T. J., Borges, P. A., & Harte, J. (2022). Land use change through the lens of macroecology: insights from Azorean arthropods and the maximum entropy theory of ecology. *Ecography*, 2022(8), e06141.

Harte, J. (2011). *Maximum entropy and ecology: a theory of abundance, distribution, and energetics*. OUP Oxford.

Harte, J., & Newman, E. A. (2014). Maximum information entropy: a foundation for ecological theory. *Trends in ecology & evolution*, 29(7), 384-389.

- Newman, E. A., Harte, M. E., Lowell, N., Wilber, M., & Harte, J. (2014). Empirical tests of within-and across-species energetics in a diverse plant community. *Ecology*, 95(10), 2815-2825.
- Stegen, J. C., & White, E. P. (2008). On the relationship between mass and diameter distributions in tree communities. *Ecology Letters*, 11(12), 1287-1293.
- Xiao, X., McGlinn, D. J., & White, E. P. (2015). A strong test of the maximum entropy theory of ecology. *The American Naturalist*, 185(3), E70-E80.
- Xu, M. (2020). Parameterized maximum entropy models predict variability of metabolic scaling across tree communities and populations. *Ecology*, 101(6), e03011.
- Xu, M., & Arranz, I. (2023). Maximum entropy models reveal spatial variation of metabolic scaling in stream fish communities. *Journal of Animal Ecology*, 92(6), 1203-1215.

## References

- Brush, M., T. J. Matthews, P. A. V. Borges, and J. Harte. 2022. Land use change through the lens of macroecology: Insights from Azorean arthropods and the maximum entropy theory of ecology. *Ecography* 2022:e06141.
- Harte, J. 2011. *Maximum Entropy and Ecology: A Theory of Abundance, Distribution, and Energetics*. Oxford University Press.
- Harte, J., and E. A. Newman. 2014. Maximum information entropy: A foundation for ecological theory. *Trends in Ecology & Evolution* 29:384–389.
- Newman, E. A., M. E. Harte, N. Lowell, M. Wilber, and J. Harte. 2014. Empirical tests of within- and across-species energetics in a diverse plant community. *Ecology* 95:2815–2825.
- Stegen, J. C., and E. P. White. 2008. On the relationship between mass and diameter distributions in tree communities. *Ecology Letters* 11:1287–1293.
- Xiao, X., D. J. McGlinn, and E. P. White. 2015. A Strong Test of the Maximum Entropy Theory of Ecology. *The American Naturalist* 185:E70–E80.
- Xu, M. 2020. Parameterized maximum entropy models predict variability of metabolic scaling across tree communities and populations. *Ecology* 101:e03011.
- Xu, M., and I. Arranz. 2023. Maximum entropy models reveal spatial variation of metabolic scaling in stream fish communities. *Journal of Animal Ecology* 92:1203–1215.