R Functions for Grand Canyon Food Base Macroinvertebrate Production

Author: Ben Koch (bkoch@uwyo.edu)

Functions:

```
get.dates
igr.prod
sf.prod
pb.prod
wrapper.site.yr
hab.weight
sampinfo.site.yr
```

General Information:

Example

The example uses data from RM30, and also includes some data for chironomids at RM0. Note that the RM30 portion of the data is preliminary data and may contain substantial errors. It should not be used other than for illustrative purposes.

The example code "Example_5" makes use of a specific set of folders in an attempt to keep data files, scripts, and output somewhat organized. To run this code, place the folder "Example_GC_Prod_5" (with all of its contents) wherever you want to keep it on your computer. Then open the example code "Example_5" and enter the new location of this folder in the "`set.wd`" command (at the very top) so that R knows where to look for the data files, scripts, output folders, etc. (e.g., you might type: "`C:/Documents and Settings/Ben/My Documents/Grand Canyon Stuff/Example_GC_Prod_5`"). Save the change you just made to "Example_5", and you are good to go.

To run the example, open up R, and if you've not yet installed the package "chron", do so now (select the "Packages" drop-down menu at the top and then "Install package(s)…"). You can now run the example code "Example_5", and everything should work.

Note that the call to `wrapper.site.yr` (near the end) will write a bunch of text files to the previously empty "Output" folder. If you play around with this and run a bunch of different calls to `wrapper.site.yr`, you may want to delete these output files periodically so that you don't end up with hundreds of them.

Data Format

Data needs to be saved in a tab-delimited text file format, and must be organized as follows, with these exact column headings used, and in this order:

Column 1: "`SITE`"	(sample sites: "`RM0`", "`RM30`", etc.)
Column 2: "`SAMPLE`"	(sample numbers: `1`, `2`, `3`, etc.)
Column 3: "`DATE`"	(sample date: "`1/5/2008`", "`9/15/2007`", etc.)
Column 4: "`HABITAT`"	(sample habitat: "`COBBLE`", "`DEPOSITIONAL`", or "`TALUS.CLIFF`")

Column 5: “`TAXON`” (taxon: “`Chironomid`”, “`NZMS`”, etc.)
Columns 6-36 (or as many as are needed):
Headings of these latter columns (“`0.5`”, “`1`”, “`2`”, “`3`”, …) denote mm size classes, and the values in these columns are abundances (units: #·m^{-2})

It is often easiest to get the data formatted and organized in excel before saving it to a tab-delimited text file. It is important that there are no empty cells in the excel file before converting it to a text file. Zeroes should be filled in as “`0`”, and truly missing data should be coded as “`NA`”. Rows with an “`NA`” entered for any abundance value will automatically be excluded from production calculations.

The data does not need to include all non-occurrences of taxa (but the functions will work fine if it does). For example, if for a given site, habitat, and sampling date, there were chironomids found in samples 1, 2, 3, and 5, but none in sample 4, there does not need to be a row of zeroes included to indicate that chironomids had abundance = 0 for all size classes in sample 4. As long as the data has at least a single occurrence of at least one other taxon in sample 4, the functions will include sample 4 as actually being sampled for all taxa.

Most of the functions (`get.dates`, `igr.prod`, `sf.prod`, `pb.prod`, `wrapper.site.yr`, `sampinfo.site.yr`) need Julian dates to run properly. In the example, right after the “`read.table`” command that imports the data, I have included code that creates a new column of Julian dates from the “`DATE`” column and inserts this new “`JULIAN`” column between the “`SAMPLE`” and “`DATE`” columns. Note that the “chron” package needs to be installed in order for this to work. The code sets day 1 equal to 1/1/2006, but this can be modified by the user as necessary. Alternatively, the user may choose to skip running this step of the example code, and instead create the “`JULIAN`” column (between the “`SAMPLE`” and “`DATE`” columns) prior to importing the data to R.

The `wrapper.site.yr` function and the `igr.prod` function rely on consistently-spelled taxa names in order to run properly. I have assumed that the names of taxa in the example data will be the same as those used in all other data yet to come (and in the taxonomic information data described below). Changing any of these in the abundance data will require making identical changes to the names in the taxonomic information data and to the code of the `igr.prod` function (for those taxa that use the instantaneous growth rate method of production).

Format for Taxon Information

The `wrapper.site.yr` function and the `sampinfo.site.yr` function use a separate “`TAXA`” data frame to grab length-mass parameters, growth parameters, cpi’s, P/B values, etc. associated with each taxon. This “Taxon Information” data must be organized as follows, with these exact column headings used:

Column 1: “`TAXON`”	(taxon: “`Chironomid`”, “`NZMS`”, etc.)
Column 2: “`METHOD`”	(method of production to use: “`igr`”, “`sf`”, “`pb`”, or “`NA`”)
Column 3: “`LM.a`”	(length-mass parameter *a*: 0.00022, 0.0006, etc.)
Column 4: “`LM.b`”	(length-mass parameter *b*: 2.871, 2.77, etc.)

Column 5: “`LM.p.ash`” (% ash: usually 0, unless regression is for DM: 10, 15, etc.)
Column 6: “`Growth.parm.a`” (growth parameter *a*: 0.051, -0.006, `NA`, etc.)
Column 7: “`Growth.parm.b`” (growth parameter *b*: 0.068, 0.0286, `NA`, etc.)
Column 8: “`Growth.parm.c`” (growth parameter *c*: 0.006, `NA`, etc.)
Column 9: “`Growth.parm.d`” (growth parameter *d*: 0.006, `NA`, etc.)
Column 10: “`rm0.min.cpi.days`” (minimum cpi in days^{-1}: 300, 335, etc.)
Column 11: “`rm0.max.cpi.days`” (maximum cpi in days^{-1}: 365, 385, etc.)
Column 12: “`num.size.classes`” (number of size classes for sf method: 3, 13, etc.)
Column 13: “`rm0.p.b`” (annual P/B: 5, 10, etc.)
Column 14: “`Growth.equation`” (growth equation: included only for user’s reference)
Column 15: “`min.growth`” (minimum growth rate if growth equation ≤ 0: 0.005, etc.)
Column 16: “`notes`” (notes: included only for user’s reference)

As with the abundance data, this data needs to be saved as a tab-delimited text file, and can have no empty cells. Zeroes should be filled in as “`0`”, and missing data or values that are not applicable for a particular method of production should be coded as “`NA`”. See the section above about maintaining consistently-spelled taxa names in this data.

`get.dates`

Description

Find the sampling dates for the specified site and habitat.

Usage

```
  get.dates(DATA, site, habitat, first.date, last.date)
```

Arguments

`DATA`	The data (as a data frame)
`site`	Site name (entered in quotes, as it appears in the data)
`habitat`	Habitat (entered in quotes, as it appears in the data)
`first.date`	(Optional) first date to include if specifying a range of dates (entered in quotes, as it appears in the data)
`last.date`	(Optional) last date to include if specifying a range of dates (entered in quotes, as it appears in the data)

Details

This function returns all sampling dates that occur in the specified data frame for a given site and habitat. If a date range is specified, the function returns only those sampling dates bounded by the first and last date provided.

Value

A data frame containing the following two variables (columns) in the order listed.

`DATE`	Sample date in the “mm/dd/yyyy” format.
`JULIAN`	Sample date in the Julian date format.

Example

```
#This is an example of a function call that uses
#the data “EX”:

#See all the sampling dates for the specified site and habitat:
get.dates(DATA=EX, site="RM30", habitat="COBBLE")

#See the sampling dates bounded by the specified range
#for the specified site and habitat:
get.dates(  DATA=EX, site="RM30", habitat="COBBLE",
            first.date="9/15/2007", last.date="9/14/2008")
```

```
igr.prod
```

Description

Calculate secondary production and bootstrap confidence intervals using the instantaneous growth rate (igr) method.

Usage

```
igr.prod(DATA, site, first.date, last.date, habitat, taxon, TEMP, wrap,
         LM.a, LM.b, LM.p.ash, g.a, g.b, g.c, g.d, min.growth, temp.corr,
         boot.num)
```

Arguments

`DATA` The data (as a data frame)

`site` Site name (entered in quotes, as it appears in the data)

`first.date` First date to include in the production calculation (entered in quotes, as it appears in the data)

`last.date` Last date to include in the production calculation (entered in quotes, as it appears in the data)

`habitat` Habitat to calculate production for (entered in quotes, as it appears in the data)

`taxon` Taxon to calculate production for (entered in quotes, as it appears in the data)

`TEMP` The temperature data (as a vector of appropriate length: one value for each interval). If an additional "wrap" interval (see below) is to be used, then the vector of temperatures needs to have a value for this additional interval too.

`wrap` (T or F) Specifies whether or not to create an additional interval of duration such that all intervals sum to 365 days. This is useful if the available samples fall short of encompassing an entire year. If `wrap=T`, the mean interval abundance and biomass of the first and last sampling intervals is used to calculate the interval production for the additional interval. The newly created additional interval is only used in interval and annual production calculations; it is NOT averaged in to estimate annual biomass and annual abundance. Default is `wrap=F`.

`LM.a` (a single value) Length-mass parameter (*a*). Assumes a power function ($M=a\cdot L^b$) to convert length (mm) to mass (mg AFDM).

`LM.b` (a single value) Length-mass parameter (*b*). Assumes a power function ($M=a\cdot L^b$) to convert length (mm) to mass (mg AFDM).

`LM.p.ash` (a single value) Percent ash (specified in %; e.g., 10, not 0.10). Should be 0, unless length-mass parameters are for length (mm) to mass (mg DM).

`g.a` (a single value) First growth rate parameter (*a*) `g.b` (a single value) Second growth rate parameter (*b*). If there is no growth parameter *b* for the specified taxon, enter `NA`.

`g.c` (a single value) Third growth rate parameter (*c*). If there is no growth parameter *c* for the specified taxon, enter `NA`.

`g.d` (a single value) Fourth growth rate parameter (*d*). If there is no growth parameter *d* for the specified taxon, enter `NA`.

`min.growth` (a single value) The minimum growth rate (day^{-1}) to use if the growth equation gives a value ≤ 0. Default is `min.growth=0`.

`temp.corr` The temperature-correction factors to apply to each interval (as a vector of appropriate length: one value for each interval). The correction factor for each interval is multiplied by the growth rate(s) for that same interval. This is done to temperature-correct growth equations that were constructed based on field trials conducted only at River Mile 0. Note that the growth equation for chironomids will not be corrected, since temperature dependence is already explicitly accounted for in that taxon's growth equation. To specify no temperature-correction, set all the values in this vector to 1.

`boot.num` (a single value) The number of bootstrap resampling iterations to conduct.

Details

The function automatically sets to zero any negative interval production values.

The "annual" production is calculated for the total number of days encompassed by all the intervals (including an additional "wrap-around" interval if `wrap=T`). Since this rarely comes to exactly 365 days (unless `wrap=T`), the user may wish to scale "annual" production to exactly one year by manually multiplying the desired production value(s) by (365/totdays).

Growth equations for the following taxa are built into the igr.prod function:

`Chironomid`	(growth in day^{-1}) = $a - b\cdot\ln$(length in mm) + $c\cdot$temperature in °C
`Gammarus`	(growth in day^{-1}) = $a\cdot\ln$(length in mm) + b
`Olig A`	(growth in day^{-1}) = $a\cdot\ln$(length in mm) + b
`NZMS`	(growth in day^{-1}) = $a\cdot$length in mm + b
`Olig B`	(growth in day^{-1}) = a

Attempting to run the function on taxa other than these will yield an error message and zeroes for all interval and annual production estimates. The function will still perform abundance and biomass calculations. The names of these taxa must be spelled exactly as above, both in the data itself and in the function call. More specific taxonomic names for these are:

Olig A = Lumbricidae
NZMS = New Zealand mudsnail (*Potamopyrgus antipodarum*)
Olig B = Tubificida (A)

Value

A list containing the following elements (in the order listed). Production values have units of (mg AFDM·m^{-2}·totdays^{-1}), biomass (mg AFDM·m^{-2}), and abundance (individuals·m^{-2}).

[1] totdays — Sum of interval durations (ideally should be close to 365)

[2] Psampest — Measured (i.e., sample) “annual” production value

[3] Pbootest — Mean, median, and 95% CI’s of bootstrapped “annual” production

[4] Pboots — Bootstrap vector of “annual” production estimates

[5] Bboots — Bootstrap vector of “annual” biomass estimates

[6] Nboots — Bootstrap vector of “annual” abundance estimates

[7] julians — Vector of Julian dates that are the sample dates

[8] intdays — Vector of interval durations

[9] Pintsampest — Measured (i.e., sample) interval production for each interval

[10] Pintboots — Matrix of bootstrap interval production estimates for each interval (each column represents an interval and each row contains bootstrap estimates for the intervals)

[11] Bdatesampinfo — Data frame of sample sizes, means, and standard deviations for biomass of the samples collected on each sampling date

[12] Ndatesampinfo — Data frame of sample sizes, means, and standard deviations for abundance of the samples collected on each sampling date

[13] Bdateboots — Matrix of bootstrap “monthly” biomass estimates for each sample date (each column represents a sample date and each row contains bootstrap estimates for the dates)

[14] Ndateboots — Matrix of bootstrap “monthly” abundance estimates for each sample date (each column represents a sample date and each row contains bootstrap estimates for the dates)

Example

```
#This is an example of a function call that uses
#the data “EX” and the temperature data “temp”:

igr.prod(   DATA=EX,
            site="RM30", first.date="9/15/2007", last.date="9/14/2008",
            habitat="COBBLE", taxon="Chironomid",
            TEMP=temp, wrap=F,
            LM.a=0.0006, LM.b=2.77, LM.p.ash=0,
            g.a=0.051, g.b=0.068, g.c=0.006, g.d=NA, min.growth=0,
            temp.corr=c(1,1,1,1),
            boot.num=1000)
```

`sf.prod`

Description

Calculate secondary production and bootstrap confidence intervals using the size-frequency (sf) method.

Usage

```
sf.prod(DATA, site, first.date, last.date, habitat, taxon, LM.a, LM.b,
        LM.p.ash, num.size.classes, min.cpi, max.cpi, temp.corr,
        boot.num)
```

Arguments

`DATA` The data (as a data frame)

`site` Site name (entered in quotes, as it appears in the data)

`first.date` First date to include in the production calculation (entered in quotes, as it appears in the data)

`last.date` Last date to include in the production calculation (entered in quotes, as it appears in the data)

`habitat` Habitat to calculate production for (entered in quotes, as it appears in the data)

`taxon` Taxon to calculate production for (entered in quotes, as it appears in the data)

`LM.a` (a single value) Length-mass parameter (*a*). Assumes a power function ($M=a\cdot L^b$) to convert length (mm) to mass (mg AFDM).

`LM.b` (a single value) Length-mass parameter (*b*). Assumes a power function ($M=a\cdot L^b$) to convert length (mm) to mass (mg AFDM).

`LM.p.ash` (a single value) Percent ash (specified in %; e.g., 10, not 0.10). Should be 0, unless length-mass parameters are for length (mm) to mass (mg DM).

`num.size.classes` (a single value) The number of size classes (usually determined by the maximum body size for a given site and year). Note that the Food Base data also includes a 0.5mm length size class, which should be accounted for when specifying the number of size classes. For more information, see Benke and Huryn (2007).

`min.cpi` (a single value) The minimum cohort production interval (cpi, days).

`max.cpi` (a single value) The maximum cohort production interval (cpi, days).

`temp.corr` (a single value) The temperature-correction factor to apply to the cpi value. The cpi value is divided by the correction factor to temperature-correct cpi's that may have been determined only for River Mile 0. To specify no temperature-correction, set to 1 (the default).

`boot.num` (a single value) The number of bootstrap resampling iterations to conduct.

Details

Calculations are performed following the suggestions in Benke and Huryn (2007). Specifically, the function uses the geometric mean of individual masses between size classes, but uses the individual mass of the largest size class for the "final" transition out of the largest size class. In addition, negative values in the (biomass·number of size classes, mg AFDM m^{-2}) column are set to zero only if they occur below a non-positive value (i.e., negative or zero). Production values are "corrected" using the cohort production interval (cpi).

A cpi value is randomly drawn from the range specified by `min.cpi` and `max.cpi` at each bootstrap iteration to be used in the production calculation for that iteration. To use a single cpi value for all bootstrap iterations, set `min.cpi` and `max.cpi` to the same value. When calculating measured (i.e., sample) production, the function always uses the mean of `min.cpi` and `max.cpi`, to prevent stochasticity in the sample production value.

Value

A list containing the following elements (in the order listed). Production values have units of (mg AFDM·m^{-2}·yr^{-1}), biomass (mg AFDM·m^{-2}), and abundance (individuals·m^{-2}). The production values returned have been "corrected" using the cohort production interval (cpi, Benke and Huryn 2007).

[1] `totdays` Sum of interval durations (ideally should be close to 365)

[2] `cpi` Bootstrap vector of cpi values used at each iteration

[3] `Psampest` Measured (i.e., sample) "annual" production value

[4] `Pbootest` Mean, median, and 95% CI's of bootstrapped "annual" production

[5] `Pboots` Bootstrap vector of "annual" production estimates

[6] `Bboots` Bootstrap vector of "annual" biomass estimates

[7] `Nboots` Bootstrap vector of "annual" abundance estimates

[8] `julians` Vector of Julian dates that are the sample dates

[9] `intdays` Vector of interval durations

[10] `Bdatesampinfo` Data frame of sample sizes, means, and standard deviations for biomass of the samples collected on each sampling date

[11] `Ndatesampinfo` Data frame of sample sizes, means, and standard deviations for abundance of the samples collected on each sampling date

[12] `Bdateboots` Matrix of bootstrap "monthly" biomass estimates for each sample date (each column represents a sample date and each row contains bootstrap estimates for the dates)

[13] `Ndateboots` Matrix of bootstrap "monthly" abundance estimates for each sample date (each column represents a sample date and each row contains bootstrap estimates for the dates)

Example

```
#This is an example of a function call that uses
#the data “EX”:

sf.prod(    DATA=EX,
            site="RM30", first.date="9/15/2007", last.date="9/14/2008",
            habitat="COBBLE", taxon="Simuliid",
            LM.a=0.004, LM.b=2.807, LM.p.ash=0,
            num.size.classes=11, min.cpi=300, max.cpi=365, temp.corr=1,
            boot.num=1000)
```

References

Benke, A.C. and A.D. Huryn 2007. Secondary production of macroinvertebrates. Pages 691-710 in F.R. Hauer and G.A. Lamberti, eds. *Methods in stream ecology*. Academic Press, San Diego, CA, USA.

`pb.prod`

Description

Calculate secondary production and bootstrap confidence intervals using the P/B (pb) method.

Usage

```
pb.prod(DATA, site, first.date, last.date, habitat, taxon, LM.a, LM.b,
        LM.p.ash, p.b, temp.corr, boot.num)
```

Arguments

`DATA` The data (as a data frame)

`site` Site name (entered in quotes, as it appears in the data)

`first.date` First date to include in the production calculation (entered in quotes, as it appears in the data)

`last.date` Last date to include in the production calculation (entered in quotes, as it appears in the data)

`habitat` Habitat to calculate production for (entered in quotes, as it appears in the data)

`taxon` Taxon to calculate production for (entered in quotes, as it appears in the data)

`LM.a` (a single value) Length-mass parameter (*a*). Assumes a power function ($M=a\cdot L^{b}$) to convert length (mm) to mass (mg AFDM).

`LM.b` (a single value) Length-mass parameter (*b*). Assumes a power function ($M=a\cdot L^{b}$) to convert length (mm) to mass (mg AFDM).

`LM.p.ash` (a single value) Percent ash (specified in %; e.g., 10, not 0.10). Should be 0, unless length-mass parameters are for length (mm) to mass (mg DM).

`p.b` (a single value) The annual production to biomass (P/B) ratio.

`temp.corr` (a single value) The temperature-correction factor to apply to the P/B value. The correction factor is multiplied by the P/B value to temperature-correct P/B's that may have been determined only for River Mile 0. To specify no temperature-correction, set to 1 (the default).

`boot.num` (a single value) The number of bootstrap resampling iterations to conduct.

Details

This function calculates annual production by multiplying the bootstrap vector of "annual" biomass estimates by the annual production to biomass (P/B) ratio specified.

Value

A list containing the following elements (in the order listed). Production values have units of (mg AFDM·m^{-2}·yr^{-1}), biomass (mg AFDM·m^{-2}), and abundance (individuals·m^{-2}).

[1]	`totdays`	Sum of interval durations (ideally should be close to 365)
[2]	`pb`	Annual production to biomass (P/B) ratio
[3]	`Psampest`	Measured (i.e., sample) “annual” production value
[4]	`Pbootest`	Mean, median, and 95% CI’s of bootstrapped “annual” production
[5]	`Pboots`	Bootstrap vector of “annual” production estimates
[6]	`Bboots`	Bootstrap vector of “annual” biomass estimates
[7]	`Nboots`	Bootstrap vector of “annual” abundance estimates
[8]	`julians`	Vector of Julian dates that are the sample dates
[9]	`intdays`	Vector of interval durations
[10]	`Bdatesampinfo`	Data frame of sample sizes, means, and standard deviations for biomass of the samples collected on each sampling date
[11]	`Ndatesampinfo`	Data frame of sample sizes, means, and standard deviations for abundance of the samples collected on each sampling date
[12]	`Bdateboots`	Matrix of bootstrap “monthly” biomass estimates for each sample date (each column represents a sample date and each row contains bootstrap estimates for the dates)
[13]	`Ndateboots`	Matrix of bootstrap “monthly” abundance estimates for each sample date (each column represents a sample date and each row contains bootstrap estimates for the dates)

Example

```
#This is an example of a function call that uses
#the data “EX”:

pb.prod(    DATA=EX,
            site="RM30", first.date="9/15/2007", last.date="9/14/2008",
            habitat="COBBLE", taxon="Mite",
            LM.a=0.00266, LM.b=1, LM.p.ash=0,
            p.b=5, temp.corr=1,
            boot.num=1000)
```

wrapper.site.yr

Description

Calculate secondary production, biomass, abundance, and bootstrap confidence intervals for all taxa and habitats for a given site and year.

Usage

```
wrapper.site.yr(DATA, site, first.date, last.date, habitat, TAXA, TEMP.COB,
          TEMP.DEP, TEMP.TAL, wrap, temp.corr.igr.cob, temp.corr.igr.dep,
          temp.corr.igr.tal, temp.corr.sfpb, boot.num)
```

Arguments

`DATA` The data (as a data frame)

`site` Site name (entered in quotes, as it appears in the data)

`first.date` First date to include in the production calculation (entered in quotes, as it appears in the data)

`last.date` Last date to include in the production calculation (entered in quotes, as it appears in the data)

`habitat` Habitat to calculate production for (entered in quotes, as it appears in the data). To calculate production for all habitats, enter "`ALL`".

`TAXA` A data frame containing information needed to calculate production for each taxon.

`TEMP.COB` (used only for igr production calculations) The temperature data for the "`COBBLE`" habitat (as a vector of appropriate length: one value for each interval). If an additional "wrap" interval (see below) is to be used, then the vector of temperatures needs to have a value for this additional interval too.

`TEMP.DEP` Same as above, but for the "`DEPOSITIONAL`" habitat.

`TEMP.TAL` Same as above, but for the "`TALUS.CLIFF`" habitat.

`wrap` (T or F, used only for igr production calculations) Specifies whether or not to create an additional interval of duration such that all intervals sum to 365 days. This is useful if the available samples fall short of encompassing an entire year. If `wrap=T`, the mean interval abundance and biomass of the first and last sampling intervals is used to calculate the interval production for the additional interval. The newly created additional interval is only used in interval and annual production calculations; it is NOT averaged in to estimate annual biomass and annual abundance. Default is `wrap=F`.

`temp.corr.igr.cob` (used only for igr production calculations) The temperature-correction factors to apply to each interval for the "`COBBLE`" habitat (as a vector of appropriate length: one value for each interval). The correction factor for each interval is multiplied by the growth rate(s) for that same interval. This is done to temperature-correct growth equations that were constructed based on field trials conducted only at River Mile 0. Note that the growth equation for chironomids will not be corrected, since temperature dependence is already explicitly accounted for in that taxon's growth equation. To specify no temperature-correction, set all the values in this vector to 1.

`temp.corr.igr.dep` Same as above, but for the "`DEPOSITIONAL`" habitat.

`temp.corr.igr.tal` Same as above, but for the "`TALUS.CLIFF`" habitat.

`temp.corr.sfpb` (a single value, used only for sf and pb production calculations) The temperature-correction factor to apply to the cpi value and P/B value. The cpi value is divided by the correction factor to temperature-correct cpi's that may have been determined only for River Mile 0. The correction factor is multiplied by the P/B value to temperature-correct P/B's that may have been determined only for River Mile 0. To specify no temperature-correction, set to 1 (the default).

`boot.num` (a single value) The number of bootstrap resampling iterations to conduct.

Details

This is a wrapper function that cycles through all the taxa that occur at the specified site on any of the sample dates within the date range specified, and calculates production, biomass, and abundance for each. If `habitat="ALL"`, then production is calculated for each taxon-habitat combination (e.g., for chironomid in COBBLE, for chironomid in DEPOSITIONAL, and for chironomid in TALUS.CLIFF). If just one habitat is specified, then the output list will contain NA's for the other two habitats. Because the habitat types "`COBBLE`", "`DEPOSITIONAL`", and "`TALUS.CLIFF`" are built into this function, the spellings of these must remain consistent in the data, and in the function call.

The function reads the name of a taxon in the data, and then looks for that taxon in the specified data frame (`TAXA`), where it uses the method of production listed to call one of `igr.prod`, `sf.prod`, or `pb.prod` to calculate production for that taxon. The appropriate production function uses the length-mass parameters, growth parameters, minimum growth values, cpi values, number of size classes, and P/B values listed for that taxon in the `TAXA` data frame. Because taxa names listed in the `TAXA` data frame are matched against taxa names in the `DATA` data frame, spellings must remain consistent between the two data frames.

If a taxon does not appear in the `TAXA` data frame or if no method of production is listed for a taxon in the `TAXA` data frame, an error message is printed, and NA values are assigned for that taxon's annual bootstrap production estimates.

Value

A list containing the following nine elements (in the order listed). Production values have units of (mg AFDM·m^{-2}·yr^{-1}) for taxa that use the size-frequency and P/B methods, and (mg AFDM·m^{-2}·totdays^{-1}) for taxa that use the instantaneous growth rate method. Biomass values have units of (mg AFDM·m^{-2}) and abundance values have units of (individuals·m^{-2}).

[1] `Pboots.cob` Data frame of bootstrap “annual” production estimates for each taxon for the “`COBBLE`” habitat.

[2] `Pboots.dep` Data frame of bootstrap “annual” production estimates for each taxon for the “`DEPOSITIONAL`” habitat.

[3] `Pboots.tal` Data frame of bootstrap “annual” production estimates for each taxon for the “`TALUS.CLIFF`” habitat.

[4] `Bboots.cob` Data frame of bootstrap “annual” biomass estimates for each taxon for the “`COBBLE`” habitat.

[5] `Bboots.dep` Data frame of bootstrap “annual” biomass estimates for each taxon for the “`DEPOSITIONAL`” habitat.

[6] `Bboots.tal` Data frame of bootstrap “annual” biomass estimates for each taxon for the “`TALUS.CLIFF`” habitat.

[7] `Nboots.cob` Data frame of bootstrap “annual” abundance estimates for each taxon for the “`COBBLE`” habitat.

[8] `Nboots.dep` Data frame of bootstrap “annual” abundance estimates for each taxon for the “`DEPOSITIONAL`” habitat.

[9] `Nboots.tal` Data frame of bootstrap “annual” abundance estimates for each taxon for the “`TALUS.CLIFF`” habitat.

The function also writes a text file of the full output for each taxon-habitat combination to the “Output” folder.

Example

```
#This is an example of a function call that uses the data “EX”,
#the taxonomic information “TAXA”, and the temperature data “temp”:

wrapper.site.yr(  DATA=EX,
                  site="RM30", first.date="9/15/2007", last.date="9/14/2008",
                  habitat="ALL", TAXA=tax,
                  TEMP.COB=temp, TEMP.DEP=temp, TEMP.TAL=temp, wrap=F,
                  temp.corr.igr.cob=c(1,1,1,1), temp.corr.igr.dep=c(1,1,1,1),
                  temp.corr.igr.tal=c(1,1,1,1), temp.corr.sfpb=1,
                  boot.num=100)
```

hab.weight

Description

Calculate a habitat-weighted estimate of production (or biomass or abundance).

Usage

```
hab.weight(dat.cob, dat.dep, dat.tal, wt.cob, wt.dep, wt.tal)
```

Arguments

`dat.cob` A vector or an array of production, biomass, or abundance estimates for the “COBBLE” habitat.

`dat.dep` A vector or an array of production, biomass, or abundance estimates for the “DEPOSITIONAL” habitat.

`dat.tal` A vector or an array of production, biomass, or abundance estimates for the “TALUS.CLIFF” habitat.

`wt.cob` A single value representing the “weight” to apply to the “COBBLE” estimates.

`wt.dep` A single value representing the “weight” to apply to the “DEPOSITIONAL” estimates.

`wt.tal` A single value representing the “weight” to apply to the “TALUS.CLIFF” estimates.

Details

The function uses the supplied weights to calculate the weighted average of corresponding values in the “COBBLE”, “DEPOSITIONAL”, and “TALUS.CLIFF” vectors or arrays. Note that production/biomass/abundance estimates can be supplied either as vectors or as arrays (data frames). If the latter, the array dimensions must match.

Value

A vector or array (depending on what was supplied) of the habitat-weighted estimates of production, biomass, or abundance.

Example

```
#This is an example of a function call that uses
#the data “EX”:

#An example with bootstrapped vectors of biomass:
#First generate bootstrap estimates of biomass in the three habitats:
out.cob <- pb.prod(      DATA=EX,
                         site="RM30", first.date="9/15/2007",
                         last.date="9/14/2008", habitat="COBBLE",
                        taxon="Mite", LM.a=0.00266, LM.b=1, LM.p.ash=0,
                        p.b=5, temp.corr=1,
                        boot.num=1000)

out.dep <- pb.prod(     DATA=EX,
                        site="RM30", first.date="9/15/2007",
                        last.date="9/14/2008", habitat="DEPOSITIONAL",
                        taxon="Mite", LM.a=0.00266, LM.b=1, LM.p.ash=0,
                        p.b=5, temp.corr=1,
                        boot.num=1000)

out.tal <- pb.prod(     DATA=EX,
                        site="RM30", first.date="9/15/2007",
                        last.date="9/14/2008", habitat="TALUS.CLIFF",
                        taxon="Mite", LM.a=0.00266, LM.b=1, LM.p.ash=0,
                        p.b=5, temp.corr=1,
                        boot.num=1000)

#Then calculate habitat-weighted biomass:
hab.weight( dat.cob=out.cob$Bboots, dat.dep=out.dep$Bboots,
            dat.tal=out.tal$Bboots, wt.cob=0.4, wt.dep=0.4, wt.tal=0.2)


#An example with bootstrapped arrays of production:
#First generate bootstrap estimates of production in the three habitats:
out <- wrapper.site.yr( DATA=EX, site="RM30", first.date="9/15/2007",
                        last.date="9/14/2008", habitat="ALL", TAXA=tax,
                        TEMP.COB=temp, TEMP.DEP=temp, TEMP.TAL=temp, wrap=F,
                        temp.corr.igr.cob=c(1,1,1,1),
                        temp.corr.igr.dep=c(1,1,1,1),
                        temp.corr.igr.tal=c(1,1,1,1), temp.corr.sfpb=1,
                        boot.num=100)

#Then calculate habitat-weighted production for all taxa:
hab.weight( dat.cob=out$Pboots.cob, dat.dep=out$Pboots.dep,
            dat.tal=out$Pboots.tal, wt.cob=0.4, wt.dep=0.4, wt.tal=0.2)
```

`sampinfo.site.yr`

Description

Calculate mean and standard deviation of sample biomass and abundance for each date for all taxa and habitats for a given site and year.

Usage

```
sampinfo.site.yr(DATA, site, first.date, last.date, habitat, TAXA)
```

Arguments

`DATA`	The data (as a data frame)
`site`	Site name (entered in quotes, as it appears in the data)
`first.date`	First date to include in the calculation (entered in quotes, as it appears in the data)
`last.date`	Last date to include in the calculation (entered in quotes, as it appears in the data)
`habitat`	Habitat to run calculations for (entered in quotes, as it appears in the data). To calculate production for all habitats, enter "`ALL`".
`TAXA`	A data frame containing information needed to calculate biomass for each taxon.

Details

This function cycles through all the taxa that occur at the specified site on any of the sample dates within the date range specified, and calculates the mean and standard deviation of biomass and abundance of all samples collected on each date for each taxon. If `habitat="ALL"`, then the calculations are performed for each taxon-habitat combination (e.g., for chironomid in COBBLE, for chironomid in DEPOSITIONAL, and for chironomid in TALUS.CLIFF). If just one habitat is specified, then the output data frame will only contain results for that habitat. Because the habitat types "`COBBLE`", "`DEPOSITIONAL`", and "`TALUS.CLIFF`" are built into this function, the spellings of these must remain consistent in the data, and in the function call.

The function reads the name of a taxon in the data, and then looks for that taxon in the specified data frame (`TAXA`), where it uses the length-mass parameters listed (`LM.a, LM.b, LM.p.ash`) to calculate mean and standard deviation biomass for that taxon on each date. Abundance (mean and standard deviation) is also calculated. The `TAXA` data frame can be of the same as that required by the `wrapper.site.yr` function. Because taxa names listed in the `TAXA` data frame are matched against taxa names in the `DATA` data frame, spellings must remain consistent between the two data frames.

If a taxon does not appear in the TAXA data frame or if not all length-mass parameters are listed for a taxon in the TAXA data frame, an error message is printed, and NA values are assigned for that taxon's mean and standard deviation biomass values.

Value

A data frame containing the following nine columns (in the order listed). Biomass values have units of (mg AFDM·m^{-2}) and abundance values have units of (individuals·m^{-2}).

[1] HABITAT Habitat (“COBBLE”, “DEPOSITIONAL”, or “TALUS.CLIFF”.

[2] TAXON Taxon.

[3] DATE Date.

[4] JULIAN Julian date corresponding to DATE.

[5] N The number of samples collected in that habitat on that date.

[6] MEAN_B The mean for biomass of all samples collected for that taxon in that habitat on that date.

[7] Stdev_B The standard deviation for biomass of all samples collected for that taxon in that habitat on that date.

[8] MEAN_N The mean for abundance of all samples collected for that taxon in that habitat on that date.

[9] Stdev_N The standard deviation for abundance of all samples collected for that taxon in that habitat on that date.

Example

```
##This is an example of a function call that uses the data “EX”,
#and the taxonomic information “TAXA:

sampinfo.site.yr( DATA=EX,
                  site="RM30", first.date="9/15/2007", last.date="9/14/2008",
                  habitat="ALL", TAXA=tax)
```