

```

# -----
# Appendix 3: Functions added to the multifunc package
# hosted at https://jebyrnes.github.io/multifunc/ for
# the calculation of effective multifunctionality from
# Byrnes, Roger, and Bagchi's Understandable
# Multifunctionality Measures Using Hill Numbers
# -----

#' @title getMF_eff
#' @description A multifunctionality index rooted in Hill numbers.
#' \code{getMF_eff} get multifunctionality index defined by function and
effective number of functions
#'
#' @details Takes a data frame, variable names, a standardizing function,
whether we want
#' an index standardized by number of functions or not, an order of Hill
number for our
#' effective number of functions as well as a dissimilarity matrix (if
desired) and value
#' for a dissimilarity cutoff (defaults to the average dissimilarity). It
then calculates
#' both the average standardized function in each plot and the effective
number of
#' functions and returns their product as a measure of effective
multifunctionality.
#'
#' @author Jarrett Byrnes.
#' @param data A data frame with functions in columns and rows as replicates
as well as other information.
#' @param vars Name of function variables
#' @param standardized Use standardized number of functions (scaled by total
number of functions, so between 0–1), or just raw effective number of
#' functions for calculation. Defaults to \code{FALSE}.
#' @param standardize_function A function to standardize each individual
function to the same scale, such as \code{standardizeUnitScale} or
#' \code{standardizeZScore}
#' @param q Order of the diversity measure. Defaults to the
Shannon case where  $q = 1$ . For Simpson,  $q=2$ .
#' @param D A distance matrix describing dissimilarity between functions.
Defaults
#' to NULL, and the index is calculated assuming all functions are different.
If
#' it is not null, it must be a symmetric matrix with dimensions matching the
number of functions listed in \code{vars}.
#' @param tau A cutoff for degree of dissimilarity under which functions are
considered
#' to be different. If tau is the minimum non-zero value of D, all functions
are different.
#' if tau is the maximum value of D are greater, all functions are considered
the same.
#' @references
#'
#' Chao, A., Chiu, C.-H., Villéger, S., Sun, I.-F., Thorn, S., Lin, Y.-C.,
#' Chiang, J.-M. and Sherwin, W. B. 2019. An attribute–diversity approach to

```

```

#' functional diversity, functional beta diversity, and related
#(dis)similarity
#' measures. Ecological Monographs. 89: e01343.
#'
#' Jost, L. 2006. Entropy and diversity. Oikos 113(2): 363–375.
#'
#' Hill, M. 1973. Diversity and evenness: A unifying notation and its
#' consequences. Ecology 54: 427–432.
#'
#' @export
#' @return Returns a vector of effective or standardized effective
multifunctionality.
#'

getMF_eff <- function(data, vars, q = 1,
                      standardized = FALSE,
                      standardize_function = standardizeUnitScale,
                      D = NULL, tau = NULL) {

  # get standardized functions
  std_funcs <- getStdAndMeanFunctions(data, vars, standardize_function)
  funcs <- std_funcs[, -ncol(std_funcs)]

  # get the average functional level
  mf_a <- std_funcs$meanFunction

  # get evenness
  eff_func <- eff_num_func(
    dat = funcs,
    vars = names(funcs),
    q = q,
    standardized = standardized,
    D = D, tau = tau
  )

  # mf = average * eff num functions
  mf_a * eff_func
}

#' @title eff_num_func
#'
#' @description Calculate the effective number of functions for rows in a
dataset
#'
#' @param dat A data frame with functions in columns and rows as replicates
as well as other information.
#' @param vars Column names of function variables
#' @param standardized Use standardized number of functions (scaled by total
number of functions, so between 0–1), or just raw effective number of
functions for calculation. Defaults to FALSE.
#' @param q Order of the diversity measure. Defaults to the
Shannon case where  $q = 1$ . For Simpson,  $q=2$ .

```

```

#' @param D A distance matrix describing dissimilarity between functions.
Defaults
#' to NULL, and the index is calculated assuming all functions are different.
If
#' it is not null, it must be a symmetric matrix with dimensions matching the
#' number of functions listed in \code{vars}.
#' @param tau A cutoff for degree of dissimilarity under which functions are
considered
#' to be different. If tau is the minimum non-zero value of D, all functions
are different.
#' if tau is the maximum value of D are greater, all functions are considered
the same.
#'
##' @details Takes a data frame, variable names, whether we want
#' an index standardized by number of functions or not, an order of Hill
number for our
#' effective number of functions as well as a dissimilarity matrix (if
desired) and value
#' for a dissimilarity cutoff (defaults to the average dissimilarity). It
then calculates
#' and returns the effective number of functions using the appropriate
method. See Chao
#' et al. 2019 for more.
#'
#'
#' @references
#'
#' Chao, A., Chiu, C.-H., Villéger, S., Sun, I.-F., Thorn, S., Lin, Y.-C.,
#' Chiang, J.-M. and Sherwin, W. B. 2019. An attribute-diversity approach to
#' functional diversity, functional beta diversity, and related
(dis)similarity
#' measures. Ecological Monographs. 89: e01343.
#'
#' Jost, L. 2006. Entropy and diversity. Oikos 113(2): 363–375.
#'
#' Hill, M. 1973. Diversity and evenness: A unifying notation and its
#' consequences. Ecology 54: 427–432.
#'
#'
#' @return Returns a vector of effective or standardized effective number of
functions
#' @export
#'

eff_num_func <- function(dat, vars, q = 1,
                        standardized = FALSE,
                        D = NULL, tau = NULL) {

  # get the variable data frame
  adf_raw <- dat[, which(names(dat) %in% vars)]

  # calculate tau if needed
  if (!is.null(D) & is.null(tau)) tau <- dmean(adf_raw, D)

```

```

# get relative levels of each function to all others per row
adf_freq <- adf_raw / rowSums(adf_raw)

# proportion of community
if (is.null(D)) {
  ret <- eff_num_func_no_d(adf_freq, q = q)
} else {
  ret <- eff_num_func_d(adf_freq, q = q, D = D, tau = tau)
}

# standardize if needed
if (standardized) {
  ret <- ret / length(vars)
}

return(ret)
}

# effective # functions not adjusting for correlation
#' @title eff_num_func_no_d
#'
#' @param adf_freq A data frame of functional "frequencies" - i.e. f_i/
sum(f_i)
#' @param q Order of hill number used for index. Defaults to q=1, as in
Shannon Diversity
#' @details Takes a data frame, with functions standardized against total
level of function
#' in their replicate as columns and replicates as rows. Returns the
effective number of functions using the appropriate method. See Chao
#' et al. 2019 or Jost 2006 for details. Does not adjust for correlation
between functions.
#' @references
#'
#' Chao, A., Chiu, C.-H., Villéger, S., Sun, I.-F., Thorn, S., Lin, Y.-C.,
#' Chiang, J.-M. and Sherwin, W. B. 2019. An attribute-diversity approach to
#' functional diversity, functional beta diversity, and related
(dis)similarity
#' measures. Ecological Monographs. 89: e01343.
#'
#' Jost, L. 2006. Entropy and diversity. Oikos 113(2): 363-375.
#'
#' Hill, M. 1973. Diversity and evenness: A unifying notation and its
#' consequences. Ecology 54: 427-432.
#'
#' @return Returns a vector of effective or standardized effective number of
functions
#' @export
#'

eff_num_func_no_d <- function(adf_freq, q = 1) {
  if (q == 1) {
    summand <- adf_freq * log(adf_freq)
    # deal with 0 log 0 = 0
    zero_idx <- which(is.nan(as.matrix(summand))), arr.ind = TRUE)
  }

```

```

    if (length(zero_idx) > 0) {
      summand[which(is.nan(as.matrix(summand)), arr.ind = TRUE)] <- 0
    }

    exp(-rowSums(summand))
  } else {
    rowSums(adf_freq^q)^(1 / (1 - q))
  }
}

# effective # functions adjusting for correlation
#' eff_num_func_d
#'
#' @param adf_freq A data frame of functional "frequencies" - i.e. f_i/
sum(f_i)
#' @param q Order of hill number used for index. Defaults to q=1, as in
Shannon Diversity
#' @param D A distance matrix describing dissimilarity between functions.
#' @param tau A cutoff for degree of dissimilarity under which functions are
considered
#' to be different. If tau is the minimum non-zero value of D, all functions
are different.
#' if tau is the maximum value of D are greater, all functions are considered
the same.
#'
#' @references
#'
#' Chao, A., Chiu, C.-H., Villéger, S., Sun, I.-F., Thorn, S., Lin, Y.-C.,
#' Chiang, J.-M. and Sherwin, W. B. 2019. An attribute-diversity approach to
#' functional diversity, functional beta diversity, and related
(dis)similarity
#' measures. Ecological Monographs. 89: e01343.
#'
#'
#' @return A vector of effective number of functions
#' @export
#'

eff_num_func_d <- function(adf_freq, q = 1, D, tau = NULL) {
  # make sure we can do this
  check_d_and_adf_errors(adf_freq, D)

  # should have supplied one! cannot calculate
  if (is.null(tau)) stop("You need to choose a value for tau.")
  if (tau == 0) {
    warning("Tau cannot be 0, as this would lead to division by 0. Changing
to the lowest value > 0")
    tau <- min(D[D != 0])
  }
  # make sure it's a matrix!
  D <- as.matrix(D)

  # truncate by tau
  D[which(D > tau, arr.ind = TRUE)] <- tau

```

```

# iterate over rows
apply(adf_freq, 1, eff_num_func_d_onerow, D = D, tau = tau, q = q)
}

#' eff_num_func_d_onerow
#'
#' @param arow_freq One replicate sample of different functions (a single
numeric vector)
#' @param q Order of hill number used for index. Defaults to q=1, as in
Shannon Diversity
#' @param D A distance matrix describing dissimilarity between functions.
#' @param tau A cutoff for degree of dissimilarity under which functions are
considered
#' to be different. If tau is the minimum non-zero value of D, all functions
are different.
#' if tau is the maximum value of D are greater, all functions are considered
the same.
#' @references
#'
#' Chao, A., Chiu, C.-H., Villéger, S., Sun, I.-F., Thorn, S., Lin, Y.-C.,
#' Chiang, J.-M. and Sherwin, W. B. 2019. An attribute-diversity approach to
#' functional diversity, functional beta diversity, and related
(dis)similarity
#' measures. Ecological Monographs. 89: e01343.
#'
#'
#' @return A single value of effective number of functions
#' @export
#'

eff_num_func_d_onerow <- function(arow_freq, D, tau, q) {
  arow_freq <- as.numeric(arow_freq)
  a <- as.vector((1 - D / tau) %**% arow_freq) # from Chao et al 2019 code

  if (q == 1) {
    exp(-1 * sum(arow_freq * log(a)))
  } else {
    sum(arow_freq * a^(q - 1))^(1 / (1 - q))
  }
}

#' @title dmean
#' @description Calculates the average distance between functions for one or
an entire
#' assemblage of replicates
#'
#' @param adf_raw A data frame frame with functions in columns and rows as
replicates
#' @param D A distance matrix describing dissimilarity between functions.
#' @references
#'

```

```

#' Chao, A., Chiu, C.-H., Villéger, S., Sun, I.-F., Thorn, S., Lin, Y.-C.,
#' Chiang, J.-M. and Sherwin, W. B. 2019. An attribute-diversity approach to
#' functional diversity, functional beta diversity, and related
#' (dis)similarity
#' measures. Ecological Monographs. 89: e01343.
#'
#' @return Single numeric of weighted average of distance matrix
#' @export
#'

dmean <- function(adf_raw, D) {
  # make sure we can do this
  check_d_and_adf_errors(adf_raw, D)

  # sum ( (tmp %**% t(tmp) ) * dij)
  p_vec <- colSums(adf_raw) / sum(adf_raw)

  # Q <- 0
  #
  # for(i in 1:ncol(D)){
  #   for(j in 1:ncol(D)){
  #     Q <- Q + D[i,j]*p_vec[i]*p_vec[j]
  #   }
  # }

  # from Chao et al. 2019 code - matches above
  sum(p_vec %**% t(p_vec) * D)
}

#' dmin
#'
#' @description Calculates the minimum non-zero value of a distance matrix
#'
#' @param D A distance matrix describing dissimilarity between functions.
#'
#' @return A numeric
#' @export
#'

dmin <- function(D) {
  min(D[D != 0])
}

#' @title cor_dist
#' @description Takes a data frame of functions and calculates the
#' correlation-based
#' distance between functions.
#'
#' @param adf A \code{data.frame} or \code{matrix} of functions
#'
#' @return A matrix
#' @export
#'

```

```

cor_dist <- function(adf) {
  if (sum(is.na(adf)) > 0) stop("Some function values are NA. Cannot compute
a distance matrix.")
  cmat <- stats::cor(adf)
  D <- (1 - cmat) / 2
  D
}

## Error checks for using distance-matrix based effective multifunctionality
## that could be needed for multiple different functions
check_d_and_adf_errors <- function(adf, D) {
  # error checks
  if (sum(is.na(adf)) > 0) stop("Some function values are NA. Cannot compute
multifunctionality for those rows. Please filter them out before running.")

  if (nrow(D) != ncol(D)) {
    stop("Distance matrix is not square")
  }

  if (!isSymmetric(D)) {
    stop("Distance matrix is not symmetric")
  }

  if (nrow(D) != ncol(adf)) {
    stop("Distance matrix does not have same number of rows as number of
variables")
  }

  if (ncol(D) != ncol(adf)) {
    stop("Distance matrix does not have same number of columns as number of
variables")
  }
}

```