

MIIN Part 2: calculate CWMs

Marissa Lee

June 1, 2015

Filename: MIIN_2_cwm.Rmd

This markdown file does the following tasks: 1. Prep trait data: a) Convert TRY trait data to standardized units, b) Aggregate TRY trait data by paper species and genus, c) Create a traits-by-spID matrix

2. Prep cover data: a) Estimate individual species cover values based on cumulative species cover values, b) Create a cover-by-spID matrix
3. Calculate community-weighted means (CWM): a) Merge the trait and cover matrices, b) eliminate observations that do not meet the minimum criteria needed to calculate CWM value, c) calculate species' relative abundances, d) calculate cwm values for each trait type
4. Annotate how each CWM value varies in quality
5. Clean
6. Export to DATA_SYNTHESIZED/cwm folder

```
#knitr::opts_chunk$set(cache=TRUE)
```

```
#library(ggplot2)
#library(gridExtra)
#library(reshape2)
```

```
source('CODE/mytheme.R')
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 3.2.4
```

```
## Loading required package: grid
```

```
figuresPath<-file.path(getwd()[1], "FIGURES_TABLES", "cwm") #where to put the saved plots
fig.height<-2.5 #inches
fig.width<- 2.5 #inches
fig.res<-300
```

```
synthdataPath<-file.path(getwd()[1], "DATA", "DATA_SYNTHESIZED", "cwm") #where to put the clean dataframes
```

```
#data synthesized by *MIIN_1_paperData.Rmd*
```

```
observations<-read.table("DATA/DATA_SYNTHESIZED/paperData/observations_procd.txt", header=TRUE, sep="\t")
cover<-read.table("DATA/DATA_SYNTHESIZED/paperData/cover_procd.txt", header=TRUE, sep="\t")
species<-read.table("DATA/DATA_SYNTHESIZED/paperData/species_procd.txt", header=TRUE, sep="\t")
traits<-read.table("DATA/DATA_SYNTHESIZED/paperData/traits_procd.txt", header=TRUE, sep="\t")
measures<-read.table("DATA/DATA_SYNTHESIZED/paperData/measures_procd.txt", header=TRUE, sep="\t")
```

```
#data synthesized by *MIIN_1_tryData.Rmd*
```

```
tryDataT<-read.table("DATA/DATA_SYNTHESIZED/tryData/tryDataT.txt", header=TRUE, sep="\t")
tryData_traitKey<-read.table("DATA/DATA_SYNTHESIZED/tryData/tryData_traitKey.txt", header=TRUE, sep="\t")
```

1. PREP TRAIT DATA

1A. Convert traits within *TRY* data to common units selected for the analogous paper traits.

```
source('CODE/cwm/script_TRYstdUnits.R') # TASK= Convert TRY data to the common units used for paper tra
#tryDataT.cu[1:10,c('StdValue','stdmean','Unit_UnitName','stdunit')]
```

1B. Aggregate *TRY* data by paper species and genus. Average multiple entries for the same trait and species/genus.

```
source('CODE/cwm/script_traitBySpecies.R') # TASK= Pulls relavant TRY data and creates a species x trait
#Don't worry about these warning messages
#nGS<-ddply(tryGS.summ, ~AccGenusSpecies+traitOI, summarise,n=length(AccGenusSpecies))
#max(nGS$n) #ok, no species x traitCats repeated
#nGX<-ddply(tryGX.summ, ~AccGenus+traitOI, summarise,n=length(AccGenus))
#max(nGX$n) #ok, no species x traitCats repeated

#save the rows of TRY data that were used so that I can acknowledge data authors
usedTry<-merge(tryGS,tryGX, all=TRUE)
newfilename<-'usedTryData.txt'
write.table(usedTry, file=paste(synthdataPath,newfilename, sep='/'), sep='\t')

contribTab<-ddply(usedTry, ~LastName+TraitName, summarize,
  nrows=sum(!is.na(stdmean)))
nrows.total<-sum(contribTab$nrows)
contribTab$perc<- (contribTab$nrows / nrows.total)*100
contribTab
```

##	LastName	TraitName	nrows
## 1	Atkin	Leaf nitrogen (N) content per dry mass	150
## 2	Chapin	Leaf nitrogen (N) content per dry mass	2
## 3	Cornelissen	Leaf nitrogen (N) content per dry mass	215
## 4	Cornelissen	Litter nitrogen (N) content per dry mass	63
## 5	Cornwell	Leaf nitrogen (N) content per dry mass	192
## 6	Craine	Leaf nitrogen (N) content per dry mass	2573
## 7	Enquist	Leaf nitrogen (N) content per dry mass	226
## 8	Fang	Leaf nitrogen (N) content per dry mass	417
## 9	Freschet	Leaf nitrogen (N) content per dry mass	15
## 10	Garnier	Leaf nitrogen (N) content per dry mass	404
## 11	Kattge	Leaf nitrogen (N) content per dry mass	577
## 12	Laughlin	Leaf nitrogen (N) content per dry mass	55
## 13	Louault	Leaf nitrogen (N) content per dry mass	38
## 14	Medlyn	Leaf nitrogen (N) content per dry mass	261
## 15	Medlyn	Litter nitrogen (N) content per dry mass	2
## 16	Niinemets	Leaf nitrogen (N) content per dry mass	93
## 17	Pe\xf1uelas	Leaf carbon/nitrogen (C/N) ratio	4
## 18	Pe\xf1uelas	Leaf nitrogen (N) content per dry mass	1818
## 19	Reich	Leaf carbon/nitrogen (C/N) ratio	507
## 20	Reich	Leaf nitrogen (N) content per dry mass	5307
## 21	Sack	Leaf nitrogen (N) content per dry mass	18

```
## 22      Shipley      Leaf carbon/nitrogen (C/N) ratio 186
## 23      Shipley      Leaf nitrogen (N) content per dry mass 277
## 24      Sosinski      Leaf carbon/nitrogen (C/N) ratio 4
## 25      Sosinski      Leaf nitrogen (N) content per dry mass 4
## 26 Soudzilovskaia      Leaf nitrogen (N) content per dry mass 44
## 27      Swaine      Leaf nitrogen (N) content per dry mass 4
## 28      van Bodegom      Leaf carbon/nitrogen (C/N) ratio 71
## 29      van Bodegom      Leaf nitrogen (N) content per dry mass 156
## 30      White      Leaf carbon/nitrogen (C/N) ratio 80
## 31      White      Litter carbon/nitrogen (C/N) ratio 93
## 32      Wright      Leaf nitrogen (N) content per dry mass 659
## 33      Yguel      Leaf nitrogen (N) content per dry mass 724
##      perc
## 1  0.98431656
## 2  0.01312422
## 3  1.41085373
## 4  0.41341295
## 5  1.25992519
## 6  16.88430999
## 7  1.48303694
## 8  2.73640003
## 9  0.09843166
## 10 2.65109259
## 11 3.78633769
## 12 0.36091607
## 13 0.24936019
## 14 1.71271081
## 15 0.01312422
## 16 0.61027626
## 17 0.02624844
## 18 11.92991666
## 19 3.32698996
## 20 34.82511976
## 21 0.11811799
## 22 1.22055253
## 23 1.81770457
## 24 0.02624844
## 25 0.02624844
## 26 0.28873286
## 27 0.02624844
## 28 0.46590984
## 29 1.02368922
## 30 0.52496883
## 31 0.61027626
## 32 4.32443074
## 33 4.75096791
```

```
newfilename<-'usedTryData_contrib.txt'
write.table(contribTab, file=paste(synthdataPath,newfilename, sep='/'), sep='\t')
```

1C. Create a *Traits x spID* matrix

```

# 1. Start with an empty spID (rows) x traits (cols) dataframe. Include all species recorded from paper
# 2. Fill the emptyDf with info from different sources/different qualities:
#     A. from within the original paper
#     B. from TRY data to the species level
#     C. from TRY data to the genus level
# 3. Sequentially merge the dataframes in order of the quality of the data, so that the higher quality
# 4. Finish with a table that has as much data as possible (preferentially keeping the high quality data)
# 5. Save the dataframe spIDtraits
source('CODE/cwm/script_t.R') # TASK= see above; NEEDS= species, traits, tryGS.summ, tryGX.summ; MAKES=
spIDtraits[1:10,c('spID','GenusSpecies','mean_cn','quality_mean_cn')]
spNam<-spIDtraits[1:10,c('Genus')]
tryGX.summ[tryGX.summ$Genus %in% spNam,]

newfilename<-'spIDtraits.txt'
write.table(spIDtraits, file=paste(synthdataPath,newfilename, sep='/'), sep='\t') #will need to use thi

```

2. PREP COVER DATA

2A. Estimate species cover in invaded and native areas for papers that present *multi-species cover values*

```

#Remember that...Var(aX) = a^2 * Var(X)
source('CODE/cwm/script_spcover.R') # TASK= Calculate species cover if possible; NEEDS= observations, cover.sp
tmp<-cover.sp[cover.sp$covNumSpp == '>1 species' ,c('obsID','covInvasive','covSpEntryID','covNumSpp','covMean')]
head(tmp)

```

2B. Create a *Cover x spID* matrix

```

# 1. Start with an empty spID (rows) x cover (cols) dataframe. Include all species recorded from papers
# 2. Fill the emptyDf with info from cover.sp
# 3. Finish with a table that has spID (species) x cover data and has an index of the quality of the cover
source('CODE/cwm/script_c.R') # TASK= see above; NEEDS= species, cover.sp; MAKES= spIDcover
View(spIDcover)

newfilename<-'spIDcover.txt'
write.table(spIDcover, file=paste(synthdataPath,newfilename, sep='/'), sep='\t') #will need to use this

```

3. CALCULATE COMMUNITY-WEIGHTED MEANS (CWM)

3A. For each trait type, merge the trait and cover matrices by spID, then subset by *minimum data requirements*, which are... ObsID must have at least 1 invasive species AND 1 native species that each have all of the following types of data: A) mean trait value for the traitOfInterest, B) mean species' cover in invaded area, C) mean species' cover in native area

```
traitsOfInterest<-c('cn', 'percN', 'littercn', 'litterpercN')
source('CODE/cwm/script_cwm_1.R') # TASK = Binds spIDtraits and spIDcover and determines spID that fit
#str(df.list)
#View(df.list[['cn']])
```

3B. Calculate the *relative abundance* for each spID in that obsID The relative abundance of each spID is based on the total percent cover in the invaded/native area of that observation. The relative abundances of all spIDs in that obsID should sum to 100%, even if the species covers do not sum to 100%. Note: Spcover values of the spIDs in an obsID might not sum to 100% if ... A) Not all of the species were measured, B) Bareground was included in the measurement, C) Cover values were measured at multiple levels - the understory and canopy

```
source('CODE/cwm/script_cwm_2.R') # TASK= Calculates the relative abundance for each spID and amends t
#View(df.list2[['cn']])
#str(df.list2)
#View(numberOfSpecies.tally)

newfilename<-'numberOfSpecies_cwm.txt'
write.table(numberOfSpecies.tally, file=paste(synthdataPath,newfilename, sep='/'), sep='\t')
```

3C. Calculate the *community-weighted trait mean* for each obsID (invaded area cwm, native area cwm, invasive species invaded area). DEFINITION: a community weighted mean value (hereafter CWM) can be calculated for each trait as the mean of trait values in the community, weighted by the relative abundance of the species carrying each value (Violle C, Navas ML, Vile D, Kazakou E, Fortunel C, Hummel I, Garnier E (2007) Oikos)

```
### make it so obsID has it own column
source('CODE/cwm/script_cwm_3.R') # TASK= Calculates the CWM in invaded and native areas for each obsID
#it is safe to ignore warnings about duplicated row names
#str(cwm.list)
```

3D. Merge spID-calculated and paper-reported CWM values into 1 CWM dataframe sequentially by quality

```
# 1. Determine the number of observations that report CWM trait values for invaded and native areas.
# 2. Start with an empty obsID (rows) x cwm traits (cols) dataframe. Include trait value quality metric.
#   A. Number of species included with cover values based on (i) 1 OR (ii) more than 1 species
#   B. Number of species included with cover values (i) measured within the paper OR (ii) esmiated ba
#   C. Number of species included with trait values (i) reported within the paper OR (ii) from specie
# 3. Sequentially merge the dataframe of spID-calculated CWMs and the paper-reported CWMs: paper-report
# 4. Finish with a table that has as much data as possible (preferentially keeping the high quality dat
source('CODE/cwm/script_cwm_4.R') # TASK= Merges CWM reported within studies with those that were calcu
#View(cwm.r.list[[1]])
```

4. ANNOTATE QUALITY

There are 4 ways in which the quality of a CWM value may be diminished:

- 1) The relative abundance of some species in the community were not available and I had to assume in some cases that species' relative abundances were equal across the species list provided (percMeasCov = Percent of species included in a CWM with measured cover values)

- 2) The relative abundance of some species in the community may be estimated from a multi-species cover value, rather than measured on a species-basis in the original study (perc1spCov = Percent of species included in a CWM with species-specific cover values (values can be measured or BOSD))
- 3) Trait data from some species in the community were not reported in the original paper (percOrigTr = Percent of species included in a CWM that were represented with trait data presented in the original paper)
- 4) Trait data from some species in the community were not even represented with trait data found in species-specific TRY database, only genus-specific (percTryGS = Percent of species included in a CWM that were represented with trait data found in species-specific TRY database)

5. CLEAN

Rename and reorganize columns

6. EXPORT

```
#View(df1)
newfilename<-'cwm.txt'
write.table(df1, file=paste(synthdataPath,newfilename, sep='/'), sep='\t')
```