Filtering datasets

Renata Diaz

2020-12-07

We filter datasets in two stages. First, **prior to trying to sample from the feasible set**, we remove communities that have a combined S and N too high for us to sample the feasible set. We also create a sub-sample of the FIA dataset, because it has so many communities (103,343) and so many of them are extremely small (92,988 with fewer than 10 species). Sampling all 100,000 communities overwhelms our computational pipeline, so we create a subsample of 20,000 communities comprising all communities with more than 10 species (10,355) and 10,000 randomly selected communities with 3-10 species. Second, **after sampling but before we aggregate results across communities**, we remove communities that are problematic for a number of more nuanced reasons. This includes having only 1 possible SAD (S = 1, N = S, or N = S + 1). We also filter to communities whose sampled feasible sets yield more than 20 unique values for skewness/evenness. We do this because, if there are fewer than 20 unique values in the comparison vector, it’s impossible to be in the 5th or 95th percentile. Finally, we remove communities with only 2 species from analyses for skewness, because e1071::skewness() always = 0 if S = 2.

## Pre-sampling

The only filtering at this stage is removing large communities and subsampling the FIA database. Communities with very large numbers of individuals become computationally intractable. We set the upper limit at 40720, because this is the largest community we were able to sample given the available resources. This upper limit results in the removal of a total of 4 communities, all of them from the Miscellaneous Abundance Database.

The download\_data function downloads raw data files from <https://github.com/weecology/sad-comparison/> (for BBS, Gentry, Mammal Community Database, and FIA) and figshare <http://figshare.com/files/3097079> (for the Miscellaneous Abundance Database). These raw files are stored in working-data\abund\_data and are not edited.

To filter, we can re-load the raw data files and go through the filtering process step by step. In the analysis this is accomplished by running dataset-specific filtering scripts and saving new .csvs, which are then loaded using load\_dataset. We can manually load the datasets and then compare them to what is returned from load\_dataset.

### Miscellaneous Abundance Database

* Misc. Abund includes datasets reported as relative abundance in addition to count data. We don’t want any communities without counts, so we filter out records where abund = 0.

misc\_abund\_raw <- read.csv(here::here("working-data", "abund\_data", "misc\_abund\_spab.csv"))  
  
misc\_abund\_raw <- misc\_abund\_raw %>%  
 dplyr::rename(site = Site\_ID,  
 abund = Abundance)  
  
misc\_abund\_raw <- misc\_abund\_raw %>%  
 mutate(site = as.character(site),  
 dat = "misc\_abund",  
 singletons = F,  
 sim = -99,  
 source = "observed") %>%  
 filter(abund > 0) %>%  
 group\_by(site) %>%  
 arrange(abund) %>%  
 mutate(rank = row\_number()) %>%  
 ungroup()  
  
misc\_abund\_loaded <- load\_dataset("misc\_abund")

any(!(misc\_abund\_loaded$abund == misc\_abund\_raw$abund))

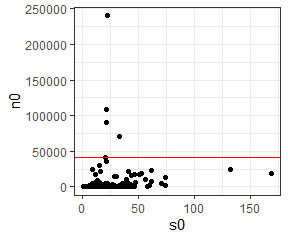
## [1] FALSE

any(!(misc\_abund\_loaded$site == misc\_abund\_raw$site))

## [1] FALSE

Check community sizes:

misc\_abund\_statevars <- get\_statevars(misc\_abund\_raw)  
  
ggplot(misc\_abund\_statevars, aes(s0, n0)) +  
 geom\_point() +  
 theme\_bw() +  
 geom\_hline(yintercept = 40720, color = "red")



Misc abund has 4 communities that get removed. They are all removed because they have high numbers of individuals.

The filtered database is saved as a .csv and can be loaded with load\_dataset. We can check that it matches the filtering we have done here:

misc\_abund\_sv\_filtered <- misc\_abund\_statevars %>%  
 filter(n0 <= 40720)  
  
misc\_abund\_filtered <- filter(misc\_abund\_raw, site %in% misc\_abund\_sv\_filtered$site)  
  
misc\_abund\_short\_loaded <- load\_dataset("misc\_abund\_short")  
  
any(!(misc\_abund\_short\_loaded$abund == misc\_abund\_filtered$abund))

## [1] FALSE

any(!(misc\_abund\_short\_loaded$site == misc\_abund\_filtered$site))

## [1] FALSE

### FIA

* Load raw FIA data
* Add columns to match what we will get from load\_dataset
* Load from load\_dataset

fia\_raw <- read.csv(here::here("working-data", "abund\_data", "fia\_spab.csv"), stringsAsFactors = F, header = F, skip = 2)  
  
colnames(fia\_raw) <- c("site", "year", "species", "abund")  
  
fia\_raw <- fia\_raw %>%  
 mutate(site = as.character(site),  
 dat = "fia",  
 singletons = F,  
 sim = -99,  
 source = "observed") %>%  
 filter(abund > 0) %>%  
 group\_by(site) %>%  
 arrange(abund) %>%  
 mutate(rank = row\_number()) %>%  
 ungroup()  
  
fia\_loaded <- load\_dataset("fia")

any(!(fia\_loaded$abund == fia\_raw$abund))

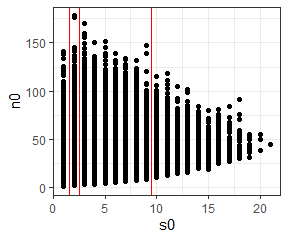
## [1] FALSE

any(!(fia\_loaded$site == fia\_raw$site))

## [1] FALSE

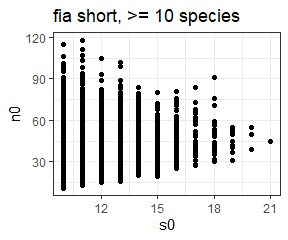
Check community sizes:

fia\_statevars <- get\_statevars(fia\_raw)  
  
ggplot(fia\_statevars, aes(s0, n0)) +  
 geom\_point() +  
 theme\_bw() +  
 geom\_vline(xintercept = c(1.5, 2.5, 9.5), color = "red")

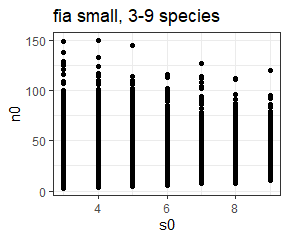


FIA has no extremely large datasets; the largest number of individuals is 178. However, it has 103343 communities, of which 92988 have fewer than 10 species. This many communities overwhelms our computational pipeline. We therefore sample all 10355 communities with 10 or more species, and a random subsample of 10,000 communities with 3-9 species. We then run these through the pipeline as two separate databases. fia\_short is the communities with 10 or more species, and fia\_small is the 10,000 communities with 3-9 species. We re-combine them as “FIA” for aggregate analyses.

fia\_sv\_short <- fia\_statevars %>%  
 dplyr::filter(s0 >= 10)  
  
fia\_short <- fia\_raw %>%  
 dplyr::filter(site %in% fia\_sv\_short$site) %>%  
 dplyr::mutate(dat = "fia\_short")  
  
fia\_short\_statevars <- get\_statevars(fia\_short)  
  
ggplot(fia\_short\_statevars, aes(s0, n0)) +  
 geom\_point() +  
 ggtitle("fia short, >= 10 species") +  
 theme\_bw()



fia\_sv\_small <- fia\_statevars %>%  
 dplyr::filter(s0 >= 3) %>%  
 dplyr::filter(s0 <= 9)  
  
 set.seed(1977)  
 fia\_sv\_small <- fia\_sv\_small[ sample.int(nrow(fia\_sv\_small), size = 10000, replace = F), ]  
  
  
  
fia\_small <- fia\_raw %>%  
 dplyr::filter(site %in% fia\_sv\_small$site) %>%  
 dplyr::mutate(dat = "fia\_small")  
  
fia\_small\_statevars <- get\_statevars(fia\_small)  
  
ggplot(fia\_small\_statevars, aes(s0, n0)) +  
 geom\_point() +  
 ggtitle("fia small, 3-9 species")



The “short” and “small” datasets are saved as .csvs and can be loaded using load\_dataset:

fia\_small\_loaded <- load\_dataset("fia\_small")  
  
any(!(fia\_small\_loaded$abund == fia\_small$abund))

## [1] FALSE

any(!(fia\_small\_loaded$site == fia\_small$site))

## [1] FALSE

fia\_short\_loaded <- load\_dataset("fia\_short")  
  
any(!(fia\_short\_loaded$abund == fia\_short$abund))

## [1] FALSE

any(!(fia\_short\_loaded$site == fia\_short$site))

## [1] FALSE

### BBS

* Load raw BBS data
* Add columns to match what we will get from load\_dataset
* Load from load\_dataset

bbs\_raw <- read.csv(here::here("working-data", "abund\_data", "bbs\_spab.csv"), stringsAsFactors = F, header = F, skip = 2)  
  
colnames(bbs\_raw) <- c("site", "year", "species", "abund")  
  
bbs\_raw <- bbs\_raw %>%  
 mutate(site = as.character(site),  
 dat = "bbs",  
 singletons = F,  
 sim = -99,  
 source = "observed") %>%  
 group\_by(site) %>%  
 arrange(abund) %>%  
 mutate(rank = row\_number()) %>%  
 ungroup()  
  
bbs\_loaded <- load\_dataset("bbs")

Compare loaded to raw:

any(!(bbs\_loaded$abund == bbs\_raw$abund))

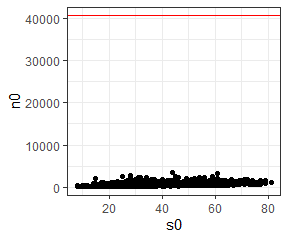
## [1] FALSE

any(!(bbs\_loaded$site == bbs\_raw$site))

## [1] FALSE

Check community sizes:

bbs\_statevars <- get\_statevars(bbs\_raw)  
  
ggplot(bbs\_statevars, aes(s0, n0)) +  
 geom\_point() +  
 theme\_bw() +  
 geom\_hline(yintercept = 40720, color = "red")



No communities in BBS have more than 40720 individuals, so all are used at this stage.

### Gentry

* Load raw Gentry data
* Add columns to match what we will get from load\_dataset
* Load from load\_dataset

gentry\_raw <- read.csv(here::here("working-data", "abund\_data", "gentry\_spab.csv"), stringsAsFactors = F, header = F, skip = 2)  
  
colnames(gentry\_raw) <- c("site", "year", "species", "abund")  
  
gentry\_raw <- gentry\_raw %>%  
 mutate(site = as.character(site),  
 dat = "gentry",  
 singletons = F,  
 sim = -99,  
 source = "observed") %>%  
 group\_by(site) %>%  
 arrange(abund) %>%  
 mutate(rank = row\_number()) %>%  
 ungroup()  
  
gentry\_loaded <- load\_dataset("gentry")

Compare loaded to raw:

any(!(gentry\_loaded$abund == gentry\_raw$abund))

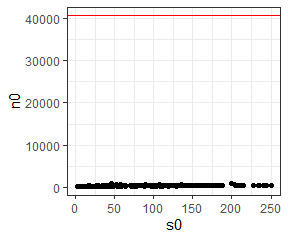
## [1] FALSE

any(!(gentry\_loaded$site == gentry\_raw$site))

## [1] FALSE

Check community sizes:

gentry\_statevars <- get\_statevars(gentry\_raw)  
  
ggplot(gentry\_statevars, aes(s0, n0)) +  
 geom\_point() +  
 theme\_bw() +  
 geom\_hline(yintercept = 40720, color = "red")



No communities in Gentry have more than 40720 individuals, so all are used at this stage.

### Mammal Community Database

* Load raw MCDB data
* Add columns to match what we will get from load\_dataset
* Load from load\_dataset

mcdb\_raw <- read.csv(here::here("working-data", "abund\_data", "mcdb\_spab.csv"), stringsAsFactors = F, header = F, skip = 2)  
  
colnames(mcdb\_raw) <- c("site", "year", "species", "abund")  
  
mcdb\_raw <- mcdb\_raw %>%  
 mutate(site = as.character(site),  
 dat = "mcdb",  
 singletons = F,  
 sim = -99,  
 source = "observed") %>%  
 group\_by(site) %>%  
 arrange(abund) %>%  
 mutate(rank = row\_number()) %>%  
 ungroup()  
  
mcdb\_loaded <- load\_dataset("mcdb")

Compare loaded to raw:

any(!(mcdb\_loaded$abund == mcdb\_raw$abund))

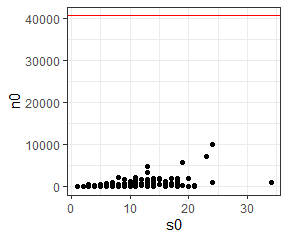
## [1] FALSE

any(!(mcdb\_loaded$site == mcdb\_raw$site))

## [1] FALSE

Check community sizes:

mcdb\_statevars <- get\_statevars(mcdb\_raw)  
  
ggplot(mcdb\_statevars, aes(s0, n0)) +  
 geom\_point() +  
 theme\_bw() +  
 geom\_hline(yintercept = 40720, color = "red")



No high N communities; so all are included at this stage.

## Post-sampling

We will remove additional communities for more substantive reasons. First we can load all\_di (“all diversity indices”), which is the combined results across all the communities in all the datasets. Every community that was included for sampling is included in all\_di. (Additionally, every community has a “singletons” counterpart, which is the same analysis run adjusted for rarefaction. That analysis is discussed elsewhere, and we ignore the rarefaction-adjusted versions here).

all\_statevars <- bind\_rows(bbs\_statevars, fia\_short\_statevars, fia\_small\_statevars, gentry\_statevars, mcdb\_statevars, misc\_abund\_sv\_filtered)  
  
  
all\_di <- read.csv(here::here("analysis", "reports", "all\_di.csv"), stringsAsFactors = F)  
  
all\_di <- all\_di %>%  
 filter(!singletons) %>%  
 mutate(dat = ifelse(grepl(dat, pattern = "fia"), "fia", dat),  
 dat = ifelse(dat == "misc\_abund\_short", "misc\_abund", dat)) %>%  
 mutate(Dataset = dat,  
 Dataset = ifelse(Dataset == "fia", "Forest Inventory and Analysis", Dataset),  
 Dataset = ifelse(Dataset == "bbs", "Breeding Bird Survey", Dataset),  
 Dataset = ifelse(Dataset == "mcdb", "Mammal Community DB", Dataset),  
 Dataset = ifelse(Dataset == "gentry", "Gentry", Dataset),  
 Dataset = ifelse(Dataset == "misc\_abund", "Miscellaneous Abundance DB", Dataset))  
  
head(all\_di)

## sim source dat site singletons s0 n0 nparts skew shannon  
## 1 -99 observed bbs 4002 FALSE 35 816 4.070604e+24 1.973575 2.655380  
## 2 -99 observed bbs 4003 FALSE 26 439 1.765612e+17 1.502247 2.609893  
## 3 -99 observed bbs 4004 FALSE 27 445 3.100591e+17 1.802663 2.748364  
## 4 -99 observed bbs 4006 FALSE 29 471 1.689307e+18 2.728160 2.432257  
## 5 -99 observed bbs 4007 FALSE 24 575 1.424232e+19 1.216833 2.491605  
## 6 -99 observed bbs 4009 FALSE 21 534 6.506488e+17 1.848653 2.420594  
## simpson skew\_percentile shannon\_percentile simpson\_percentile  
## 1 0.9016995 69.60 0.050 1.375  
## 2 0.9009605 50.75 8.800 15.225  
## 3 0.9138139 64.45 23.775 28.450  
## 4 0.8597058 92.65 0.425 0.800  
## 5 0.8915962 36.80 4.550 11.475  
## 6 0.8785787 79.55 8.050 11.325  
## skew\_percentile\_excl simpson\_percentile\_excl skew\_range simpson\_range  
## 1 69.60 1.375 5.013388 0.1071673  
## 2 50.75 15.225 4.406489 0.1807068  
## 3 64.45 28.450 4.384616 0.3824770  
## 4 92.65 0.800 4.369746 0.1462940  
## 5 36.80 11.450 4.293175 0.1741853  
## 6 79.55 11.300 3.932031 0.1747465  
## nsamples skew\_unique simpson\_unique skew\_2p5 skew\_97p5 skew\_95 skew\_min  
## 1 4000 4000 3445 0.7046398 3.358540 2.998938 0.05485603  
## 2 4000 4000 2860 0.5050514 3.157090 2.846737 -0.19707578  
## 3 4000 4000 2883 0.5441635 3.182604 2.879527 0.02681027  
## 4 4000 4000 2948 0.5689934 3.270669 2.928296 -0.16145549  
## 5 4000 4000 3283 0.4274895 2.971295 2.710518 -0.21809489  
## 6 4000 4000 3267 0.3496395 2.780707 2.531622 -0.38791363  
## simpson\_max simpson\_2p5 simpson\_5 simpson\_97p5 skew\_95\_ratio\_2t  
## 1 0.9631061 0.9064803 0.9144545 0.9551500 0.5293626  
## 2 0.9517697 0.8671956 0.8799140 0.9437090 0.6018485  
## 3 0.9522030 0.8740614 0.8860598 0.9451334 0.6017496  
## 4 0.9562164 0.8815458 0.8929864 0.9484996 0.6182684  
## 5 0.9501822 0.8638114 0.8769887 0.9396030 0.5925231  
## 6 0.9418353 0.8506425 0.8631493 0.9325567 0.6182726  
## simpson\_95\_ratio\_2t skew\_95\_ratio\_1t simpson\_95\_ratio\_1t Dataset  
## 1 0.4541467 0.5872441 0.4539785 Breeding Bird Survey  
## 2 0.4234121 0.6907570 0.3976368 Breeding Bird Survey  
## 3 0.1858206 0.6506195 0.1729337 Breeding Bird Survey  
## 4 0.4576662 0.7070780 0.4322117 Breeding Bird Survey  
## 5 0.4351207 0.6821556 0.4202049 Breeding Bird Survey  
## 6 0.4687608 0.7425008 0.4502870 Breeding Bird Survey

nrow(all\_di) == nrow(all\_statevars)

## [1] TRUE

We remove communities with only one possible SAD (N = S, N = S + 1, or S = 1).

all\_di %>%  
 group\_by\_all() %>%  
 mutate(only\_one\_sad = sum(s0 == n0, s0 == 1, n0 == (s0 + 1)) > 0) %>%  
 ungroup() %>%  
 group\_by(dat) %>%  
 summarize(total\_only\_one\_sad = sum(only\_one\_sad),  
 total\_sites = dplyr::n()) %>%  
 ungroup() %>%  
 mutate(all\_sites\_one\_sad = sum(total\_only\_one\_sad),  
 all\_sites = sum(total\_sites))

## # A tibble: 5 x 5  
## dat total\_only\_one\_sad total\_sites all\_sites\_one\_sad all\_sites  
## <chr> <int> <int> <int> <int>  
## 1 bbs 0 2773 258 24647  
## 2 fia 176 20355 258 24647  
## 3 gentry 0 224 258 24647  
## 4 mcdb 56 730 258 24647  
## 5 misc\_abund 26 565 258 24647

all\_di\_filtered <- all\_di %>%  
 filter(s0 != n0,  
 s0 != 1,  
 n0 != (s0 + 1))  
  
nrow(all\_di\_filtered)

## [1] 24389

Removing those with only one SAD results in the removal of 258 sites total. 176 from FIA, 56 from MCDB, and 26 from Misc. Abund.

Finally, we will restrict aggregate analyses to sites whose feasible sets have more than 20 unique values for whichever shape metric we’re interested in, **and** we will restrict analyses with skewness to sites with >2 species. This results in these final totals:

all\_di\_filtered %>%  
 filter(simpson\_unique > 20) %>%  
 group\_by(dat) %>%  
 summarize(sites\_for\_evenness = dplyr::n()) %>%  
 ungroup() %>%  
 mutate(total\_sites\_for\_evenness = sum(sites\_for\_evenness))

## # A tibble: 5 x 3  
## dat sites\_for\_evenness total\_sites\_for\_evenness  
## <chr> <int> <int>  
## 1 bbs 2773 22142  
## 2 fia 18113 22142  
## 3 gentry 224 22142  
## 4 mcdb 542 22142  
## 5 misc\_abund 490 22142

all\_di\_filtered %>%  
 filter(skew\_unique > 20, s0 > 2) %>%  
 group\_by(dat) %>%  
 summarize(sites\_for\_skewness = dplyr::n()) %>%  
 mutate(total\_sites\_for\_skewness = sum(sites\_for\_skewness))

## # A tibble: 5 x 3  
## dat sites\_for\_skewness total\_sites\_for\_skewness  
## <chr> <int> <int>  
## 1 bbs 2773 22325  
## 2 fia 18300 22325  
## 3 gentry 223 22325  
## 4 mcdb 537 22325  
## 5 misc\_abund 492 22325